LEONARDO DE BRITO JORDÃO

USO DO PROTOCOLO DE AUTENTICAÇÃO KERBEROS EM REDES LINUX

Monografia apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras, como parte das exigências do curso de Pós-Graduação *Lato Sensu* em Administração de Redes Linux (ARL) para obtenção do título de Especialista em Administração de Redes Linux.

Orientador
Professor Joaquim Quinteiro Uchôa

Lavras
Minas Gerais – Brasil
2005

LEONARDO DE BRITO JORDÃO

USO DO PROTOCOLO DE AUTENTICAÇÃO KERBEROS EM REDES LINUX

Monografia apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras, como parte das exigências do curso de Pós-Graduação *Lato Sensu* em Administração de Redes Linux (ARL) para obtenção do título de Especialista em Administração de Redes Linux.

Aprovada em *16 de Abril de 2005*

Professor Herlon Ayres Camargo

Professor Ricardo Martins de Abreu Silva

Professor Joaquim Quinteiro Uchôa
(Orientador)

Lavras
Minas Gerais – Brasil

À minha esposa Lucineide e aos meus filhos,
Lucas e Mariana, pelo apoio e compreensão .

Agradecimentos

Agradeço a Deus, acima de tudo, e a meus pais, por tudo que me deram e proporcionaram.
Agradeço a meus amigos e a todos que, direta ou indiretamente, contribuíram para que eu pudesse concluir esse curso.

Resumo

O protocolo Kerberos é um sistema de autenticação segura para aplicações cliente/servidor que utiliza criptografia de chave simétrica para autenticar usuários e servidores, eliminando o tráfego de senhas pela rede e garantindo segurança a sistemas de informação. O Kerberos é um sistema de *logon* único, onde a senha é informada uma única vez e o Kerberos efetua a autenticação e criptografia de forma transparente entre usuários e serviços. Este documento descreve o protocolo de autenticação Kerberos e apresenta os passos para instalação e configuração de servidores de autenticação Kerberos em uma rede Linux.

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

Muitos dos sistemas computacionais utilizam esquemas de autenticação baseados em senhas, onde um usuário precisa se identificar para o servidor que deseja utilizar fornecendo um código de usuário e uma senha. Para muitos serviços, a transmissão das informações de autenticação se dá sem o uso de criptografia. Para se ter segurança nessas condições, a rede deveria ser inacessível externamente e todos os computadores e usuários da rede interna deveriam ser confiáveis.

Uma alternativa para um sistema de autenticação segura é a utilização do protocolo Kerberos[1]. O objetivo primário pelo qual o Kerberos foi projetado é a eliminação da transmissão de senhas descriptografadas pela rede. Se implantado e utilizado adequadamente o Kerberos acrescenta um novo nível de segurança à rede.

Neste documento são apresentados os conceitos e terminologia do Kerberos, além dos passos para instalação e configuração de servidores de autenticação baseada em Kerberos na plataforma GNU/Linux.

O Capítulo 2 apresenta o protocolo Kerberos em linhas gerais, seu funcionamento básico e elementos principais.

O Capítulo 3 mostra o Kerberos dentro de um ambiente GNU/Linux, os serviços básicos para fornecimento de credenciais de autenticação e como interagem os diversos componentes do Kerberos numa rede Linux.

O Capítulo 4 trata do planejamento da instalação e a configuração dos

1 Kerberos vem do grego Cerberus e, na mitologia grega, era o nome do cão de três cabeças que guardava a entrada do mundo inferior.

servidores Kerberos, servidores de aplicação e clientes numa rede que utiliza o Kerberos para autenticação.

O Capítulo 5 apresenta os principais comandos para administração do ambiente Kerberos e, finalmente, o Capítulo 6 apresenta aspectos relativos a possíveis falhas de segurança e formas de ataque os quais o Kerberos se propõe a reduzir; vantagens e desvantagens de sua utilização.

Capítulo 2

O Protocolo Kerberos

2.1 Visão geral

Kerberos é um protocolo de autenticação de rede criado pelo *Massachusetts Institute of Technology* (MIT[2]) e que utiliza criptografia de chave simétrica para autenticar usuários no acesso a serviços de rede – eliminando o envio de senhas através da rede – provendo um sistema de autenticação segura para aplicações cliente/servidor. O Kerberos suporta diversos tipos de criptografia de chave simétrica, entre eles DES, 3DES, AES-128, AES-256 e RC4. Mais sobre criptografia pode ser visto em [FERGUSON; SCHNEIER (2003)].

O Kerberos é um sistema de *logon* único, onde a senha é informada uma única vez e o Kerberos efetua a autenticação e criptografia de forma transparente. Quando um usuário efetua *logon*, o Kerberos o autentica utilizando a senha desse usuário e fornece um mecanismo pelo qual o mesmo pode provar sua identidade a servidores e serviços sem fornecer a senha novamente.

A instalação do Kerberos e dos programas que o utilizam reduz a possibilidade de captura de senhas por usuários mal-intencionados ou de falsificação de identidade, onde um usuário tenta se passar por outro. Em suma, o Kerberos é uma solução que provê ferramentas de autenticação e criptografia forte sobre a rede para garantir segurança a sistemas de informação.

O Kerberos foi desenvolvido pelo MIT em um projeto chamado *Athena* e uma

2 http://web.mit.edu/kerberos/www

implementação livre desse protocolo está disponível pelo MIT sob licenciamento semelhante aos utilizados pelo BSD e X-Window. O MIT disponibiliza o Kerberos em código-fonte, de modo que seu conteúdo pode ser revisto e analisado por quem deseja implementá-lo, como forma de certificação do conteúdo, aumentando a confiabilidade na instalação. Para os que preferem produtos com suporte profissional, há versões do Kerberos disponibilizadas por diversos fornecedores, além de implementações do Kerberos em vários produtos comerciais.

Diversos programas servidores e clientes em Linux foram modificados para suportar o uso do Kerberos, incluindo-se `telnet`, `rlogin`, `rsh`, `rcp` e `ftp`. Há também bibliotecas e APIs para permitir a adição de segurança baseada em Kerberos nos aplicativos.

Há duas versões do Kerberos atualmente em uso: a versão 4 e a versão 5. As versões de 1 a 3 foram versões de desenvolvimento e nunca foram disponibilizadas. A versão 4 possui diversas vulnerabilidades e seu uso deve ser evitado. O pacote do Kerberos V5 foi projetado para ser de fácil uso. Este documento trata da versão 5 do Kerberos, definido na RFC 1510 [KOHL; NEUMAN (1993)].

2.2 Razões para a utilização do Kerberos

As redes são inseguras e grande parte dos protocolos utilizados não provê segurança. Aplicações que enviam senhas abertas através de uma rede estão vulneráveis a, por exemplo, ferramentas para captura de senhas. Muitas aplicações cliente/servidor confiam nos programas clientes quanto à identidade do usuário sem nenhum outro reforço de segurança por parte do servidor.

Alguns ambientes utilizam *firewalls* para solucionar problemas de segurança nas redes. Infelizmente, essa opção considera que os “maus elementos” estão do lado de fora da empresa, o que nem sempre é verdade já que há ataques também por parte de usuários internos de uma rede.

Uma vez que o Kerberos negocia comunicação autenticada e, opcionalmente, criptografada, entre dois pontos numa rede, ele provê uma camada de segurança que

não depende, por exemplo, de qual lado de um *firewall* o usuário está, tornando a implantação do Kerberos de grande importância dentro de uma rede corporativa.

O protocolo Kerberos foi criado como uma solução para problemas de segurança de autenticação em redes. Com o Kerberos um cliente pode provar sua identidade a um servidor (e vice-versa) numa conexão de rede insegura. Depois de autenticados pelo Kerberos, cliente e servidor podem também criptografar toda a sua comunicação de modo a garantir privacidade e integridade dos dados trafegados.

Ou seja, o Kerberos é um sistema que auxilia a solucionar algumas questões de segurança: reduzir o número de senhas que o usuário necessita para acessar diversos serviços a apenas uma senha no Kerberos e acrescentar criptografia aos dados trafegados, assegurando que dados sensíveis não trafeguem de forma aberta na rede.

2.3 Funcionamento básico do Kerberos

O Kerberos difere de outros métodos de autenticação. Ao invés de autenticar cada usuário em cada serviço de rede, o Kerberos utiliza uma chave secreta comum e um "mediador" confiável para validação da identidade dos clientes frente a um conjunto de serviços de rede. O "mediador" confiável é um servidor conhecido como **KDC** (*Key Distribution Center* – Centro de Distribuição de Chaves) que executa os *deamons* do Kerberos. A chave secreta comum é a senha do usuário convertida numa chave para criptografia. No caso de servidores ou aplicações a chave, comumente, é gerada de forma aleatória.

No Kerberos, os clientes podem ser usuários, servidores ou programas, e são conhecidos como ***principals*** (ou "**principais**"). O KDC armazena a base de dados de principais e suas chaves, que são utilizadas para autenticação.

Para o Kerberos, o conhecimento da chave secreta é considerado suficiente como prova de identidade. Assim, o servidor Kerberos é confiado para autenticar qualquer cliente para qualquer outro cliente. A autenticação é feita de forma que não há o envio de senhas através da rede em momento algum.

O funcionamento do Kerberos se dá, basicamente, da seguinte forma: um cliente envia uma solicitação ao KDC e o KDC procura esse principal na sua base de dados.

Se é encontrado, o KDC cria um **TGT** (*Ticket-Granting Ticket – ticket* para obter *ticket*) para o cliente, criptografa o TGT usando a senha do cliente como chave e envia o TGT criptografado para o cliente.

O cliente tenta descriptografar o TGT, usando sua própria chave (computada a partir da sua senha). Se o cliente consegue descriptografar o TGT, este é mantido em *cache* na máquina do cliente para futuras provas de identidade perante qualquer serviço "kerberizado", uma vez que, obtido o TGT, não é mais necessário informar a senha novamente até que o TGT expire ou que o usuário efetue novo *logon*. O TGT é marcado para expirar depois de certo tempo (geralmente, 8 horas). O tempo de expiração é utilizado para que, caso um TGT seja capturado por um invasor, este o tenha apenas por um período de tempo.

O TGT permite ao cliente obter *tickets* adicionais para acesso a serviços. A solicitação e obtenção desses *tickets* adicionais é transparente ao usuário. Sempre que o usuário precisar acessar um serviço de rede, o software cliente usa o TGT para solicitar ao KDC um novo *ticket* específico para aquele serviço, não sendo necessária nova autenticação utilizando uma senha. Esta é apenas uma visão geral de como a autenticação Kerberos funciona numa rede. Maiores detalhes são apresentados no Capítulo 3.

2.4 Infraestrutura do Kerberos

O termo infraestrutura do Kerberos refere-se aos programas, servidores, clientes e configurações que permitem a utilização do protocolo Kerberos em uma rede.

Essa infraestrutura consiste do software do Kerberos, servidores de autenticação redundantes e seguros, um armazenamento centralizado de contas e senhas, e sistemas configurados para utilizar a autenticação através do protocolo Kerberos.

Além da infraestrutura inerente ao próprio Kerberos, este depende de certos serviços de rede para seu correto funcionamento. Em primeiro lugar, o Kerberos requer que os horários das máquinas da rede estejam sincronizados entre si. Para isso, um programa de sincronização de horário deve existir na rede. O *Network Time Protocol (*NTP), definido na RFC 1305 [MILLS (1992)], é o protocolo mais

indicado e o mais utilizado para essa finalidade. No Linux, o NTP é implementado pelo `ntpd`. Mais sobre o NTP pode ser visto no Capítulo 4.

Outro serviço importante no funcionamento do Kerberos é o *Domain Name Service* (DNS), definido nas RFC 1034 [MOCKAPETRIS (1987) (1)] e RFC 1035 [MOCKAPETRIS (1987) (2)]. Certos aspectos do Kerberos dependem de que o DNS e máquinas da rede estejam devidamente configurados. Algumas considerações sobre DNS no Kerberos são mostradas no Capítulo 4.

Capítulo 3

O Protocolo Kerberos no GNU/Linux

Como já exposto no Capítulo 2, o Kerberos efetua a autenticação dos usuários sem que a senha seja transmitida pela rede. O Kerberos baseia sua autenticação em *tickets* eletrônicos que são enviados criptografados pela rede. Um servidor central chamado KDC autentica o usuário e o fornece um *ticket* que o permite acessar máquinas e serviços.

Para o Kerberos, cada usuário, máquina ou serviço é chamado de **principal** e cada principal possui uma chave. Para usuários, a chave é criada a partir da senha individual e para máquinas e serviços, a partir de um número aleatório. A chaves, assim como as senhas, são mantidas secretas e são conhecidas apenas pelo principal ao qual a chave pertence e pelo KDC.

As chaves são utilizadas para autenticação e envio seguro de mensagens através da criptografia de chave simétrica, o que significa que se uma mensagem é criptografada com uma chave, ela só pode ser descriptografada com a mesma chave. Uma vez que a chave só é conhecida por duas partes, o seu principal e o KDC, isso permite uma troca segura de mensagens entre eles.

O servidor KDC executa dois importantes *deamons* do Kerberos V5, o `kadmind` e o `krb5kdc`, que são executados como super-usuário (*root*). O `kadmind` é o *deamon* administrativo do servidor Kerberos e é utilizado pelo programa `kadmin` para manutenção da base de dados de principais e configuração de políticas. O `kadmin` permite a administração remota do Kerberos, mesmo se não houverem programas de acesso remoto ao servidor Kerberos. O `krb5kdc` é o

principal programa do Kerberos, sendo responsável por realizar o papel de "mediador" na autenticação Kerberos, ou seja, é o programa que executa o KDC.

No Capítulo 2 foi apresentado o funcionamento do Kerberos em linhas gerais. Neste Capítulo é mostrado com maiores detalhes o processo de fornecimento de *tickets* e autenticação.

3.1 Elementos do Kerberos

3.1.1 Reino Kerberos

Um reino (*realm*) Kerberos é um domínio administrativo que consiste de um conjunto de máquinas que compartilham informações contidas numa base de dados Kerberos em comum e confiam nos mesmos servidores KDC.

É possível efetuar autenticação entre usuários e serviços de reinos diferentes através de relações de confiança entre dois reinos, chamadas "autenticação entre reinos" (*cross-realm authentication*). Para isso os administradores dos dois reinos compartilham chaves, de modo que um reino consegue tratar *tickets* do outro reino. Com isso, principais de um reino conseguem se autenticar em outro reino.

O reino possui um nome que o identifica e, geralmente, é o mesmo nome do domínio DNS, porém em letras maiúsculas. Apesar dessa característica não ser obrigatória, ela tem sido comumente usada nas implementações do Kerberos.

3.1.2 Principal do Kerberos

Um principal do Kerberos é qualquer entidade que deva ser autenticada. Isso inclui usuários, máquinas e serviços. Cada principal possui uma chave secreta associada, a qual só é conhecida pelo próprio principal e pelo KDC.

Tradicionalmente, um principal é formado por três partes:

- primário – é a primeira parte do principal. Pode ser o nome de um usuário ou serviço. Para um servidor é a palavra *host*.
- instância – serve para qualificar o primário. É uma *string* separada do primário por uma barra (/). Pode ser vazia ou indicar características especiais:

no caso de um usuário descreve possíveis credenciais; no caso de uma máquina é o seu nome DNS completo.

- reino (*realm*) – parte do nome do principal que indica a qual reino o principal pertence. Um reino é a estrutura lógica da rede atendida pelo Kerberos.

O formato típico de um principal é *primário/instância@REINO*. Para um principal de usuário pode-se ter:

- josex@MYWARE.NET
- josex/admin@MYWARE.NET

onde a instância *admin* é típica para indicar que o principal tem privilégios administrativos no Kerberos. Os dois principais acima são entidades diferentes, possuindo senhas e permissões distintas.

No caso de principais de máquinas ou serviços, a parte primária indica o tipo de serviço, por exemplo *host* ou *ftp*. A instância indica a máquina na qual o serviço está disponível, por exemplo:

- host/server1.myware.net@MYWARE.NET
- ftp/ftp.df.myware.net@MYWARE.NET

Se o nome do reino não for informado no principal, será assumido que o principal está no reino padrão do contexto utilizado, informado nos arquivos de configuração do Kerberos.

3.1.3 A base de dados Kerberos

Cada reino Kerberos possui pelo menos um servidor KDC que armazena a base de dados mestre do Kerberos para aquele reino. Um reino Kerberos pode ter servidores KDC escravos, que possuem cópias de leitura da base de dados Kerberos, a qual é replicada periodicamente do KDC mestre para os escravos. Mais informações sobre servidores KDC escravos são apresentadas no Capítulo 4.

O Kerberos só fornece credenciais àquelas entradas existentes na base de dados do Kerberos. Cada usuário, serviço ou máquina no ambiente Kerberos deve possuir uma entrada na base de dados. A base de dados do Kerberos possui os principais e suas respectivas senhas.

3.1.4 O arquivo *keytab*

Cada principal na base de dados possui uma senha. Os usuários recebem sua senha do administrador e a armazenam de alguma forma (memória, agenda etc.). Da mesma forma, cada serviço possui uma chave, conhecida apenas pelo KDC e pelo próprio serviço.

No servidor KDC, a chave dos serviços fica armazenada na base de dados do Kerberos. No servidor que executa o serviço, porém, as chaves são armazenadas localmente em tabelas de chaves, que são arquivos conhecidos como *keytabs*, geralmente armazenados no arquivo `/etc/krb5.keytab`.

O serviço utiliza o arquivo *keytab* da mesma forma que um usuário usa sua senha, ou seja, quando há necessidade de autenticação, o servidor utiliza as chaves armazenadas no arquivo *keytab* e as apresenta ao KDC para se identificar. O Capítulo 4 apresenta mais detalhes sobre a configuração de serviços e a necessidade das chaves armazenadas no arquivo *keytab*.

3.1.5 *Tickets* Kerberos

As credenciais do Kerberos, ou *tickets*, são um conjunto de informações eletrônicas que podem ser usadas para verificar a identidade de um usuário, servidor ou serviço. Os *tickets* são armazenados num *cache* de credenciais, geralmente em arquivos temporários no `/tmp`. É através de um programa cliente que as credenciais para identificar um usuário são automaticamente obtidas a partir de um servidor KDC na rede.

O primeiro *ticket* a ser obtido é o *Ticket-Granting Ticket* (TGT), o qual permite obter *tickets* adicionais para acesso a serviços Kerberos. A solicitação e obtenção desses *tickets* adicionais acontecem de forma transparente. Um exemplo de solicitação do TGT é através do comando `kinit`. Uma vez executado, o `kinit` solicita a senha do usuário, que, se informada corretamente, irá obter do KDC um TGT e uma chave de sessão para permitir o uso correto do TGT. Essa combinação de TGT mais chave de sessão é conhecida como credencial.

Uma vez obtida a credencial, os programas clientes executados pelo usuário podem fazer uso dessa credencial para obter do KDC *tickets* específicos para acesso

a outros serviços na rede Kerberos. De posse desses *tickets* o programa cliente pode provar a identidade do usuário e obter acesso ao serviço.

Um exemplo é o *deamon* `klogind` que, executando num servidor, permite acesso remoto a esse servidor. O programa `rlogin`, executado na máquina do usuário, apresenta ao KDC as credenciais desse usuário e obtém um *ticket* específico para provar sua identidade e obter acesso remoto ao servidor.

3.2 Interação entre o usuário e o Kerberos

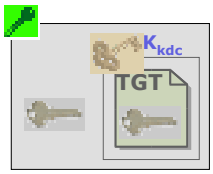
3.2.1 Acessando o KDC

A autenticação do Kerberos é normalmente iniciada pelo usuário executando um programa de solicitação de autenticação do Kerberos. Um exemplo de programa desse tipo é o `kinit`. O `kinit` envia uma mensagem ao KDC solicitando um TGT (que permitirá solicitar novos *tickets* no futuro) para o usuário. Essa mensagem é enviada aberta e não contém informações sensíveis, apenas a solicitação.

Em sistemas com o Kerberos V5 a utilização do programa `login` do Kerberos permite que esse processo aconteça no processo de *logon* do usuário, sem a necessidade de execução de um programa separado para solicitar o TGT. Após a solicitação do TGT, o seguinte processo acontece:

1. O KDC (`krb5kdc`) recebe a solicitação e busca pelo principal do usuário na base de dados Kerberos. Se o principal existe, o KDC obtém a chave secreta do principal (K_p) e gera duas coisas: uma chave de sessão (K_{s1}) e o TGT, que contém uma cópia da chave de sessão.

2. O KDC criptografa o TGT com outra chave conhecida apenas pelo próprio KDC (K_{kdc}) e criptografa a chave de sessão e o TGT com a chave secreta do usuário, enviando-os para o cliente.



3. Ao receber essa resposta, o cliente (`kinit` ou `login`) solicita a senha ao usuário e após a digitação da senha ela é convertida na chave secreta do usuário (K_p). Essa chave é, então, utilizada para descriptografar a mensagem recebida do KDC, que contém a chave de sessão (K_{s1}) e o TGT (criptografado pela chave do KDC).

4. A chave de sessão e o TGT são armazenados no *cache* de credenciais que é um arquivo temporário na máquina local.

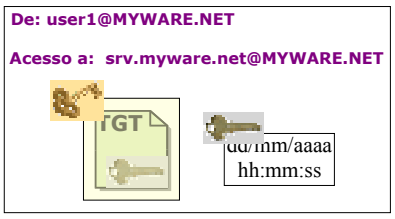
A partir de então o usuário está apto a solicitar novos *tickets* para acesso a serviços na rede Kerberos usando o seu TGT. Para isso ele apresenta o TGT ao KDC que irá gerar um novo *ticket* para o usuário, de modo que ele possa acessar o serviço solicitado. É possível perceber que a senha não foi transmitida pela rede em nenhum momento.

3.2.2 Acessando um servidor

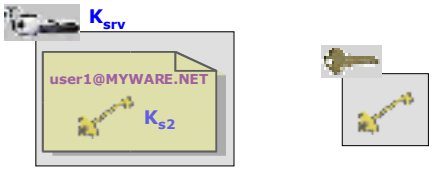
Dada a suposição que o usuário deseja efetuar um *logon* remoto usando o Kerberos, executando o programa Kerberos `rlogin` para acessar uma máquina na rede, dá-se início aos seguintes passos:

1. O usuário executa o cliente: `rlogin nome_do_host`
2. O programa `rlogin` verifica no *cache* de credenciais se há um *ticket* para acesso ao servidor solicitado. Se não há, envia ao KDC uma solicitação de acesso ao servidor desejado, que contém, além da solicitação em si, o TGT

(ainda criptografado pela chave do KDC) e uma mensagem com informações de horário (*timestamp*), criptografada com a chave de sessão.



3. O KDC descriptografa o TGT recebido usando sua própria chave secreta (K_{kdc}) e, usando a chave de sessão contida no TGT (K_{s1}), descriptografa a mensagem de *timestamp* recebida para validar o usuário. Assim, como foi possível ler a mensagem com a respectiva chave de sessão, o KDC sabe que o usuário é quem diz ser.
4. O KDC, então, gera uma nova chave de sessão (K_{s2})que será compartilhada entre o usuário e o servidor que ele pretende utilizar. Ele cria um novo *ticket* (TGS) contendo essa nova chave de sessão e a identificação do usuário e criptografa-o com a chave do servidor (K_{srv}). O KDC também criptografa uma cópia da nova chave de sessão com a chave de sessão que ele recebeu no TGT do usuário ($K_{s1}[K_{s2}]$).



5. O novo *ticket* (criptografado com a senha do servidor) e a nova chave de sessão (criptografada com a chave de sessão do usuário) são enviados ao usuário.
6. O cliente `rlogin` recebe o *ticket* (TGS) e a chave de sessão (K_{s2}) enviados pelo KDC, descriptografa a chave de sessão (pois ele possui a chave K_{s1}) e armazena ambos no *cache* de credenciais. Ele usa a nova chave de sessão para criptografar uma mensagem (M1), efetua a conexão ao servidor

pretendido e envia a ele essa mensagem e o *ticket* do servidor (TGS) ainda criptografado (recebido do KDC e que está em *cache*) .

7. O *deamon* do `rlogin` no servidor (`klogind`) obtém a sua própria chave secreta (K_{srv} – a qual está segura no disco local, no arquivo `keytab`) e a usa para descriptografar o *ticket* (TGS) recebido do usuário, tendo acesso, portanto, à identidade do usuário e à chave de sessão (K_{s2}) . Com essa chave de sessão o servidor abre a mensagem (M1) recebida do cliente, isso serve para autenticar o usuário pois foi possível usar a chave para abrir a mensagem.
8. Se o usuário tem permissão para efetuar *logon* (a ID do usuário está no `/etc/passwd` ou o principal está no arquivo `.k5login`), o `klogind` efetua o *logon* do usuário. A partir daí, tanto o servidor como o usuário conhecem a nova chave de sessão (K_{s2}), podendo utilizá-la para criptografar o tráfego entre eles.

Uma vez que apenas o servidor e o KDC sabem a chave do servidor, o *deamon* do `rlogin` tem a certeza de que o *ticket* foi criptografado pelo KDC (apesar de tê-lo recebido do usuário) e pode confiar na informação sobre a identidade do usuário contida no *ticket*.

A informação do tempo de expiração também é incluída nos *tickets* para prevenir contra ataques de reenvio, que é a apresentação fraudulenta de um *ticket* capturado na rede, na intenção de se obter uma autenticação. Através do uso de *tickets* criptografados, os quais só podem ser descriptografados se o cliente conhece a chave secreta, a autenticação segura acontece.

Capítulo 4

Instalação e Configuração

A implantação do Kerberos numa rede é um processo que envolve definições diversas. O Kerberos utiliza bases de dados centralizadas, servidores de autenticação, servidores de aplicação e, principalmente, tem relação com a segurança lógica de todo o ambiente da rede. Por isso, sua instalação e configuração, requerem análise e planejamento prévio de todos os elementos a ele relacionados, de forma direta ou indireta.

Neste Capítulo será visto o planejamento do ambiente Kerberos e os passos para instalar e configurar um servidor KDC, além de servidores de aplicação e clientes para utilizar o Kerberos para autenticação. A maioria das distribuições possuem suporte ao Kerberos. A título de informação, este documento encontra-se baseado na distribuição Red Hat Linux[3].

4.1 Planejamento da instalação do Kerberos

Antes de iniciar a instalação propriamente dita, é necessário considerar diversos aspectos, entre eles:

- o nome do reino;
- as portas utilizadas pelo KDC e pelo serviço de administração da base de dados;

3 http://www.redhat.com

- quantidade de servidores KDC (mestre e escravos) necessários;
- os nomes dos servidores KDC;
- como será feita a propagação da base de dados.

Maiores detalhes sobre a instalação e configuração também podem ser encontrados em [BRENNEN (2004)], [FERREIRA (2003)], [MIT (2002) (1)], [Red Hat (2003)], [STANGER; LANE (2002)] e [TUNG (1996)].

4.1.1 Definição do nome do reino

O nome do reino Kerberos pode ser qualquer conjunto de caracteres ASCII. Porém, por convenção e simplicidade, é sugerido que o reino tenha o mesmo nome do domínio DNS, porém em letras maiúsculas. Por exemplo, se seu domínio DNS é "suaempresa.com.br", o nome do reino Kerberos pode ser SUAEMPRESA.COM.BR.

O nome do reino é sensível ao formato, ou seja, maiúsculas e minúsculas são tratadas de forma diferente. Além disso, deve ser único na rede.

4.1.2 Portas para acesso ao KDC e aos serviços administrativos

As portas utilizadas pelo Kerberos para acesso ao KDC e para uso do programa de administração `kadmin` são, por padrão, as portas 88 e 749, respectivamente. Para o uso de portas diferentes é necessário alterar os arquivos `/etc/services` e `krb5.conf` em cada máquina e o arquivo `kdc.conf` em cada servidor KDC.

Mais detalhes sobre a configuração e uso dos arquivos `krb5.conf` e `kdc.conf` serão apresentados mais adiante.

4.1.3 Servidores KDC escravos

Um servidor KDC escravo é um servidor KDC que possui uma cópia de leitura da base de dados e que executa a autenticação de principais da mesma forma que o KDC mestre. Os servidores KDC escravos não permitem alterações na base de dados – isto só pode ser feito no KDC mestre. Caso o KDC mestre fique indisponível por um longo tempo, é possível alterar o papel de um KDC escravo, de modo que ele passe a ser o KDC mestre.

O uso de servidores KDC escravos acrescenta redundância para os serviços de fornecimento de *tickets* e autenticação de principais, além de permitir uma divisão das demandas para esses serviços entre os servidores KDC. No caso de falhas ou impossibilidade de acesso ao servidor KDC mestre, o serviço de autenticação não fica interrompido. Para que a autenticação Kerberos possa acontecer é necessário que os clientes tenham acesso a pelo menos um servidor KDC, seja mestre ou escravo.

Algumas sugestões sobre servidores KDC incluem:

- ter pelo menos um KDC escravo, como uma forma de *backup* e tolerância a falhas. Assim, o serviço de autenticação não é interrompido em casos de parada do KDC mestre, seja por problemas físicos, atualizações de *hardware* ou *software* etc., já que os principais continuam sendo autenticados pelo servidor KDC escravo;
- ter um KDC escravo em prédio diferente do KDC mestre, para os casos mais graves de indisponibilidade como quedas de energia ou incêndios.

A definição do número de servidores KDC, bem como sua localização, deve fazer parte do planejamento da instalação do Kerberos e depende de fatores específicos da rede como largura de banda e o número de usuários da rede.

4.1.4 Replicação da base de dados

A base de dados do Kerberos reside no servidor KDC mestre e deve ser replicada para os servidores KDC escravos que existirem (geralmente através do `cron`) de modo a manter as informações de autenticação atualizadas entre esses servidores. Assim, é necessário planejar com que freqüência essa propagação deve ocorrer.

Caso a propagação tome muito tempo para ser concluída até todos os servidores KDC, seja devido a uma base de dados muito grande ou um número elevado de servidores escravos ou mesmo retardos na rede, é possível efetuar a cópia em paralelo, ou seja, replicar a base de dados para um conjunto de servidores escravos e destes para outros conjuntos, reduzindo o tempo total de propagação.

4.1.5 Usando o DNS na configuração

Numa instalação onde há um grande número de máquinas a configurar, em que seria bastante trabalhosa a distribuição e manutenção dos arquivos de configuração do Kerberos, uma opção é utilizar o DNS para fornecer as informações sobre os serviços Kerberos, não mais colocando essas informações nos arquivos de configuração locais.

Mesmo em instalações de pequeno porte essa opção torna-se válida como preparação do ambiente para futuros crescimentos.

Mesmo que não se use o DNS para fornecer configurações do Kerberos, ele é fundamental para a resolução de nomes e funcionamento do Kerberos. Por isso, é importante que o DNS da rede esteja corretamente configurando e com as informações sempre atualizadas, incluindo-se os registros da zona reversa.

Mapeamento entre máquinas e o reino Kerberos

O mapeamento de nomes de máquinas para o reino Kerberos pode ocorrer de duas maneiras. A primeira é a forma tradicional, que utiliza o arquivo de configuração `krb5.conf`. Através de regras específicas contidas nesse arquivo é possível mapear um domínio DNS para o reino Kerberos.

A segunda forma é efetuando uma busca por um registro especial do tipo `TXT` no DNS. Se o cliente está habilitado para essa forma de busca, ele tenta encontrar esse registro no DNS adicionando o prefixo `_kerberos` no nome da máquina. Caso não encontre, o cliente continua a busca adicionando esse prefixo com as diversas possibilidades de domínios existentes no nome até encontrar um registro coincidente, que será então considerada o nome do reino. Assim, um máquina chamada `ftp.df.myware.net` irá procurar pelos seguintes registros TXT no DNS:

```
_kerberos.ftp.df.myware.net
_kerberos.df.myware.net
_kerberos.myware.net
_kerberos.net
_kerberos
```

Mesmo que a opção de uso do DNS para configuração do Kerberos não seja

utilizada, é recomendado que a configuração do DNS seja feita, visando crescimento futuro e interação com outros reinos Kerberos.

Definição de nomes de máquina para os servidores KDC

É recomendado que os servidores KDC tenham nomes definidos como registros CNAME no DNS. Um CNAME é um apelido para um nome de máquina. Por exemplo o KDC mestre poderia se chamar `kerberos` e os servidores KDC escravos, `kerberos1`, `kerberos2`, etc. A vantagem de utilizar registros CNAME é que, caso seja necessário uma troca de máquina (por exemplo, instalar um servidor KDC mais potente) apenas a entrada CNAME precisaria ser alterada sem a necessidade de troca de nomes de máquinas. Dessa forma, o nome do servidor Kerberos fica independente do nome físico da máquina.

Outra forma de localizar os servidores KDC de um reino é através do uso de registros `SRV` no DNS. Esse tipo de registro é suportado pelo Kerberos V5 e por versões mais recentes do DNS. Esses registros indicam o nome da máquina e a porta do serviço. Mais detalhes sobre registros `SRV` podem ser vistos na RFC 2782 [GULBRANDSEN; VIXIE; ESIBOV (2000)].

```
$ORIGIN myware.net.
_kerberos           TXT             "MYWARE.NET"
kerberos            CNAME           cascao.myware.net
kerberos1           CNAME           cebola.myware.net
kerberos2           CNAME           monica.myware.net
_kerberos._udp              SRV     0 0 88  cascao
                            SRV     0 0 88  cebola
                            SRV     0 0 88  monica
_kerberos-master._udp       SRV     0 0 88  cascao
_kerberos-adm._tcp          SRV     0 0 749 cascao
_kpasswd._udp               SRV     0 0 464 cascao
```

Figura 4.1: Exemplo de zona DNS

A Figura 4.1 mostra um exemplo de arquivo de zona DNS com entradas `SRV` para o Kerberos. É importante observar que nos registros `SRV` devem ser usados os nomes das máquinas e não seus apelidos.

A Tabela 4.1 explica os registros `SRV` para o Kerberos utilizados na Figura 4.1.

No decorrer deste documento será utilizada a forma tradicional de configuração,

ou seja, através dos arquivos do próprio Kerberos. O DNS será utilizado unicamente para resolução de nomes.

Tabela 4.1: Registros `SRV` para o Kerberos

Registro	Descrição
`_kerberos._udp`	Define a conexão aos KDCs via UDP. É a conexão mais usada e, geralmente, é definida para porta 88.
`_kerberos-master._udp`	Indica o KDC mestre para os casos de troca de senha recente e o usuário ter acessado um KDC que ainda não está atualizado, aparentando senha incorreta. Nessa situação o KDC mestre é contactado e a mesma senha é utilizada para verificação. Se há apenas um KDC, essa entrada não é necessária.
`_kerberos-adm._tcp`	Indica a porta para conexão do programa `kadmin`, geralmente a porta 749 no KDC mestre. No Kerberos do MIT é necessário configurar a entrada chamada `admin_server` no arquivo `krb5.conf`.
`_kpasswd._udp`	Deve listar a porta 464 do KDC mestre. É a porta usada para troca de senha pelo usuário.

Maiores detalhes sobre o uso do DNS no Kerberos podem ser vistos no "Guia do Administrador do Kerberos V5", do MIT [MIT (2002) (2)].

4.1.6 Sincronização de Horário

Parte do processo de autenticação do Kerberos é baseada no tempo de expiração dos *tickets*, tornando a precisão dos relógios dos servidores e estações um ponto crítico da infra-estrutura. Por isso, o sincronismo dos horários entre as máquinas que utilizam o Kerberos deve estar configurado corretamente ou o Kerberos não irá funcionar. Clientes que tentem se autenticar a partir de uma máquina com relógio fora de sincronismo irão falhar na tentativa de autenticação com o servidor KDC. Uma forma de manter os relógios das máquinas sincronizados é utilizar um protocolo como o *Network Time Protocol* (NTP) .

O NTP é um protocolo para sincronismo de horários entre máquinas. Existem diversos servidores públicos de NTP disponíveis. Esses servidores são divididos em níveis. Os servidores de nível 1 não devem ser usados para sincronismo de clientes pois são em pequeno número e o acesso é restrito aos servidores que compõem a hierarquia do serviço. Os servidores de nível 2 sincronizam seus relógios com os

servidores do nível 1 e estão disponíveis para uso de sincronização pelos clientes. Uma lista de servidores públicos de NTP de nível 2 pode ser obtida no site da RNP[4].

No GNU/Linux, a ativação do NTP se dá pela instalação do pacote NTP e edição do arquivo de configuração, geralmente o `/etc/ntp.conf`. Os valores padrão podem ser mantidos, devendo-se apenas informar os servidores de NTP que serão utilizados para sincronismo do relógio local. A Figura 4.2 mostra um exemplo do arquivo `/etc/ntp.conf`.

```
server  ntps2-1.gnud.ie
server  ntps2-2.gnud.ie

driftfile /etc/ntp/drift
broadcastdelay      0.008

authenticate no
```

Figura 4.2: Exemplo do arquivo `ntp.conf`

O comando de atualização do NTP pode ser adicionado ao `cron` para que seja executado periodicamente, por exemplo:

```
30 * * * * /usr/sbin/ntpdate -s
```

É possível ter uns poucos servidores na rede interna atualizando seus relógios com os servidores públicos de NTP. Os clientes e demais servidores podem ser configurados para atualizar-se a partir desses servidores internos, mesmo que não usem o Kerberos. Mais informações sobre o protocolo NTP e configuração de servidores NTP podem ser obtidas no *site* do *Network Time Protocol Project*[5].

4.2 Instalação do servidor Kerberos

Ao se configurar o Kerberos, o primeiro passo é a instalação do servidor. Considerando que os servidores são dedicados ao serviço de autenticação, além da instalação dos pacotes necessários ao funcionamento do Kerberos, é recomendável seguir alguns passos para manter o servidor o mais seguro possível:

4 http://www.rnp.br/ntp/ntp-stratum2.html

5 http://www.ntp.org

- instalar o sistema operacional e *patches* de segurança;
- não instalar interfaces ou pacotes gráficos;
- opcionalmente, instalar o SSH para administração remota do servidor;
- verificar as portas em uso, desabilitando qualquer porta desnecessária;
- restringir o acesso ao servidor apenas às máquinas que necessitam autenticação. Isso pode ser feito pela edição dos arquivos `/etc/hosts.allow` e `/etc/hosts.deny` ou com `iptables`.

Neste documento não será abordada a configuração de servidores KDC escravos bem como a replicação da base de dados. Se for necessário implementar servidores escravos, detalhes de como configurar os relacionamentos entre o servidor mestre e os escravos podem se encontrados em [BRENNEN (2004)] e no "Guia de Instalação do Kerberos V5", do MIT [MIT (2002) (1)].

4.2.1 Hardware dos servidores

O serviço do Kerberos não requer muito do *hardware* do servidor KDC. Segundo consta em [BRENNEN (2004)], um servidor monoprocessado (um PIII-500) com discos redundantes que possuam capacidade suficiente para armazenamento da base de dados Kerberos e 128 MB de memória RAM, é suficiente para atender cerca de quarenta a cem mil autenticações por dia. Além disso, há a vantagem da utilização de servidores secundários, o que permite a distribuição da carga pela divisão das demandas de autenticação entre o servidor KDC mestre e os escravos.

Uma das recomendações a considerar é que os servidores sejam dedicados ao serviço de autenticação Kerberos, com permissão de *logon* local apenas ao administrador do Kerberos. Isso implica em desativar todos os serviços desnecessários. Apesar do serviço de *shell* remoto (SSH) ser um serviço que pode permanecer ativo para administração remota do servidor, a administração da base de dados do Kerberos, porém, pode ser feita à distância através do programa `kadmin`.

Além das restrições de acesso lógico, é também fundamental implementar restrições de acesso físico ao servidor. Devido às informações que armazena, a segurança dos servidores KDC é de suma importância para a segurança da própria

rede, ou seja, devem ser tomadas todas as precauções para evitar que qualquer servidor KDC seja comprometido.

4.2.2 Pacotes do Kerberos

Os seguintes pacotes são necessários para instalar o servidor Kerberos nas distribuições GNU/Linux:

- `krb5-libs`
- `krb5-client`
- `krb5-server`

Há também pacotes de documentação e desenvolvimento, `krb5-doc` e `krb5-devel`, respectivamente, porém é recomendado que não sejam instalados nos servidores KDC, mas em outras máquinas. Também é possível obter os arquivos para instalação no site do MIT[6].

4.3 Configuração do servidor KDC

Conforme visto no Capítulo 3, os servidores KDC armazenam uma cópia da base de dados do Kerberos e são responsáveis pela distribuição de *tickets*. O KDC mestre possui a cópia principal da base de dados, na qual são feitas as alterações, e propaga essa base aos servidores KDC escravos em intervalos regulares.

Os servidores KDC escravos também efetuam a distribuição de *tickets*, mas neles a base de dados é apenas de leitura, não sendo possível a administração da base de dados a partir de um servidor KDC escravo. Dessa forma, é possível aos clientes continuar efetuando *logon* mesmo que o KDC mestre esteja indisponível. Lembre-se que só há um KDC mestre, podendo haver diversos servidores KDC escravos.

Ao se configurar o servidor Kerberos, os seguintes passos devem seguidos:

1. Assegurar-se de que a sincronização de horário está funcionando no seu servidor e também entre os clientes. Se houver diferenças de horário superior a cinco minutos entre eles, os clientes não serão capazes de se autenticar com

6 http://web.mit.edu/kerberos/dist/index.html

o servidor. Esse valor padrão é configurável e pode ser alterado. O sincronismo de horário é necessário para prevenir ataques com *tickets* antigos capturados na rede. Além disso, o DNS deve estar funcionando adequadamente para permitir a correta resolução de nomes.

2. Instalar os pacotes `krb5-libs`, `krb5-server`, e `krb5-client` na máquina que executará o serviço de KDC. Essa máquina deve estar segura e, se possível, não deve executar mais nenhum serviço além do KDC.
3. Editar os arquivos de configuração do Kerberos `/etc/krb5.conf` e `/var/kerberos/krb5kdc/kdc.conf` de modo a refletir os nomes de reinos e domínios planejados para o seu ambiente. Por convenção, os nomes de reino são escritos em maiúsculas e os de domínios DNS e nomes de máquinas em minúsculas.
4. Criar a base de dados do Kerberos e as contas dos principais.

Os passos 1 e 2 acima foram citados nas sessões 4.1 e 4.2. Os passos 3 e 4 serão melhor detalhados no decorrer deste Capítulo.

4.3.1 O arquivo krb5.conf

O arquivo `krb5.conf` (geralmente localizado no diretório `/etc`) contém as informações de configuração do Kerberos, entre elas a definição dos servidores KDC para o reino e o mapeamento de nomes de domínio para nomes de reinos Kerberos. O arquivo é dividido em seções que contém diversos parâmetros de configuração. A Figura 4.3 apresenta um exemplo do arquivo `/etc/krb5.conf`.

No arquivo de exemplo, a seção `[libdefaults]` define que o reino MYWARE.NET é o reino padrão e que o DNS não será utilizado para busca das configurações do Kerberos. Na seção `[realms]` estão definidos, para o reino MYWARE.NET, os nomes dos servidores KDC, qual deles é o KDC mestre e quais as portas de acesso. A seção `[domain_realm]` faz o mapeamento entre domínios DNS e reinos Kerberos e a seção `[logging]` determina a localização dos arquivos de log do Kerberos. Para mais detalhes sobre o formato e seções do arquivo

/etc/krb5.conf consulte as suas respectivas páginas do *man*[7] ou o "Guia do Administrador do Kerberos V5", do MIT [MIT (2002) (2)].

```
[libdefaults]
      default_realm = MYWARE.NET
      dns_lookup_kdc = false
      dns_lookup_realm = false

[realms]
      MYWARE.NET = {
         kdc = kerberos.myware.net:88
         kdc = kerberos1.myware.net:88
         admin_server = kerberos.myware.net:749
         default_domain = myware.net
      }

[domain_realm]
      .myware.net = MYWARE.NET
      myware.net = MYWARE.NET

[logging]
      kdc = FILE:/var/log/krb5kdc.log
      admin_server = FILE:/var/log/kadmin.log
      default = FILE:/var/log/krb5lib.log
```

Figura 4.3: Exemplo do arquivo krb5.conf

4.3.2 O arquivo kdc.conf

O arquivo kdc.conf contém informações de configuração próprias do servidor KDC, entre elas os padrões usados ao fornecer *tickets*. Geralmente fica localizado no diretório /var/kerberos/krb5kdc/kdc.conf e possui formato semelhante ao do arquivo krb5.conf, com parâmetros de configuração distribuídos em seções. Um exemplo do arquivo kdc.conf é mostrado na Figura 4.4.

No arquivo kdc.conf podem ser definidos diversos parâmetros, como as portas de acesso ao KDC, a localização dos arquivos da base dados, de configuração e de logs, como também os tipos de criptografia suportados. Para mais detalhes sobre o formato e seções do arquivo kdc.conf consulte as suas respectivas

7 man krb5.conf

páginas do *man*[8] ou o "Guia do Administrador do Kerberos V5", do MIT [MIT (2002) (2)].

```
[kdcdefaults]
      kdc_ports = 88
      acl_file = /var/kerberos/krb5kdc/kadm5.acl
      admin_keytab = /var/kerberos/krb5kdc/kadm5.keytab

[realms]
      MYWARE.NET = {
        kdc_ports = 88
        kadmind_port = 749
        key_stash_file = /var/kerberos/krb5kdc/.k5stash
        master_key_type = des3-hmac-sha1
        supported_enctypes = des3-hmac-sha1:normal des-cbc-
crc:normal des-cbc-crc:v4 des-cbc-crc:afs3
      }

[logging]
    kdc = FILE:/usr/local/var/krb5kdc/kdc.log
    admin_server = FILE:/usr/local/var/krb5kdc/kadmin.log
```

Figura 4.4: Exemplo do arquivo `kdc.conf`

4.3.3 Criação da base de dados do Kerberos

O próximo passo na configuração do KDC é a criação da base de dados do Kerberos. Isso é feito através do comando `kdb5_util` executado no servidor KDC mestre. Para inicializar e, ao mesmo tempo, criar a base de dados do Kerberos, o comando `kdb5_util` deve ser executado da seguinte forma:

```
/usr/sbin/kdb5_util create -s
```

A opção `create` cria a base de dados para armazenamento dos principais e das chaves de autenticação para o reino Kerberos. Opcionalmente, pode-se criar um arquivo oculto (chamado *stash file*) que armazena, localmente e de forma criptografada, a senha-mestra do KDC utilizada para autenticação automática na inicialização dos *deamons* do KDC, o `kadmind` e o `krb5kdc`.

O parâmetro `-s` indica que o Kerberos deve criar o *stash file* no qual a senha mestre será armazenada. Sem esse arquivo presente para a leitura da chave, o servidor Kerberos irá solicitar a senha sempre que for iniciado. Por ser um ponto de

8 `man kdc.conf`

ataque caso seja comprometido, o *stash file* deve ser mantido localmente no disco do KDC e protegido com permissão de leitura apenas para o usuário *root*.

No momento da criação, o Kerberos irá solicitar uma senha que será a senha-mestra da base de dados. Essa senha pode ser qualquer cadeia de caracteres, mas deve seguir as regras para composição de senhas seguras. Ao mesmo tempo, não pode ser esquecida, sob pena de não ser mais possível iniciar e administrar o servidor.

A Figura 4.5 mostra um exemplo de criação de uma base de dados Kerberos. Após a execução são criados 5 arquivos no diretório especificado no `kdc.conf` (o diretório padrão é o `/var/kerberos/krb5kdc`): `principal`, `principal.ok`, `principal.kadm5`, `principal.kadm5.lock` e o *stash file*, `.k5stash`.

```
# /usr/local/sbin/kdb5_util create -s
Initializing database '/var/kerberos/krb5kdc/principal' for
realm 'MYWARE.NET',
master key name 'K/M@MYWARE.NET'
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:
#
```

Figura 4.5: Criando a base de dados Kerberos

4.3.4 Criação do arquivo de controle de acesso

O arquivo de controle de acesso deve ser editado para especificar políticas de controle de acesso à base de dados. Deve existir, pelo menos, a política de acesso relativa aos administradores do Kerberos. O arquivo de controle de acesso, geralmente, é o `/var/kerberos/krb5kdc/kadm5.acl` mas sua localização pode ser alterada no arquivo `/var/kerberos/krb5kdc/kdc.conf`.

O `kadm5.acl` é usado pelo `kadmind` para determinar quais principais têm acesso administrativo à base de dados do Kerberos e qual o nível de acesso. O formato do conteúdo do `kadm5.acl` é:

```
principal  permissões [principais_afetados]
```

As permissões são bastante variadas e sua composição pode permitir ou negar uma série de tarefas como a criação de entradas na base de dados, listagem dos principais ou troca de senhas de principais. A permissão "*" dá acesso completo à base de dados do Kerberos.

Um exemplo do `kadm5.acl` pode ser visto na Figura 4.6. De acordo com a figura, o arquivo especifica que todo principal cuja instância termine com */admin* no reino `MYWARE.NET` terá privilégios de acesso completo.

```
*/admin@MYWARE.NET  *
```

Figura 4.6: Exemplo do arquivo `kadm5.acl`

Mais detalhes sobre as permissões no arquivo `kadm5.acl` podem ser vistas no Capítulo 5.

4.3.5 Iniciando os serviços do KDC mestre

Finalmente deve-se iniciar os serviços necessários à execução do KDC. Esses serviços são o `krb5kdc` e o `kadmind`:

```
/usr/sbin/krb5kdc
/usr/sbin/kadmind
```

Sua inicialização automática também pode ser configurada, de acordo com a distribuição utilizada. Para que a inicialização automática seja possível é necessário que o arquivo local com a senha-mestre (*stash file*) tenha sido criado.

Os arquivos de log especificados no arquivo `/etc/krb5.conf` podem ser lidos para verificação da inicialização dos serviços, conforme mostra a Figura 4.7.

```
/var/log/krb5kdc.log :
Feb 07 10:39:02 cascao.myware.net krb5kdc[1222](info):
commencing operation

/var/log/kadmin.log :
Feb 07 10:39:18 cascao.myware.net kadmind[1244](info): starting
```

Figura 4.7: Logs de inicialização do KDC

4.3.6 Adição de administradores à base de dados

O utilitário `kadmin` comunica-se com o `kadmind` através da rede e usa o Kerberos para autenticação. Porém, é necessário criar o primeiro principal antes de utilizar os programas de administração. Para isso, utiliza-se o comando `kadmin.local`, que é específico para utilização no próprio servidor KDC mestre e não utiliza o Kerberos para autenticação. A Figura 4.8 apresenta um exemplo de criação do usuário *admin* como administrador do KDC.

```
# /usr/sbin/kadmin.local -q "addprinc admin/admin"
```

Figura 4.8: Criação do principal "admin"

Mais detalhes sobre as opções do comando `kadmin.local` podem ser vistas no Capítulo 5.

4.3.7 Criação do principal para o servidor

Cada servidor KDC necessita de um principal do tipo *host* na base de dados do Kerberos. Supondo a configuração do arquivo `/etc/krb5.conf` da Figura 4.3, mostrada anteriormente, há um KDC mestre e um KDC escravo. Conforme mostrado na Figura 4.9, para criar as chaves desses servidores pode-se utilizar o programa `kadmin` acessando o KDC mestre, que está executando o *deamon* `kadmind`.

```
# /usr/sbin/kadmin -p admin/admin
Authenticating as principal admin/admin with password.
Enter password:
kadmin: addprinc -randkey host/kerberos.myware.net
WARNING: no policy specified for
"host/kerberos.myware.net@MYWARE.NET"; defaulting to no policy
Principal "host/kerberos.myware.net@MYWARE.NET" created.
kadmin: addprinc -randkey host/kerberos1.myware.net
NOTICE: no policy specified for
"host/kerberos1.myware.net@MYWARE.NET"; defaulting to no policy
Principal "host/kerberos1.myware.net@MYWARE.NET" created.
kadmin:
```

Figura 4.9: Criação do principal para o KDC

É possível associar uma política de senhas quando um principal é criado. Uma política de senhas define aspectos para as senhas dos principais como tamanho mínimo da senha, tempo de expiração da senha, etc. Caso não seja especificado, o Kerberos procura a política padrão. Caso não existam políticas definidas, o Kerberos não poderá associar o principal a nenhuma política. No exemplo da Figura 4.9, não há políticas definidas na base de dados do Kerberos. Mais sobre políticas de senhas pode ser visto no Capítulo 5.

4.3.8 Criação do arquivo de chave para o KDC (*keytab*)

Cada servidor KDC (mestre ou escravo) precisa de uma chave para descriptografar *tickets* (vide seção 3.2.2, ítem 7). Isso é feito através do programa `kadmin` executado *no próprio servidor* para o qual se quer gerar a chave. A chave é lida da base de dados do Kerberos e armazenada localmente no arquivo `/etc/krb5.keytab`, conforme ilustrado na Figura 4.10.

```
kadmin: ktadd host/kerberos.myware.net
Entry for principal host/kerberos.myware.net with kvno 3,
encryption type Triple DES cbc mode with HMAC/sha1 added to
keytab WRFILE:/etc/krb5.keytab.
kadmin: quit
#
```

Figura 4.10: Criação da chave para o KDC

Para que seja possível a criação do arquivo de chave deve existir um principal para o servidor na base de dados do Kerberos. Os passos da Figura 4.10 teriam que ser executados localmente nos demais servidores KDC, se houverem.

O arquivo de chave é um arquivo criptografado, armazenado localmente e que contém a lista de chaves da máquina que funcionam de forma semelhante à senha de um usuário. Do mesmo modo que um usuário deve manter a senha em segurança, é importante manter segura a lista de chaves de uma máquina. Assim como o *stash file*, o arquivo *keytab* deve ter permissão de leitura apenas para o usuário *root* e deve existir unicamente no disco local.

Listando o conteúdo do arquivo *keytab*

É possível listar as chaves contidas no arquivo *keytab* utilizando o comando `klist`, como mostrado na Figura 4.11, onde é possível visualizar a chave criada anteriormente.

```
# klist -k
Keytab name: /etc/krb5.keytab
KVNO Principal
---- --------------------------------------------
   3 host/kerberos.myware.net@MYWARE.NET
#
```

Figura 4.11: Listando o arquivo `krb5.keytab`

4.3.9 Criando as contas dos principais

Uma vez que o servidor KDC está operacional é possível utilizar o `kadmin` para criar as entradas para principais de usuários, máquinas e serviços na base de dados do Kerberos. A Figura 4.12 mostra a criação do usuário *user1*.

Mais detalhes sobre a criação e manutenção de principais serão apresentados no Capítulo 5.

```
# kadmin -q "addprinc user1" -p admin/admin
Authenticating as principal admin/admin with password.
Enter password:
WARNING: no policy specified for "user1@MYWARE.NET"; defaulting
to no policy
Enter password for principal "user1@MYWARE.NET":
Re-enter password for principal "user1@MYWARE.NET":
Principal "user1@MYWARE.NET" created.
#
```

Figura 4.12: Criação de principal para usuário

4.3.10 Testando a funcionalidade do servidor KDC

A partir de então é possível verificar se o servidor KDC consegue distribuir *tickets*. Basta executar o comando `kinit` para obter um *ticket* e armazená-lo no *cache* de credenciais.

Por padrão, o `kinit` tenta se autenticar utilizando o usuário logado. Como, nestes exemplos, o usuário logado não é o mesmo principal que foi cadastrado na seção 4.3.6, é necessário informar ao `kinit` o nome do principal, da seguinte forma:

```
kinit admin/admin
```

É possível, então, utilizar o comando `klist` para visualizar a lista de credenciais armazenadas no *cache*, conforme Figura 4.13, e o comando `kdestroy` caso se deseje apagar o *cache* de credenciais.

```
# klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: admin/admin@MYWARE.NET

Valid starting   Expires           Service principal
02/07/05 14:42:55  02/08/05 00:42:55    krbtgt/MYWARE.NET@MYWARE.NET
#
```

Figura 4.13: Listando o arquivo de credenciais

4.4. Instalação e configuração de máquinas clientes

A configuração de clientes para uso do Kerberos é mais simples que configurar um servidor KDC. Basicamente, é necessário instalar os pacotes de cliente, que são o `krb5-libs`, o `krb5-client` e o `krb5-apps-clients`, e configurar o Kerberos através do arquivo `/etc/krb5.conf` que pode ser o mesmo usado no servidor KDC.

Além disso, é necessário assegurar-se que a sincronização de horário e a resolução de nomes através do DNS estão funcionando adequadamente na máquina cliente.

4.4.1 Programas clientes

Alguns dos programas "kerberizados" instalados pelos pacotes são `krlogin`, `ktelnet`, `kftp`, `krcp`, `krsh`, `ksu`, `kinit`, `klist`, `kdestroy` e `kpasswd`,

localizados em `/usr/bin`, e o `login.krb5` localizado no diretório `/usr/sbin`.

Pode ser necessário alterar configurações nos clientes para que esses programas sejam utilizados no lugar das versões padrão, por exemplo, alterando a variável $PATH ou criando *alias* ou *links*. É recomendável que o `login.krb5` seja utilizado no lugar do `/bin/login` de modo que os usuários não precisem executar o programa `kinit` para obter o TGT.

Caso não sejam usados *links* ou alteradas variáveis de ambiente é necessário educar os usuários a utilizar os programas "kerberizados" como, por exemplo, `kftp` e `kpasswd` no lugar do `ftp` e `passwd`, respectivamente. Além disso, os usuários precisam se habituar a utilizar programas de gerenciamento de *tickets* como `kinit`, `klist` e `kdestroy`.

As versões "kerberizadas" mantêm as características dos programas originais correspondentes, adicionando a característica de autenticação transparente, pelo uso dos *tickets*, e a opção de uso de criptografia na comunicação com o servidor remoto. Na maioria das vezes a única característica visível é que a senha não é solicitada quando há a conexão ao serviço remoto uma vez que a autenticação é feita pelo Kerberos. Mais sobre as opções de uso das versões dos programas podem ser vistos no "Guia do usuário do Kerberos V5", do MIT [MIT (2002) (3)].

4.4.2 Criação do principal e da chave para o cliente

Antes que uma máquina cliente possa obter *tickets* e utilizar serviços remotos é necessário criar um principal para essa máquina através do programa `kadmin`. O primário do principal é a palavra *host* e a instância é o nome da máquina. A Figura 4.14 mostra a criação do principal para a máquina *cliente1*.

```
# kadmin -q "addprinc -randkey host/cliente1.myware.net"
WARNING: no policy specified for
"host/servidor1.myware.net@MYWARE.NET";
defaulting to no policy
Principal "host/cliente1.myware.net@MYWARE.NET" created.
#
```

Figura 4.14: Criação de principal de máquina cliente

Depois de criado o principal na base de dados do Kerberos, é necessário armazenar localmente a chave da máquina cliente, executando, *na própria máquina cliente*, o programa `kadmin`, com o comando `ktadd`, como apresentado na Figura 4.15.

```
# kadmin -q "ktadd host/cliente1.myware.net"
Entry for principal telnet/servidor1.myware.net with kvno 3,
encryption type Triple DES cbc mode with HMAC/sha1 added to
keytab WRFILE:/etc/krb5.keytab.
#
```

Figura 4.15: Criação da chave para máquina cliente

Por padrão, o arquivo de chave é o `/etc/krb5.keytab` e deve ser mantido seguro, da mesma forma que para o servidor KDC (vide seção 4.3.8).

4.5. Servidores de aplicação

Um servidor de aplicação é uma máquina que provê serviços para a rede. Num ambiente Kerberos é possível ter os servidores executando os *deamons* dos serviços "kerberizados", autenticando no servidor KDC os usuários que tentem utilizar seus serviços. Caso existam clientes sem o Kerberos instalado é possível ainda utilizar o Kerberos para *logon* na rede, porém os serviços terão que aceitar conexões anônimas ou autenticações locais.

A instalação de um servidor de aplicação com Kerberos requer a instalação dos pacotes `krb5-libs`, `krb5-client` e `krb5-apps-servers`, e a configuração do Kerberos através do arquivo `/etc/krb5.conf`, de modo semelhante à configuração do cliente.

Da mesma forma, é necessário assegurar-se que a sincronização de horário e a resolução de nomes através do DNS estão funcionando adequadamente na máquina servidora.

Além disso, é necessário adicionar o servidor e seus serviços à base de dados do

Kerberos e gerar o arquivo com as chaves (*keytab*) para o servidor e para os serviços.

4.5.1 Programas de servidor

Assim como há os programas para clientes, o Kerberos possui versões "kerberizadas" de programas para o servidor, entre eles, `kftpd`, `klogind`, `kshd`[9], e `ktelnetd`. Esse programas são instalados no diretório `/usr/sbin`.

Supondo um servidor que irá prover os serviços `ftp` e `telnet` baseados no Kerberos, e utilizando o `inetd`, a Figura 4.16 mostra as alterações necessárias no arquivo `/etc/inetd.conf`, para aceitar conexões autenticadas pelo Kerberos.

As linhas para os serviços `ftp` e `telnet` existentes no arquivo `/etc/inetd.conf` devem ser comentadas e as linhas da Figura 4.16 adicionadas.

```
ftp     stream tcp nowait root /usr/sbin/kftpd kftpd -a
telnet  stream tcp nowait root /usr/sbin/ktelnetd ktelnetd -a
valid
```

Figura 4.16: Configuração do `inetd.conf` para serviços

Assim, são desabilitados os serviços padrão `ftp` e `telnet`, e passam a ser usados os programas do Kerberos. Mais sobre as opções para as versões dos programas de servidor podem ser vistas no "Guia do Administrador do Kerberos V5", do MIT [MIT (2002) (2)].

4.5.2 Criação do principal e da chave para o servidor e serviços

Todo servidor de aplicações Kerberos precisa de uma chave para o servidor e para os seus serviços. Para isso é necessário criar o arquivo de chave `/etc/krb5.keytab` para permitir a autenticação do servidor e dos serviços no KDC. Para que o arquivo seja criado é necessário que existam os principais do servidor e dos serviços na base de dados do Kerberos.

9 Os programas `klogind` e `kshd` são as versões "kerberizadas" do servidor de *login* remoto e o servidor de *shell* remoto, respectivamente.

O arquivo de chave deve ser mantido seguro, da mesma forma que para o servidor KDC (vide seção 4.3.8).

A criação dos principais da máquina e dos serviços é feita através do programa `kadmin`. A Figura 4.17 apresenta a criação do principal para a máquina *servidor1* e para os serviços `ftp` e `telnet` no mesmo servidor.

```
# kadmin -q "addprinc -randkey host/servidor1.myware.net"
WARNING: no policy specified for
"host/servidor1.myware.net@MYWARE.NET";
defaulting to no policy
Principal "host/servidor1.myware.net@MYWARE.NET" created.
# kadmin -q "addprinc -randkey ftp/servidor1.myware.net"
WARNING: no policy specified for
"ftp/servidor1.myware.net@MYWARE.NET";
defaulting to no policy
Principal "ftp/servidor1.myware.net@MYWARE.NET" created.
# kadmin -q "addprinc -randkey telnet/servidor1.myware.net"
WARNING: no policy specified for
"telnet/servidor1.myware.net@MYWARE.NET";
defaulting to no policy
Principal "telnet/servidor1.myware.net@MYWARE.NET" created.
#
```

Figura 4.17: Criação de principais para servidor de aplicação

Depois de criados os principais na base de dados do Kerberos, é necessário criar o arquivo com as chaves. Isso é feito no *próprio servidor* de aplicação com o programa `kadmin` utilizando o comando `ktadd`. A Figura 4.18 apresenta a criação do arquivo de chaves para os principais da Figura 4.17.

```
# kadmin -q "ktadd host/servidor1.myware.net
ftp/servidor1.myware.net telnet/servidor1.myware.net"
Entry for principal host/servidor1.myware.net with kvno 3,
encryption type Triple DES cbc mode with HMAC/sha1 added to
keytab WRFILE:/etc/krb5.keytab.
Entry for principal ftp/servidor1.myware.net with kvno 3,
encryption type Triple DES cbc mode with HMAC/sha1 added to
keytab WRFILE:/etc/krb5.keytab.
Entry for principal telnet/servidor1.myware.net with kvno 3,
encryption type Triple DES cbc mode with HMAC/sha1 added to
keytab WRFILE:/etc/krb5.keytab.
#
```

Figura 4.18: Criação de chaves para serviços

4.6 Exemplo de uso de aplicação

Supondo o principal *user1@MYWARE.NET* criado na seção 4.3.9 efetuando *logon* na estação *cliente1.myware.net* e acessando o serviço de `ftp` no servidor *servidor1.myware.net*, pode-se apresentar situação semelhante à apresentada na Figura 4.19.

```
cliente1 login: user1
Password for user1:
Linux 2.4.20.
user1@cliente1:~$ /usr/bin/kftp servidor1.myware.net
Connected to servidor1.myware.net.
220 servidor1 FTP server ready.
334 Using authentication type GSSAPI; ADAT must follow
GSSAPI accepted as authentication type
GSSAPI authentication succeeded
Name (servidor1.myware.net:user1):
232 GSSAPI user user1@MYWARE.NET is authorized as user1
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Figura 4.19: Acesso a serviço com autenticação Kerberos

Nesse exemplo é também suposto que o programa `/usr/sbin/login.krb5` está sendo executado no lugar do `/bin/login` padrão. Desse modo, o usuário não necessita executar o comando `kinit` para obter seus *tickets*.

A Figura 4.20 apresenta o resultado da execução do comando `klist` para exibir os *tickets* do usuário. O usuário possui 4 *tickets*: o seu TGT e o *ticket* de acesso à estação *cliente1.myware.net*, obtidos após o *logon*, o *ticket* de acesso ao *servidor1.myware.net*, obtido após a conexão ao servidor, e o *ticket* de acesso ao serviço `ftp`, obtido após a autenticação no serviço. A opção `-5` especifica que apenas os *tickets* da versão 5 do Kerberos devem ser listados.

É importante ressaltar que o Kerberos armazena apenas as informações para autenticação dos principais. Assim, a conta local do usuário continua sendo necessária para que o usuário efetue *logon*, uma vez que o Linux precisa obter outras informações do usuário como grupos, número do ID para configuração das permissões locais de acesso a arquivos, *shell* padrão, diretório *home* etc.

```
user1@cliente1:~$ klist -5
Ticket cache: FILE:/tmp/krb5cc_p334
Default principal: user1@MYWARE.NET

Valid starting  Expires      Service principal
02/08/05 18:23:06 02/09/05 04:23 krbtgt/MYWARE.NET@MYWARE.NET
              renew until 02/09/05 18:22:55
02/08/05 18:23:06 02/08/05 18:28 host/cliente1.myware.net@MYWARE.NET
              renew until 02/09/05 18:22:55
02/08/05 18:24:39 02/08/05 18:29 host/servidor1.myware.net@MYWARE.NET
              renew until 02/09/05 18:22:55
02/08/05 18:24:44 02/09/05 04:23 ftp/servidor1.myware.net@MYWARE.NET
              renew until 02/09/05 18:22:55
user1@cliente1:~$
```

Figura 4.20: Listando os *tickets* do usuário

Ou seja, o Kerberos permite a autenticação centralizada, mas o Linux precisa de outras informações para permitir o acesso local. Essas informações, que normalmente são armazenadas no arquivo `/etc/passwd` de cada máquina, podem também ser centralizadas. Uma opção para ter essas informações de forma centralizada é o uso de serviços como o *Name Service Switch* (NSS).

Com o NSS, o administrador pode centralizar as informações de *logon* do usuário e manter o serviço de autenticação no Kerberos, com o qual é possível o *logon* único, ou seja, o usuário digita a senha uma única vez e tem acesso aos serviços "kerberizados" sem necessidade de redigitar senhas ou ter senhas diferentes para os vários serviços.

Capítulo 5

Administração do Kerberos

A base de dados do Kerberos contém os principais, suas senhas e outras informações administrativas necessárias para funcionamento de um reino. Grande parte das alterações das informações contidas na base de dados do Kerberos é feita através do programa `kadmin`, com o qual é efetuada a manutenção dos principais, gerenciadas as políticas e arquivos de chaves. Outro programa que altera informações da base de dados é o `kpasswd`, utilizado pelos usuários para troca de senha.

Para administração da base de dados do Kerberos é utilizado o programa `kdb5_util`, que manipula a base de dados como um todo, por exemplo para criação, remoção ou cópia dos arquivos que compõem a base de dados do Kerberos.

Do ponto de vista do usuário, o gerenciamento dos *tickets* e senhas dos usuários é outro fator importante na administração. Este Capítulo apresenta opções de comando e exemplos de uso dos programas para gerenciamento da base de dados, por parte do administrador, e do gerenciamento de *tickets* e senhas, por parte dos usuários.

5.1 Controle de acesso

O controle de acesso e permissões administrativas à base de dados do Kerberos é feito através de entradas no arquivo `kadm5.acl`. Como mostrado na seção 4.3.4, o formato do conteúdo do `kadm5.acl` é:

```
principal  permissões [principais_afetados]
```

Os campos `principal` e `principais_afetados` podem incluir um asterisco (*), de modo a generalizar a permissão dada. A permissão é dada de acordo com a primeira entrada que for coincidente. A Figura 5.1 apresenta um exemplo de um arquivo de permissões e a Tabela 5.1 as permissões possíveis. As permissões são representadas por letras minúsculas. Letras maiúsculas indicam negação de permissão.

```
*/admin@MYWARE.NET   *
root@MYWARE.NET      ADMCIL
```

Figura 5.1: Exemplo do arquivo `kadm5.acl`

Tabela 5.1: Permissões de controle de acesso à base de dados

Opção	Descrição
a / A	Permite / impede a adição de principais ou políticas.
d / D	Permite / impede a remoção de principais ou políticas.
m / M	Permite / impede a modificação de principais ou políticas.
c / C	Permite / impede a modificação de senhas de principais.
i / I	Permite / impede consultar a base de dados.
l / L	Permite / impede listar principais ou políticas.
* ou x	Todos os privilégios (admcil).

No exemplo da Figura 5.1, qualquer principal com a instância *admin* tem acesso completo à base de dados. O principal *root* não tem privilégios. Mais informações sobre o arquivo `kadm5.acl` podem ser vistas no "Guia do Administrador do Kerberos V5", do MIT [MIT (2002) (2)].

5.2 O programa `kadmin`

O programa `kadmin` é utilizado para administração remota do KDC mestre e geração do arquivo de chaves (*keytab*) e existe tanto no servidor Kerberos como nos clientes e servidores de aplicações. Há também uma versão local do programa, chamado `kadmin.local` que é usado diretamente no KDC, principalmente para

as configurações iniciais de preparação e *startup* dos serviços, após as quais é possível efetuar a administração de forma remota. A Tabela 5.2 apresenta as principais opções para o `kadmin`.

Tabela 5.2: Opções do `kadmin`

Opção	Descrição
-r REINO	Utiliza o reino especificado como reino padrão. Também pode ser usado com o `kadmin.local`.
-p principal	Usa o principal especificado para autenticação. Também pode ser usado com o `kadmin.local`.
-q comando	Passa comandos diretamente ao programa. Também pode ser usado com o `kadmin.local`.
-w senha	Utiliza a senha especificada ao invés de solicitar a entrada pelo teclado.
-s servidor[:porta]	Especifica o servidor e a porta para conexão.
-m	Utilizado apenas com o `kadmin.local`. Solicita a senha-mestra da base de dados.

Mais detalhes sobre os programas `kadmin` e `kadmin.local` e as suas opções de comando podem ser vistas no "Guia do Administrador do Kerberos V5", do MIT [MIT (2002) (2)].

5.3 Políticas

Uma política é um conjunto de regras sobre senhas. Políticas podem determinar o tempo máximo e mínimo de duração de senhas, o tamanho mínimo e tipo de caracteres que uma senha deve conter, além do tamanho do histórico de senhas de um principal na base de dados.

5.3.1 Criação e alteração de políticas

Para criar uma política é utilizado o comando `add_policy` (ou `addpol`) do programa `kadmin`. Esse comando necessita da permissão de adição (“a”) e sua sintaxe é

```
add_policy [opções] política
```

A alteração de configuração de uma política requer a permissão de modificação ("m") e é feita com o comando `modify_policy` (ou `modpol`) do programa `kadmin`. A sintaxe é idêntica à do comando `add_policy`.

As opções mais comuns para os dois comandos estão listadas na Tabela 5.3. A Figura 5.2 mostra um exemplo de criação de uma política.

Tabela 5.3: Opções para os comandos `add_policy` e `modify_policy`

Opção	Descrição
-maxlife *tempo*	Configura o tempo de vida máximo para senhas.
-minlife *tempo*	Configura o tempo de vida mínimo para senhas.
-minlength *tam*	Especifica o comprimento mínimo de caracteres de uma senha.
-minclasses *num*	Requer, no mínimo, *num* classes de caracteres na senha.
-history *num*	Especifica o número de senhas antigas a serem mantidas como histórico para um principal.

```
# kadmin
Authenticating as principal root/admin@MYWARE.NET with
password.
Password for root/admin@MYWARE.NET:
kadmin:  addpol -maxlife "10 day" -minlength 7 -history 5 adm-
pol
kadmin:
```

Figura 5.2: Criando uma política

5.3.2 Listando as políticas

Para listar as políticas existentes na base de dados é usado o comando `list_policies` do programa `kadmin`, que requer o privilégio administrativo de listar a base de dados ("l"). Sua sintaxe é

```
list_policies [expressão]
```

onde `expressão` pode conter caracteres para limitar a pesquisa, de modo semelhante a alguns comandos do *shell* do Linux. Outras formas do comando são

`listpols`, `get_policies` e `getpols`. A Figura 5.3 apresenta um exemplo de uso do comando.

```
kadmin:  listpols
adm-pol
polcomum
user-pol
kadmin:
```

Figura 5.3: Listando políticas

5.3.3 Obtendo a configuração de uma política

Para ver a configuração de uma política específica utiliza-se o comando `get_policy` (ou `getpol`) que necessita do privilégio de consulta à base de dados ("i"). A sua sintaxe é

```
get_policy política
```

e a Figura 5.4 mostra a configuração de uma das políticas listadas na Figura 5.3.

A informação *reference count* representa o número de principais que estão utilizando essa política.

```
kadmin:  getpol adm-pol
Policy: adm-pol
Maximum password life: 864000
Minimum password life: 0
Minimum password length: 7
Minimum number of password character classes: 1
Number of old keys kept: 5
Reference count: 0
kadmin:
```

Figura 5.4: Listando a configuração de uma política

5.3.4 Remoção de políticas

A remoção de uma política é feita com o comando `delete_policy` (ou `delpol`) do programa `kadmin`, o qual requer a permissão administrativa de remoção ("d") na base de dados.

Antes de remover uma política é necessário que não exista nenhum principal associado a ela. A sintaxe do comando é

```
delete_policy [-force] política
```

onde a opção `-force` remove a política sem solicitar confirmação. A Figura 5.5 mostra a remoção de uma política.

```
kadmin:  delpol polcomum
Are you sure you want to delete the policy "polcomum"?
(yes/no): yes
kadmin:
```

Figura 5.5: Exclusão de uma política

5.4 Administração de Principais

A administração de principais é um processo simples, porém, bastante comum dentro da base de dados e, provavelmente, a atividade que mais tomará tempo do administrador. Entre as tarefas de administração estão a adição, alteração e remoção de principais e a alteração de senha de um principal.

5.4.1 Criação e modificação de principais

Para adicionar um principal ao Kerberos utiliza-se o comando `add_principal` (ou `addprinc`) do programa `kadmin`. Esse comando necessita da permissão de adição ("a") e sua sintaxe é

```
add_principal [opções] principal
```

Ao ser executado ele solicita uma senha para o novo principal.

A alteração de atributos de um principal requer a permissão de modificação ("m") e é feita com o comando `modify_principal` (ou `modprinc`) do programa `kadmin`. A sintaxe é idêntica à do comando `add_principal`.

As opções mais comuns para os dois comandos estão listadas na Tabela 5.4. A Figura 5.6 apresenta um exemplo de uso do comando `add_principal`.

Tabela 5.4: Opções dos comandos `add_principal` e `modify_principal`

Opção	Descrição
-expire *data*	Configura a data de expiração do principal para *data*.
-pwexpire *data*	Configura a data de expiração da senha para *data*.
-maxlife *max*	Configura o tempo de expiração do *ticket* para *max*.
-maxrenewlife *max*	Configura o tempo de renovação do *ticket* para *max*.
-policy *pol*	Especifica uma política para o principal.
-clearpolicy	Remove a política existente ou cria um principal sem política associada.
{-|+}needchange	"+needchange" força o usuário a trocar a senha ao efetuar *logon*. "-needchange" retira essa obrigação. O padrão é "-needchange".
-randkey	Cria uma chave aleatória para o principal (usado apenas com o comando `add_principal`). É recomendada na criação de principais de máquinas (principal do tipo *host)*.
-pw *senha*	Configura a senha para o principal sendo criado e, assim, não solicita a senha no momento da criação.

```
kadmin:  addprinc -policy user-pol user3
Enter password for principal "user3@MYWARE.NET":
Re-enter password for principal "user3@MYWARE.NET":
Principal "user3@MYWARE.NET" created.
kadmin:
```

Figura 5.6: Uso do comando `add_principal`

5.4.2 Listando principais

É possível listar os principais cadastrados no Kerberos com o comando `list_principals` do programa `kadmin`. Sua sintaxe é

```
list_principals [expressão]
```

onde `expressão` pode conter caracteres para limitar a pesquisa, de modo semelhante a alguns comandos do *shell* do Linux. Se nenhuma expressão for passada todos os principais são listados. O mesmo comando possui outros nomes que podem ser usados no seu lugar: `listprincs`, `get_principals` e

`getprincs`. Para ser utilizado é necessária permissão para listar ("l") a base de dados. A Figura 5.7 apresenta um exemplo de uso do comando.

```
kadmin:  listprincs
K/M@MYWARE.NET
admin/admin@MYWARE.NET
ftp/servidor1.myware.net@MYWARE.NET
host/kerberos.myware.net@MYWARE.NET
host/cliente1.myware.net@MYWARE.NET
host/servidor1.myware.net@MYWARE.NET
kadmin/admin@MYWARE.NET
kadmin/changepw@MYWARE.NET
kadmin/history@MYWARE.NET
krbtgt/MYWARE.NET@MYWARE.NET
root/admin@MYWARE.NET
root@MYWARE.NET
telnet/servidor1.myware.net@MYWARE.NET
user1@MYWARE.NET
user2@MYWARE.NET
user3@MYWARE.NET
kadmin:
```

Figura 5.7: Listando principais

5.4.3 Obtendo informações de um principal

Para obter informações sobre determinado principal cadastrado na base de dados do Kerberos é necessária a permissão de consultar a base de dados ("i") e utiliza-se o comando `get_principal` do programa `kadmin`. A Figura 5.8 apresenta um exemplo do uso do comando.

5.4.4 Remoção de principais

A exclusão de principais é feita com o comando `delete_principal` (ou `delprinc`) do programa `kadmin`, o qual requer a permissão de remoção ("d") na base de dados.

A sintaxe do comando é

```
delete_principal [-force] principal
```

onde a opção `-force` remove o principal sem solicitar confirmação. A Figura 5.9 mostra a remoção de um principal.

```
kadmin:  getprinc user3
Principal: user3@MYWARE.NET
Expiration date: [never]
Last password change: Wed Feb 09 02:49:44 BRST 2005
Password expiration date: Fri Mar 11 01:49:44 BRT 2005
Maximum ticket life: 0 days 10:00:00
Maximum renewable life: 7 days 00:00:00
Last modified: Wed Feb 09 02:49:44 BRST 2005
(root/admin@MYWARE.NET)
Last successful authentication: [never]
Last failed authentication: [never]
Failed password attempts: 0
Number of keys: 6
Key: vno 1, Triple DES cbc mode with HMAC/sha1, no salt
Key: vno 1, DES cbc mode with CRC-32, no salt
Key: vno 1, DES cbc mode with RSA-MD5, Version 4
Key: vno 1, DES cbc mode with RSA-MD5, Version 5 - No Realm
Key: vno 1, DES cbc mode with RSA-MD5, AFS version 3
Attributes:
Policy: user-pol
kadmin:
```

Figura 5.8: Listando informações de um principal

```
kadmin:  delprinc user3
Are you sure you want to delete the principal
"user3@MYWARE.NET"? (yes/no): yes
Principal "user3@MYWARE.NET" deleted.
Make sure that you have removed this principal from all ACLs
before reusing.
kadmin:
```

Figura 5.9: Exclusão de um principal

5.4.5 Alteração de senha

O administrador do Kerberos pode alterar a senha de um principal qualquer, o que requer a permissão de modificação (“m”). Caso a alteração seja da senha do seu próprio principal o privilégio de alteração de senha (“c”) é suficiente. A alteração de senha de principais é feita com o comando `change_password` (ou `cpw`) do programa `kadmin`, e a sintaxe do comando é

```
change_password [opções] principal
```

A Tabela 5.5 apresenta as opções do comando `change_password` e a Figura 5.10 um exemplo de uso do mesmo.

Tabela 5.5: Opções para o comandos `change_password`

Opção	Descrição
-randkey	Atribui um valor aleatório à senha do principal.
-pw *senha*	Atribui diretamente a cadeia de caracteres *senha* ao principal.

```
kadmin: cpw user2
Enter password for principal user2@MYWARE.NET:
Re-enter password for principal user2@MYWARE.NET:
Password for user2@MYWARE.NET changed.
kadmin:
```

Figura 5.10: Alteração de senha de um principal pelo administrador

5.4.6 Principais para máquinas

Quando se instalam programas do Kerberos em um servidor de aplicações é necessário criar um principal do tipo *host* na base de dados do Kerberos para esse servidor e um principal para cada serviço. Além disso, deve-se criar um arquivo local no servidor que contém a lista de chaves da máquina. Esse arquivo local é chamado *keytab* e funciona para a autenticação do servidor no KDC, de modo semelhante à senha de um usuário (vide seção 4.5).

A criação do arquivo de chaves é feita com o `kadmin` e deve ser executado localmente na própria máquina onde residirá. Caso isso não seja possível, pode ser gerado em outra máquina e copiado, de forma segura, para o local correto.

Para criação do arquivo de chaves, ou adição de um principal ao mesmo, utiliza-se o comando `ktadd` do programa `kadmin`, o qual requer a permissão de consulta ("i") na base de dados. A sintaxe do comando é

```
ktadd [-k[eytab] keytab] [principal |
-glob princ_exp] [...]
```

A Tabela 5.6 apresenta as principais opções do comando `ktadd`. Um exemplo de criação e listagem do conteúdo do arquivo de chaves pode ser visto na Figura 5.11.

Tabela 5.6: Opções do `ktadd`

Opção	Descrição
-k[eytab] *keytab*	Usa *keytab* como arquivo de chaves. Se não for especificado, será usado o arquivo padrão (`/etc/krb5.keytab`).
principal | -glob *princ_exp*	Adiciona ao arquivo de chaves o principal *principal*, ou todos aqueles que casam com a expressão *princ_exp*. A opção -glob *princ_exp* requer o privilégio de listar a base de dados ("l").

```
kadmin: ktadd host/kerberos.myware.net
Entry for principal host/kerberos.myware.net with kvno 3,
encryption type Triple DES cbc mode with HMAC/sha1 added to
keytab WRFILE:/etc/krb5.keytab.
kadmin: quit
#
# klist -k
Keytab name: /etc/krb5.keytab
KVNO Principal
---- --------------------------------------------
   3 host/kerberos.myware.net@MYWARE.NET
#
```

Figura 5.11: Criando e listando o arquivo de chaves

5.5 Operações administrativas sobre a base de dados do Kerberos

Além das atividades administrativas referentes aos principais de um reino Kerberos, o administrador também executa atividades sobre os arquivos que compõem a base de dados como um todo em um servidor KDC. Dentre essas atividades estão a criação dos arquivos do Kerberos, o *backup* e recuperação da base de dados.

O programa `kdb5_util` é a principal ferramenta para administrar a base de dados e sua sintaxe é

```
kdb5_util comando [opções_kdb5_util][opções_comando]
```

A Tabela 5.7 mostra as principais opções do programa. Se utilizadas, as opções sobrepõem qualquer padrão especificado nos arquivos de configuração.

Tabela 5.7: Opções do programa `kdb5_util`

Opção	Descrição
-r *reino*	Especifica o reino da base de dados.
-d *nome_arq*	Especifica o nome do arquivo da base de dados de principais.
-m	Especifica que a senha-mestra deve ser lida do terminal e não de um arquivo em disco.
-sf *arq*	Especifica o arquivo da senha-mestra (*stash file*)

5.5.1 Copiando a base de dados para um arquivo

Para copiar o conteúdo da base de dados para um arquivo, o administrador deve utilizar o comando `dump` do `kdb5_util`, executado no servidor KDC. A Figura 5.12 mostra um exemplo de execução.

```
# kdb5_util dump -r MYWARE.NET -verbose arqdump
K/M@MYWARE.NET
admin/admin@MYWARE.NET
ftp/servidor1.myware.net@MYWARE.NET
host/kerberos.myware.net@MYWARE.NET
host/cliente1.myware.net@MYWARE.NET
host/servidor1.myware.net@MYWARE.NET
kadmin/admin@MYWARE.NET
kadmin/changepw@MYWARE.NET
kadmin/history@MYWARE.NET
krbtgt/MYWARE.NET@MYWARE.NET
root/admin@MYWARE.NET
root@MYWARE.NET
telnet/servidor1.myware.net@MYWARE.NET
user1@MYWARE.NET
user2@MYWARE.NET
user3@MYWARE.NET
#
```

Figura 5.12: Copiando a base de dados

A opção `-verbose` mostra os principais na tela, à medida que são gravados no arquivo. O comando `dump` é utilizado para efetuar cópias de segurança da base dados e para gerar o arquivo de replicação da base de dados para servidores KDC escravos. Mais detalhes sobre replicação da base de dados do Kerberos podem ser vistos em [BRENNEN (2004)] e no "Guia de Instalação do Kerberos V5", do MIT [MIT (2002) (1)].

5.5.2 Restaurando a base de dados de um arquivo

Para recuperar o conteúdo da base de dados do Kerberos existente em um arquivo, o administrador deve executar o programa `kdb5_util` com o comando `load` no KDC que irá ser atualizado. A Figura 5.13 mostra um exemplo de execução.

```
# kdb5_util load -update -verbose arqdump
K/M@MYWARE.NET
admin/admin@MYWARE.NET
ftp/servidor1.myware.net@MYWARE.NET
host/kerberos.myware.net@MYWARE.NET
host/cliente1.myware.net@MYWARE.NET
host/servidor1.myware.net@MYWARE.NET
kadmin/admin@MYWARE.NET
kadmin/changepw@MYWARE.NET
kadmin/history@MYWARE.NET
krbtgt/MYWARE.NET@MYWARE.NET
root/admin@MYWARE.NET
root@MYWARE.NET
telnet/servidor1.myware.net@MYWARE.NET
user1@MYWARE.NET
user2@MYWARE.NET
user3@MYWARE.NET
created policy adm-pol
created policy user-pol
#
```

Figura 5.13: Restaurando a base de dados

A opção `-update` faz com que se atualizem principais existentes na base de dados, além de adicionar novos principais contidos no arquivo. Se essa opção não for utilizada o comando irá sobrepor a base existente com os principais do arquivo lido.

5.5.3 Criação do arquivo para a senha-mestra

Conforme visto na seção 4.3.3, ao se criar a base de dados do Kerberos, pode-se, opcionalmente, criar um arquivo oculto (chamado *stash file*) que armazena, localmente e de forma criptografada, a senha-mestra do KDC de modo que este autentique a si mesmo perante os programas `kadmind`, `krb5kdc`, `kadmin` e `kdb5_util`. Sem esse arquivo presente para a leitura da chave, o servidor

Kerberos irá solicitar a senha-mestra sempre que for necessário acessar a base de dados.

Caso não seja criado junto com os arquivos da base dados, é possível criá-lo posteriormente com o comando `stash` do programa `kdb5_util`, conforme mostra a Figura 5.14.

É possível especificar o arquivo para gravação da senha-mestra, com a opção `-f arquivo`. Caso não seja especificado será usada a localização existente no arquivo `kdc.conf`.

```
# kdb5_util stash
Enter KDC database master key:
#
```

Figura 5.14: Criando o arquivo de senha-mestra

5.5.4 Criação e remoção da base de dados

A criação dos arquivos da base de dados do Kerberos foi apresentada na seção 4.3.3. Para remover a base dados utiliza-se o comando `destroy` do `kdb5_util`, conforme mostra a Figura 5.15. Antes de remover os arquivos, o comando sobrescreve os setores do disco onde estão os arquivos.

```
# kdb5_util destroy
Deleting KDC database stored in
'/var/kerberos/krb5kdc/principal', are you sure?
(type yes to confirm)? yes
OK, deleting database '/var/kerberos/krb5kdc/principal'...
#
```

Figura 5.15: Removendo a base de dados

5.6 Administração do lado cliente

Como mostrado na seção 4.4.1, a instalação dos programas clientes numa estação

inclui programas para troca de senha e administração de *tickets*. Uma vez instalados, os usuários devem se habituar a utilizá-los de modo a poder trocar a senha do principal do Kerberos, além de obter, remover e verificar *tickets*.

5.6.1 Alteração de senha

A troca de senha do principal do Kerberos é feita com o comando `kpasswd` e sua utilização é semelhante ao comando `passwd` do Linux. A Figura 5.16 mostra a troca de senha com o comando `kpasswd`.

```
user1@cliente1:~$ kpasswd
Password for user1@MYWARE.NET:
Enter new password:
Enter it again:
Password changed.
user1@cliente1:~$
```

Figura 5.16: Alterando a senha do principal em uso

5.6.2 Obtenção de *tickets*

Para obter *tickets* de acesso a máquinas e serviços é necessário que se tenha obtido um TGT. A obtenção de um TGT se dá automaticamente no momento do *logon*, caso o programa de *login* do Kerberos (`/usr/sbin/login.krb5`) esteja sendo executado no lugar do programa de *login* padrão do Linux.

Caso contrário, o usuário necessita obter o TGT explicitamente através do programa `kinit`. O programa `kinit` também é necessário para obter novos *tickets* após sua expiração. Ao chamar o programa será solicitada a senha do Kerberos e, caso esteja correta, o *ticket* será armazenado no *cache* de credenciais.

Também é possível obter *tickets* para um principal que não é o mesmo que efetuou *logon*. Por exemplo, para um usuário comum obter um *ticket* para seu principal de administrador, basta executar o comando passando o principal administrativo como parâmetro.

A Figura 5.17 mostra a execução do `kinit` para obter um *ticket* para um principal diferente do atualmente logado.

```
user1@cliente1:~$ kinit admin/admin
Password for admin/admin@MYWARE.NET:
user1@cliente1:~$
```

Figura 5.17: Obtendo um *ticket*

Mais detalhes sobre a obtenção de *tickets*, bem como suas propriedades, podem ser vistos no "Guia do usuário do Kerberos V5", do MIT [MIT (2002) (3)].

5.6.3 Visualização dos *tickets*

Para visualizar os *tickets* que estão em *cache* utiliza-se o comando `klist`. Logo após o *logon* o único *ticket* existente é o TGT. Após acessar outras máquinas e serviços, outros *tickets* estarão em *cache* e podem ser visualizados.

A Figura 5.18 mostra um exemplo de listagem de credenciais.

```
# klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: admin/admin@MYWARE.NET

Valid starting   Expires           Service principal
02/07/05 14:42:55  02/08/05 00:42:55    krbtgt/MYWARE.NET@MYWARE.NET
#
```

Figura 5.18: Listando o arquivo de credenciais

5.6.4 Destruição dos *tickets*

A destruição dos *tickets* é importante e deve ser executada sempre que o usuário se ausentar da máquina ou efetuar *logoff*. Isso porque um invasor pode capturar *tickets* e se passar pelo usuário até que o *ticket* expire. Para destruir todos os *tickets* que estão em *cache* utiliza-se o comando `kdestroy`.

É recomendado que o comando seja colocado no arquivo `.logout` dos usuários para que seja executado sempre que um usuário efetuar *logoff*.

Capítulo 6

Considerações Finais

É importante frisar que implementar o Kerberos na rede não é garantia de total segurança. Teoricamente, o Kerberos é extremamente seguro. Porém, pelo lado prático, há diversos aspectos de segurança que precisam ser considerados. Alguns cuidados de segurança devem ser tomados de modo a não comprometer a infraestrutura do Kerberos, como dedicar servidores exclusivos para o KDC e desativar qualquer serviço desnecessário nesses servidores.

O Kerberos provê apenas o serviço de autenticação e criptografia; ele não pode prevenir falhas causadas por problemas em *software* de servidores e serviços, administradores relapsos que passam sua senha a outros, uma política de senhas mal elaborada, etc. A seguir são apresentados e discutidos diversos aspectos relativos a possíveis falhas de segurança e formas de ataque contra um sistema de autenticação Kerberos, seguidos de possíveis soluções para prevenção desses ataques.

6.1 Ataques à infraestrutura Kerberos

A primeira forma pela qual um invasor pode comprometer um ambiente Kerberos é diretamente sobre os servidores Kerberos. Porém, mesmo sem o acesso físico ao KDC, onde seria possível, por exemplo, desligá-lo, há uma série de ataques eletrônicos que um invasor pode utilizar na tentativa de comprometer o sistema Kerberos:

Acesso *root* ao servidor KDC

Conseguindo o acesso de *root* ao servidor KDC (principal ou escravos) um *cracker* obteria controle total sobre os serviços e arquivos instalados no servidor. Apesar do Kerberos, por padrão, não dar acesso aos bancos de dados para o usuário *root*, toda a base do Kerberos pode ser considerada comprometida caso um invasor consiga ter o acesso de *root* a qualquer um dos servidores KDC da rede pois ele terá acesso ao *software* e arquivos de configuração do Kerberos podendo modificá-los e danificar todo o sistema.

Uso de senha de um principal administrador

Outra forma de ataque é com a obtenção da senha de um principal que seja administrador do Kerberos. Isso daria ao invasor acesso completo à base de dados Kerberos permitindo que, mesmo remotamente, aquele pudesse criar, remover ou alterar qualquer principal. Um número reduzido de administradores, aliada a um conjunto de políticas rígidas para esses usuários, é a melhor forma de minimizar essa possibilidade de ataque.

Acesso *root* a um servidor

Para o funcionamento da autenticação mútua no Kerberos, os serviços devem ter um principal cadastrado. As chaves dos principais de serviços residem no arquivo *keytab* no servidor. Se um invasor consegue a senha do *root* de um servidor, todos os serviços que rodam nesse servidor ficam comprometidos. Além de comprometer os serviços, o invasor pode tentar descriptografar o tráfego entre os clientes e esse servidor. A segurança de serviços "kerberizados" que rodam em um servidor depende da segurança individual desse servidor; portanto, todos os servidores devem ser seguros em proporção ao valor dos recursos nele armazenados.

Acesso *root* a uma máquina cliente

Nesse caso, um invasor teria acesso a todos os *tickets* gravados em *cache* naquela estação. Como os *tickets* possuem tempo de expiração, o invasor teria pouco tempo para tentar reutilizar algum deles. Porém, o invasor pode implantar um *sniffer* de teclado e obter as senhas de usuários para uso

malicioso posterior. Assim, uma estação comprometida compromete as senhas de qualquer usuário que a tenha utilizado, devendo todos trocar a senha imediatamente.

Comprometimento das credenciais de usuário

Como mostrado acima há duas possibilidades nessa situação: o *ticket* que está em *cache* é exposto ou a senha do usuário é obtida. Se um invasor consegue acesso ao *ticket* este é válido apenas por um período de tempo. Por outro lado, conseguindo a senha de um usuário, o invasor pode se fazer passar por esse usuário até que este efetue uma troca de senha.

A partir da lista acima percebe-se a importância de manter todas as máquinas da rede seguras. Mesmo a instalação do Kerberos na rede não elimina a necessidade de tomar os devidos cuidados para que as máquinas, tanto servidores como estações de usuários, estejam seguras contra invasões e ataques. O comprometimento de qualquer máquina da rede terá, de alguma forma, efeitos sobre a segurança do sistema de autenticação Kerberos.

6.2 Segurança do protocolo

O Kerberos foi projetado para evitar que informações de autenticação passem abertas pela rede e, dessa forma, não possam ser violadas mesmo que capturadas. Assim, o Kerberos criptografa todos os dados relativos a uma autenticação quando esta ocorre através da rede. Na seção anterior foram apresentadas formas de ataques a uma rede com Kerberos. Aqui são apresentadas algumas formas de ataques sobre o protocolo Kerberos em si.

Ataques de dicionário e força-bruta

O ataque de força-bruta num ambiente Kerberos consiste em capturar um *ticket* e tentar descriptografá-lo. Se o invasor obtém a chave usada para criptografar a mensagem ele terá a senha do usuário e pode se passar por este no futuro. Como não há como descriptografar o *ticket* diretamente, o

invasor pode usar força-bruta sobre o TGT utilizando um ataque de dicionário em *off-line*.

Durante o ataque de dicionário é utilizado um programa, com uma lista contendo senhas comuns e possivelmente utilizadas por usuários (dicionário), que tenta descriptografar a mensagem utilizando cada uma das senhas. Se uma coincidência é encontrada o programa avisa o invasor sobre a senha do usuário.

É muito provável que, depois da descoberta da senha, o *ticket* tenha expirado seu tempo de validade. Porém, o *cracker* tem agora uma combinação de usuário e senha válidos, podendo obter novos *tickets* válidos.

Ataques de reenvio

Um ataque de reenvio envolve interceptar um *ticket* e então reapresentá-lo na tentativa de obter uma autenticação. Como há a troca de mensagens eletrônicas entre os computadores no momento da autenticação, um invasor pode capturar mensagens e reapresentá-las momentos depois. Ele não necessita descobrir senhas ou descriptografar mensagens. É um tipo de ataque ativo, pois o invasor tem que estar *on-line*, ouvindo a rede todo o tempo e com a capacidade de reenviar as falsas mensagens na rede.

Ataques *Man-in-the-middle*

Também é um tipo de ataque ativo, o que significa que o invasor precisa ser capaz de ler todas a mensagens que trafegam na rede, assim como enviar falsas mensagens. É um ataque que afeta protocolos que tentam verificar a identidade dos pontos finais de uma conexão, neste caso, o usuário e o servidor que esse usuário tenta acessar.

O objetivo do invasor é fazer se passar pelo servidor, de modo que o usuário pense que está em conexão com o servidor real quando, na verdade, está trocando mensagens com o invasor. Quando intercepta e passa a controlar a sessão, o invasor passa a ser parte da comunicação, atuando como um repassador das mensagens entre o usuário e o servidor real,

podendo adicionar novos pacotes, alterar ou remover mensagens, daí o nome *Man-in-the-middle.*

O protocolo Kerberos possui algumas proteções contra esse tipo de ataque. Como é feita autenticação mútua, pela confirmação não só do cliente mas também a identificação do servidor, um invasor teria que capturar e descriptografar as mensagens de ambos os lados da comunicação.

Um programa cliente pode detectar se o servidor é confiável solicitando a autenticação do servidor, que deve demonstrar sua identidade solicitando um *ticket* ao KDC (autenticando-se no Kerberos) e apresentando essas credenciais ao programa cliente. Caso essas credencias não sejam apresentadas, o programa cliente percebe que não é um servidor real e desconecta a sessão.

Um invasor também pode explorar produtos desatualizados. O Kerberos V5, apesar de ser um avanço e corrigir diversos problemas de segurança da versão 4, também já apresentou alguns pontos de falha para os quais são lançadas correções sempre que descobertas. Algumas da vulnerabilidades já conhecidas podem ser encontradas no *site* do MIT[10].

Além disso, os programas dos diversos serviços também devem estar preparados para usar Kerberos. O uso de algum serviço "não-kerberizado" pode levar ao trânsito de senhas abertas na rede podendo comprometer o sistema Kerberos. Portanto, o uso de serviços que não façam uso do Kerberos não é aconselhável. Para tais casos, é possível o uso de outros protocolos com criptografia, tais como SSH ou SSL/TLS.

6.3 Ataques genéricos

Os seguintes ataques não são ataques diretos contra um sistema Kerberos, mas são problemas relacionados a manter seguro e disponível um sistema de autenticação. Essas técnicas podem ser utilizadas contra qualquer sistema de autenticação.

10 http://krbdev.mit.edu/rt/NoAuth/krb5-1.4/bugs-1.4.html

Negação de serviço (*Denial of Service* – DoS)

Esse tipo de ataque consiste em inundar os servidores do KDC com solicitações de autenticação a ponto de tornar as respostas lentas e impedir as autenticações por *timeout* ou mesmo derrubar o serviço de autenticação. O uso de *firewall* e servidores KDC adicionais podem minimizar esse tipo de ataque.

Malefícios internos

Não há como se proteger de um administrador que resolva fazer mal uso dos seus privilégios. Um usuário interno mal intencionado, que possua autorização administrativa, poderia alterar danosamente as informações na base de dados Kerberos.

Engenharia social e exposição de senhas

De modo igual, não há proteção contra usuários que resolvam divulgar sua senha a outros, seja intencionalmente ou não, como num ataque de engenharia social. Daí a importância de treinamento em segurança da informação para o usuário final, incluindo-se como manter a senha segura e em sigilo, não a revelando mesmo para quem se identifique como sendo da área de tecnologia da empresa. Outra forma onde a senha pode ser capturada é o uso da mesma senha do Kerberos em outros sistemas inseguros.

6.4 Soluções de segurança

Tendo o conhecimento dos aspectos relacionados às possíveis vulnerabilidades e limitações do Kerberos, é possível relacionar soluções para diversos desses problemas e, assim, executar uma implementação de um sistema de autenticação o mais seguro possível.

Por exemplo, não se deve instalar servidores e ambientes gráficos nem permitir o *logon* para usuários comuns nos servidores que executarão o KDC. Outro fator importante na segurança do servidor KDC é a sua visibilidade na rede. É

fundamental que serviços de rede que não sejam do próprio KDC estejam desabilitados. Um servidor `ssh` pode estar habilitado para administração remota do servidor.

É de suma importância estar informado sobre correções de segurança do sistema operacional, do Kerberos e das versões de aplicações que dele fazem uso e aplicá-las tão logo seja possível, mantendo-se o ambiente sempre atualizado.

Em suma, a proteção dos servidores Kerberos, a manutenção do produto atualizado com relação às correções de segurança, aliados a uma política de segurança e senhas são grandes aliados para uma infraestrutura Kerberos segura, bem implementada e funcional.

6.5 Desvantagens do Kerberos

Apesar do Kerberos eliminar diversas brechas de segurança, sua implementação pode ser um pouco complexa. Uma dessas razões é a migração da base de dados de usuários existente. Não há ferramentas para migrar as senhas dos arquivos `/etc/passwd` ou `/etc/shadow` para a base de dados do Kerberos, tornando essa tarefa bastante tediosa.

Outra desvantagem é a necessidade (ou recomendação, visto que não é obrigatório) de servidores dedicados ao serviço do Kerberos.

O tempo de vida do *ticket* é um outro ponto que pode ser considerado como problemático. Um tempo de vida muito longo pode dar mais tempo a um invasor caso um *ticket* seja capturado ou uma estação comprometida. Um tempo muito pequeno impõe ao usuário a necessidade de redigitar a senha após a expiração. Também é necessário um suporte especial para processos que executam por longos períodos.

Finalmente, o Kerberos pode ser considerado como uma solução do tipo "tudo-ou-nada". Para ter segurança com o Kerberos é necessário utilizar versões "kerberizadas" de todas as aplicações cliente/servidor que utilizam autenticação ou não se terá benefícios com o Kerberos. As aplicações precisam estar adequadas para utilizar o Kerberos, ou seja, pode ser necessário modificar seu código para utilizar as

chamadas às bibliotecas do Kerberos. Dependendo da aplicação, tais alterações podem ser bastante complexas.

6.6 Vantagens do Kerberos

Para aqueles que não o conhecem totalmente, os benefícios de implantação do Kerberos em uma rede podem não ser muito claros. Porém, o Kerberos foi projetado para solucionar diversos problemas, como captura de informações através de escuta da rede, roubo de senhas e arquivos de autenticação, trânsito de senhas abertas na rede e a dificultosa tarefa administrativa de manutenção de múltiplas bases de dados de autenticação.

Uma infraestrutura Kerberos bem implementada irá, certamente, auxiliar na resolução desses problemas, tornando a rede da empresa mais segura. O uso do Kerberos previne que senhas abertas sejam transmitidas pela rede e permite centralizar as informações de usuários e suas senhas simplificando a administração. O Kerberos também previne o armazenamento local de senhas, reduzindo as possibilidades de danos à rede caso uma máquina em particular seja comprometida de alguma forma.

Além dos benefícios de um sistema de autenticação mais seguro, o Kerberos oferece outras características vantajosas, como a possibilidade de criptografar todo o tráfego que vai pela rede, reduzindo as chances de captura de informações por invasores.

O Kerberos também permite o uso de uma única senha para acesso aos serviços "kerberizados", permitindo que, uma vez autenticado, o usuário possa fazer diversas conexões a diversos servidores sem ter que redigitar sua senha.

Em suma, para grandes empresas, os benefícios do Kerberos se traduzem em redução dos custos administrativos, simplificação dos processos de gerenciamento de usuários e senhas, aliados ao aumento da segurança da rede. Em ambientes menores, a implantação de uma infraestrutura de autenticação escalável e uma rede com maior nível de segurança são vantagens claras.

6.7 Propostas para futuros trabalhos

Um implantação completa do protocolo Kerberos numa rede Linux envolve outras ações e vão além do propósito deste documento. Seguem algumas sugestões que podem ser utilizadas para desenvolvimento de outros trabalhos:

- configuração de servidores KDC escravos e replicação da base de dados;
- autenticação entre reinos;
- serviços de rede com Kerberos (WEB, SMTP, POP etc.);
- Uso de PAM, NSS e LDAP com Kerberos;
- Integração com outros sistemas operacionais (Windows, Mac, Unix).

Referências Bibliográficas

[BRENNEN (2004)] BRENNEN, V. A. **Kerberos Infrastructure HOWTO.** 2004. Disponível em: <http://www.linux.com/howtos/Kerberos-Infrastructure-HOWTO/index.shtml>. Acesso em: 28 fev. 2005.

[FERGUSON; SCHNEIER (2003)] FERGUSON, N.; SCHNEIER, B. **Practical Cryptography.** EUA: Wiley, 2003. 432 p.

[FERREIRA (2003)] FERREIRA, R. E. **LINUX – Guia do Administrador do Sistema.** São Paulo: Novatec, 2003. 512 p.

[GULBRANDSEN; VIXIE; ESIBOV (2000)] GULBRANDSEN, A.; VIXIE, P.; ESIBOV, L. **A DNS RR for specifying the location of services (DNS SRV)**. Internet Engineering Task Force (IETF), Fevereiro 2000. (Request for Comments: 2782). Disponível em: <http://www.ietf.org/rfc/rfc2782.txt>. Acesso em: 03 mar. 2005.

[HORNSTEIN (2000)] HORNSTEIN, K. **Kerberos FAQ, v2.0.** 2000. Disponível em: <http://www.faqs.org/faqs/kerberos-faq/general>. Acesso em: 02 mar. 2005.

[KOHL; NEUMAN (1993)] KOHL, J.; NEUMAN, C. **The Kerberos Network Authentication Service (V5).** Internet Engineering Task Force (IETF), Setembro 1993. (Request for Comments: 1510). Disponível em: <http://www.ietf.org/rfc/rfc1510.txt>. Acesso em: 03 mar. 2005.

[LEIPOLD (1999)] LEIPOLD, Cosimo. **Kerberos. Linux Journal.** Jan-1999. Disponível em: <http://www.linuxjournal.com/article/3329>. Acesso em: 01 mar. 2005.

[MILLS (1992)] MILLS, D. L. **Network Time Protocol (Version 3) – Specification, Implementation and Analysis.** Internet Engineering Task Force (IETF), Março 1992. (Request for Comments: 1305). Disponível em: <http://www.ietf.org/rfc/rfc1305.txt>. Acesso em: 03 mar. 2005.

[MIT (2002) (1)] Massachusetts Institute of Technology – MIT. **Kerberos V5 Installation Guide - Kerberos 5 Release 1.4.** 1985-2002. Disponível em: <http://web.mit.edu/kerberos/www/krb5-1.4/krb5-1.4/doc/krb5-install.html>. Acesso em: 28 fev. 2005.

[MIT (2002) (2)] Massachusetts Institute of Technology – MIT. **Kerberos V5 System Administrator's Guide - Kerberos 5 Release 1.4.** 1985-2002. Disponível em: <http://web.mit.edu/kerberos/www/krb5-1.4/krb5-1.4/doc/krb5-admin.html>. Acesso em: 28 fev. 2005.

[MIT (2002) (3)] Massachusetts Institute of Technology – MIT. **Kerberos V5 User's Guide - Kerberos 5 Release 1.4.** 1985-2002 Disponível em: <http://web.mit.edu/kerberos/www/krb5-1.4/krb5-1.4/doc/krb5-user.html>. Acesso em: 28 fev. 2005.

[MOCKAPETRIS (1987) (1)] MOCKAPETRIS, P. **Domain Names – Concepts and Facilities.** Internet Engineering Task Force (IETF), Novembro 1987. (Request for Comments: 1034). Disponível em: <http://www.ietf.org/rfc/rfc1034.txt>. Acesso em: 03 mar. 2005.

[MOCKAPETRIS (1987) (2)] MOCKAPETRIS, P. **Domain Names – Implementation and Specification.** Internet Engineering Task Force (IETF), Novembro 1987. (Request for Comments: 1035). Disponível em: <http://www.ietf.org/rfc/rfc1035.txt>. Acesso em: 03 mar. 2005.

[NCSA (2002)] University of Illinois - National Center for Supercomputing Applications (NCSA). **NCSA Kerberos Information.** 2004. Disponível em: <http://www.ncsa.uiuc.edu/UserInfo/Resources/Software/kerberos/index.html >. Acesso em: 02 mar. 2005.

[Red Hat (2003)] Red Hat, Inc. **Red Hat Linux 9 - Red Hat Linux Reference Guide - Chapter 17: Kerberos.** 2003. Disponível em: <http://www.redhat.com/docs/manuals/linux/RHL-9-Manual/ref-guide/ch-kerberos.html>. Acesso em: 01 mar. 2005.

[STANGER; LANE (2002)] STANGER, J.; LANE, P. T. **Rede Segura Linux.** Rio de Janeiro: Alta Books, 2002. 607 p.

[TUNG (1996)] TUNG, B. **The Moron's Guide to Kerberos.** 1996. Disponível em: <http://www.isi.edu/gost/brian/security/kerberos.html>.Acesso em: 01 mar. 2005.