William da Silva Vianna

PROPOSTA DE IMPLEMENTAÇÃO DE SEGURANÇA PARA REDES LOCAIS COM ACESSO A INTERNET.

Monografia apresentada ao Departamento de Computação da Universidade Federal de Lavras, como parte das exigências do curso de Pós-graduação Latu Sensu em Administração de Redes Linux, para obtenção do título de especialista em Administração em Redes Linux.

Orientador:
Prof. Fernando Cortez Sica
Co-orientador:
Joaquim Quinteiro Uchôa

LAVRAS
MINAS GERAIS – BRASIL
2004

William da Silva Vianna

PROPOSTA DE IMPLEMENTAÇÃO DE SEGURANÇA PARA REDES LOCAIS COM ACESSO A INTERNET.

Monografia apresentada ao Departamento de Computação da Universidade Federal de Lavras, como parte das exigências do curso de Pós-graduação Latu Sensu em Administração de Redes Linux, para obtenção do título de especialista em Administração em Redes Linux

APROVADA em _____ de _____ de _____

_____

Prof. Luiz Henrique Andrade Correia (Msc – CC/UFMG)

_____

Prof. Joaquim Quinteiro Uchoa (Msc – CC/UFLA)

(Co-orientador)

_____

Prof. Fernando Cortez Sica (Msc - CC/UFOP)

(Orientador)

LAVRAS

MINAS GERAIS – BRASIL

RESUMO

Desde o surgimento da Internet, a busca por melhores estratégias de segurança tem aumentado consideravelmente, tendo em vista milhares de ataques à segurança da informação, causando perdas e pânico. Esses ataques têm causado prejuízos financeiros e de imagem para empresas, instituições e pessoas físicas. Políticas de acesso e uso, *firewalls*, sistemas de detecção de intrusos, projetos de rede, *backups*, entre outros; tem sido as "armas" defensivas nesta guerra da informação. Nesta guerra existem jogadores ofensivos e defensivos, os jogadores defensivos devem estar preparados para tentar vencer as batalhas, logo o conhecimento e uso adequado das "armas" são as medidas de importância considerável para o bom administrador.

Nesse trabalho, foi feito um estudo de algumas técnicas e sistemas para implementação da segurança em redes e, a partir de um estudo de caso, foi apresentada uma proposta para implementação de segurança em redes locais com servidores e clientes com acesso a Internet.

Palavras chaves: segurança redes, Linux, Internet, Firewall, segurança informação, política de segurança, IDS.

SUMÁRIO

Lista de Figuras

LISTA DE TABELAS

1 INTRODUÇÃO

Com o crescimento da Internet, surgiu um ambiente onde todo tipo de informação ficasse acessível. Empresas, instituições e pessoas físicas criam seus microambientes, nos quais podem se apresentar com objetivos puramente culturais, comerciais, etc.

Desde o surgimento da Internet, a busca por melhores estratégias de segurança tem aumentado consideravelmente, tendo em vista milhares de ataques à segurança da informação, causando perdas e pânico. Esses ataques têm aumentado à medida que novas falhas de segurança são encontradas. Estar preparado para assegurar medidas de segurança adicionais é fundamental, logo a escolha das ferramentas, políticas e estratégias, têm sido uma medida de importância considerável para o bom administrador.

Diante do exposto, esse trabalho tem como objetivo:

- Apresentar um estudo de algumas técnicas e estratégias utilizadas na segurança em redes;
- Desenvolver, baseado no estudo de caso, uma proposta para implementação da segurança em redes.

Para melhor compreensão de como e por quem é feito o ataque, são apresentados conceitos e técnicas relativas a: perfil do inimigo, *footprinting*, enumeração, tipos de ataques, entre outros.

No desenvolvimento do estudo de caso, são apresentados conceitos e técnicas de defesa, políticas de segurança, sistema de detecção de intrusos (IDS), entre outros. A partir desse documento não se pretende exaurir os assuntos, pretende-se apresentar uma boa noção do caminho para se implementar segurança em redes.

Na elaboração do estudo de caso, utilizou-se a rede do Centro Federal de Educação Tecnológica de Campos (autarquia pública federal), portanto, dados importantes acerca da rede interna e sistema de proteção foram expostos nesse documento. Esse é um dos elementos que devem ser evitados, pois quanto menos o atacante souber melhor. Porém, acredita-se que as informações apresentadas serão utilizadas para fins lícitos e que em hipótese alguma os dados serão utilizados para fins contrários contra esta ou qualquer outra instituição, empresa ou pessoa física.

2 CONCEITOS BÁSICOS

A tecnologia não é tangível, não tem limites, e mesmo após décadas de desenvolvimento e revoluções no mercado de informática, as necessidades no mundo dos negócios estão muito longe de serem solucionadas. Com o crescimento exponencial do desenvolvimento tecnológico versus a competitividade que leva as empresas a uma redução de custos, as áreas de Tecnologia da Informação (TI) são forçadas a possuírem cada vez mais agilidade na administração dos recursos e aumento de produtividade.

Diante desse ciclo vicioso, o desenvolvimento de ferramentas que otimizem os recursos existentes e implementem técnicas de auditoria e controle na infra-estrutura de TI é cada vez maior. Estas necessidades levam as empresas de tecnologia a quebrarem paradigmas e derrubarem fronteiras, desenvolvendo mecanismos cada vez mais próximos da realidade virtual, isto é, mecanismos que controlem e monitorem em tempo real as atividades humanas impactantes no negócio através das redes.

As redes de dados convencionais atuais; incluindo LAN, WAN e MAN; são freqüentemente alvos de ataques de pessoas mal intencionadas, já que a Internet tornou-se um dos principais meios de comunicação do planeta.

Em alguns casos não é considerado que os invasores nem sempre utilizam a Internet como meio de comunicação, e muitas vezes eles utilizam nossa própria estrutura tecnológica. Isto acontece porque a maioria das redes LAN não possui mecanismo de controle de acesso, o que facilita o trabalho de pessoas não autorizadas, mas que possuem acesso físico a sua rede de dados. Esses invasores na maioria das vezes não possuem *login* e *password* para acessarem os servidores da rede, e conseqüentemente não possuem acesso ao

sistema de diretórios contidos no mesmo, mas quando um agente não autorizado possui acesso físico à infra-estrutura de rede, ele pode executar uma série de ataques, como por exemplo: tentar inicializar uma sessão telnet no servidor, *switch*, roteador, *firewall*, etc…; executar *scans* ou monitorar através de *sniffers* o tráfego da rede, acessar diretórios compartilhados em outras máquinas; atacar estações de trabalho ou servidores, explorar vulnerabilidades dos serviços; descobrir senhas de compartilhamento utilizando ataques de força bruta; entre outros (Francisco, 2003).

Em outras palavras, a rede está à mercê desses intrusos, tornando sua organização, sua produção e suas informações confidenciais, disponíveis, podendo causar impactos irreversíveis aos negócios. Esses intrusos são pessoas que possuem perfis distintos para conseguir o acesso.

2.1 PERFIL DO INIMIGO

Para melhor entendimento do perfil do inimigo, pode-se classificá-los em: (Marcelo, 2000, Rufuno, 2002, Hacker, 2003).

Script Kid – confundido como *hacker*. Utiliza alguns programas prontos para descobrir senhas ou invadir sistemas (receitas de bolo). Também conhecido como *Wannabe*;

Lamer **-** geralmente tem um conhecimento pouco maior que o *Script Kid*. Normalmente é um adolescente que aspira ser um *hacker*. Também confundido como *hacker*;

Larva – geralmente tem conhecimentos próximos de um *cracker*. Já consegue desenvolver suas próprias técnicas de como invadir sistemas;

Hacker **-** tem conhecimentos reais de programação (geralmente C, C++, *Assembler*, entre outras) e de sistemas operacionais, principalmente o *Unix*, o mais usado nos servidores da Internet. Conhece falhas de segurança dos sistemas e procura achar novas. Desenvolve suas próprias técnicas e desprezas as

"receitas de bolo". Normalmente trabalha para resolver problemas e não criá-los;

Cracker – geralmente invade sistemas a fim de dar problemas a algo ou a alguém. Também desenvolve rotinas que quebram as proteções dos *software* com licença comercial;

***Phreaker* -** tem bons conhecimentos de telefonia e consegue inclusive fazer chamadas internacionais sem pagar, o que lhe permite desenvolver seus ataques a partir de um servidor de outro país;

***Guru ou Coder* -** geralmente um hacker codificador com anos de experiência;

Virii – programadores e colecionadores de vírus;

Como pode ser observado, o termo *hacker* e *cracker* é confundido. Logo para fins de discussão e por ser mais conhecido, optou-se por utilizar o termo atacante ou invasor no desenvolvimento do texto.

Partindo do raciocínio de quem conhece as falhas de segurança é quem poderá resolvê-las, alguns administradores passaram a achar que somente um invasor pode ajudá-lo a resolver seus problemas. Mas infelizmente não é isso que se tem visto, pois o primeiro ponto que não é questionado na contratação de invasor, é a ética. Invadir computadores tornou-se um vício, e, como em qualquer outro vício, dizer que “parou” não costuma ser indicativo suficiente de que isso realmente tenha ocorrido; o que se tem observado de forma recorrente são funcionários recém-contratados invadindo redes alheias, às vezes de dentro das próprias empresas contratantes.

Por outro lado, existem as empresas especializadas em segurança. Algumas empresas de segurança tentam vender seus produtos apresentando em seu quadro funcional um ou mais “*hackers* éticos” como sendo um diferencial em relação a concorrência. Ainda em alguns casos, o “*hacker*” invade e em seguida se oferece para fazer proteção. Não somente “consultores independentes”, mas também empresas de segurança, vêm se utilizando,

incrivelmente com sucesso, desses artifícios. Pelo menos em um caso de incidente de segurança isso foi comprovado, quando em julho de 1999 os jornais noticiaram que a Polícia Federal apreendeu 14 computadores de uma empresa de segurança em Brasília envolvida, entre outras ilegalidades, com invasões seguidas de oferta de consultoria (Rufuno, 2002).

2.2 DEFINIÇÃO DE INCIDENTE DE SEGURANÇA

Tratando-se de segurança em redes, a questão é “quando e não se” o incidente vai ocorrer, ou seja, em algum momento ocorrerá um incidente de segurança na rede com um maior ou menor nível de gravidade, desta forma, faz-se necessário definir claramente o que é um incidente de segurança. Esta definição deve estar contida em nossa política de segurança, mas de forma genérica pode-se classificar como incidente de segurança em:

"Invasões de computador, ataques de negação de serviços, furto de informações por pessoal interno e/ou terceiros, atividades em rede não autorizadas ou ilegais" (Keating, 2001).

Desta forma, além do conhecimento do que é um incidente de segurança, deve-se estabelecer medidas de pré e pós-incidente, ou seja, deve-se estar preparados para a ocorrência de um incidente.

Entre as medidas pré-incidentes pode-se incluir (Francisco, 2003):

- Classificação dos recursos a serem protegidos;
- Implementação de mecanismos de segurança;
- Definição de equipe multidisciplinar para atuar em caso de incidentes;
- Classificação dos incidentes quanto ao nível de gravidade;
- Elaboração da estrutura administrativa de escalonamento do incidente (do operador, passando pelos gerentes até o presidente);
- Montagem de *kit* de ferramentas para atuar em incidentes em plataforma diversas;

- Definição de procedimentos a serem adotados;
- Entre as medidas de pós-incidentes pode-se incluir:
- Procedimentos de coleta e preservação de evidências;
- Procedimentos de recuperação dos sistemas afetados;
- Procedimentos de rastreamento da origem;
- Elaboração de processo legal contra o causador do incidente;

Estes são alguns dos procedimentos/ações a serem adotados. Observa-se que a resposta a um incidente inicia antes da ocorrência do mesmo com a adoção de certos procedimentos.

2.2.1 INCIDENTES DE SEGURANÇA REPORTADOS

A cada ano vem aumentando o número de incidentes de segurança que são reportados. A figura 1 apresenta gráfico do número de incidentes reportados no *Coordination Center Software Engineering Institute Carnegie Mellon University Pittsburgh* (CERT) de 1988 até agosto de 2003. No Brasil, os incidentes podem ser reportados ao Centro de Atendimento a Incidentes de Segurança (CAIS) ligado a Rede Nacional de Pesquisa (RNP). A figura 2 apresenta gráfico dos incidentes reportados no CAIS de 1997 até setembro de 2003.

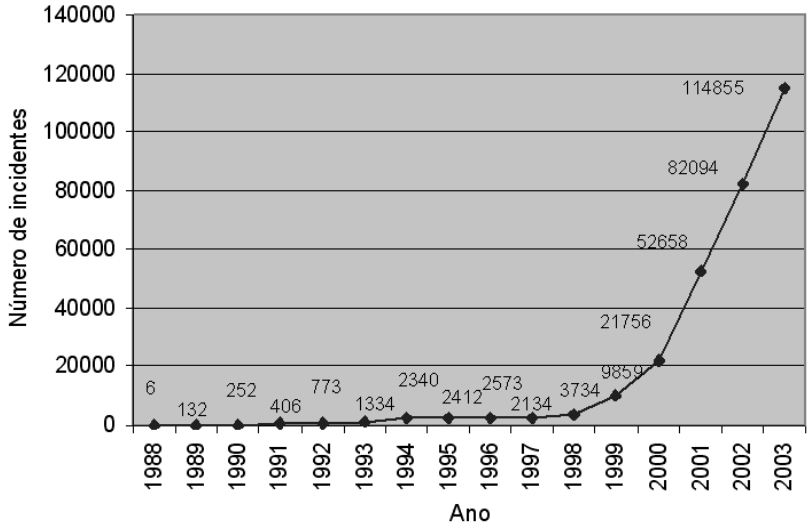


Figura 1 - Gráfico do número de incidentes reportados no CERT versus ano (CERT, 2003)

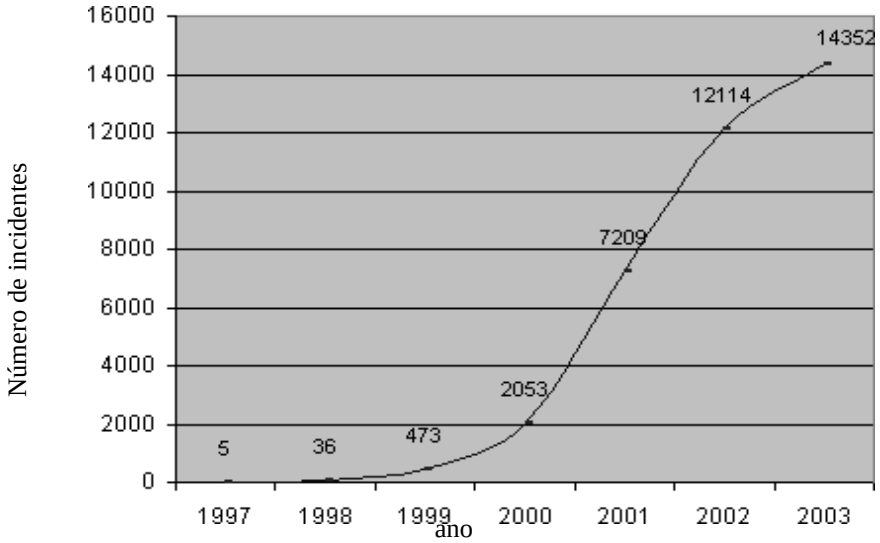


Figura 2 - Gráfico do número de incidentes reportados no CAIS versus ano (CAIS, 2003)

Analisando os gráficos das figuras 1 e 2, pode ser observado que, os números são relativamente altos e aumentam a cada ano. Nos gráficos da figura 3, pode ser observado o número de incidentes reportados por mês nos anos de 2000 a 2003. Entretanto muitos dos incidentes não são reportados, o que acaba gerando uma leitura falsa da realidade da segurança.

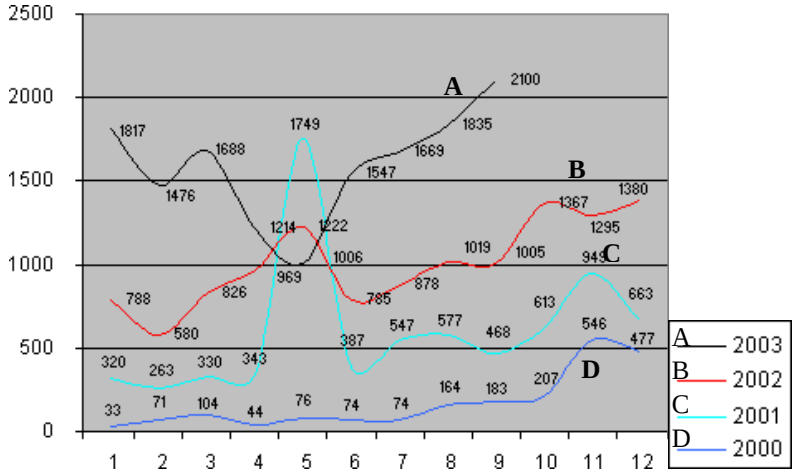


Figura 3 - Gráfico do número de incidentes reportados no CAIS versus meses para anos de 2000 a 2003

A partir dos gráficos da figura 3, pode-se concluir que não existe um período do ano em que a quantidade de incidentes é maior, entretanto conclui-se que no últimos anos a quantidade de incidentes vem aumentando. Isto pode ser gerado por uma série de fatores, tais como:

- Programação insegura;
- Falta de atualização dos *software* de rede;
- Disseminação de tecnologia de banda larga;
- Falta de conhecimento em segurança pelos administradores;
- Falta de políticas de segurança;
- Aumento da facilidade do uso das ferramentas modernas de ataque.

A facilidade de uso das ferramentas atuais vem contribuindo para o

aumento do número de ataques a segurança. A figura 4 apresenta esquema do nível de conhecimento em função do nível de sofisticação das ferramentas utilizadas nos ataques.

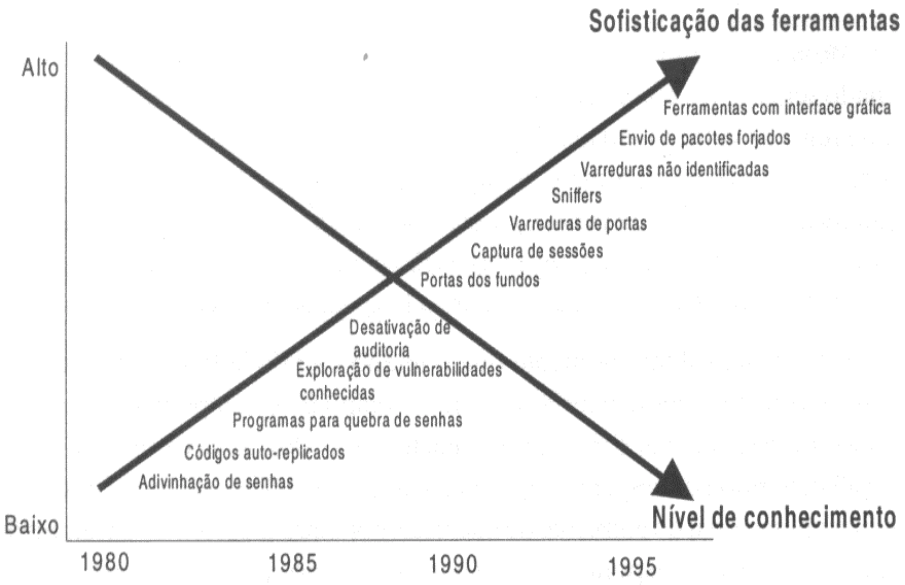


Figura 4 – Esquema do nível de sofisticação da ferramenta versus nível de conhecimento ao longo dos anos (Rufino, 2002).

Pode-se observar que em função dos anos as ferramentas tornaram-se mais simples de serem utilizadas. Desta forma qualquer um pode se tornar um potencial atacante mesmo sem conhecimento e/ou habilidades. Antigamente executava-se “receitas de bolo” (*cook book guy*), indicando que tinha um roteiro a seguir, normalmente com vários passos, de maneira, por vezes não sincronizada e não necessariamente seqüencial. Em seguida passaram a ser os que executavam *scripts* (*scripts kids*), ou seja, programa ou rotinas previamente ordenadas, em que só era preciso executar um único comando e o *script* fazia o resto. Hoje existe um novo termo: “apertadores de botões” (*push botton kids*), indicando a forma “complexa” de interação com ferramentas de ataques atuais.

Desta forma, a segurança da informação precisa ser tratada com estratégias militares, pois o inimigo pode vir de qualquer lugar.

2.3 SEGURANÇA EM REDES VERSUS ESTRATÉGIAS MILITARES

É possível fazer uma analogia entre a estrutura de segurança das redes com as estratégias militares. A figura 5 apresenta esta analogia de forma esquemática.

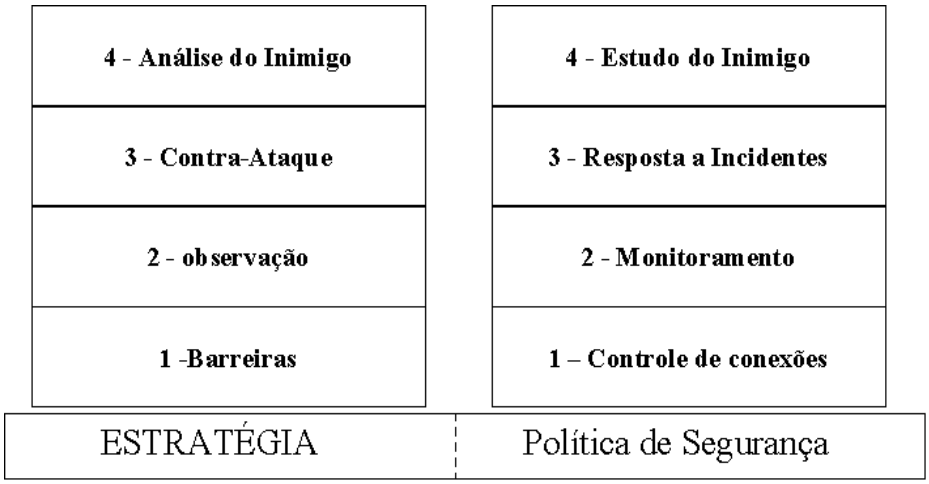


Figura 5 - Comparativo entre estratégia militar e segurança em redes (Francisco, 2003).

De forma genérica, a base do sistema militar em situações de conflito é a estratégia de combate no qual são definidas as ações, recursos a utilizar, entre outros. Já nos sistemas de segurança de redes, a base deve ser a definição de uma política de segurança, que irá reger as estratégias a serem adotadas, os procedimentos, os níveis hierárquicos, a classificação de ativos, a classificação da informação, etc.

Em seguida, no sistema militar, existem as barreiras que têm por objetivo "conter" o avanço das tropas inimigas. Paralelamente, nos sistemas de

informação, existem os elementos de controle de conexão, tais como: lista de controle de acesso em roteadores, programas para controle de conexão *stateful*, e remoção de serviços desnecessários dos servidores.

A próxima camada é a observação do inimigo. No sistema de informação tem-se os sistemas de detecção de intruso, os IDS - *Intrusion Detection System*, esses sistemas podem funcionar de diversas formas, entre elas: "monitoram" o tráfego da rede e *hosts* com o objetivo de identificar padrões de ataque e tomar algumas ações de contra-resposta, detectar tentativas de conexão não autorizadas, verificação de integridade de arquivos, entre outras.

Na quarta camada dos sistemas militares tem-se o contra-ataque que é executado em função das ações adotadas pelo inimigo. Nos sistemas de informação tem-se a resposta a incidente de segurança que envolve vários procedimentos, tais como: identificação do incidente, notificação das pessoas responsáveis, coleta e preservação de evidências, rastreamento da origem, ações de contra-resposta.

Finalmente tem-se o último estágio que nos sistemas militares corresponde à análise do inimigo, sendo representado pela espionagem, onde se busca descobrir as técnicas e táticas que serão adotadas. Nos sistemas de informação o correspondente é o estudo do inimigo, que significa a implementação de *honeypots* e *honeynets* para o estudo das ações e comportamento dos invasores com a finalidade de compreender sua mentalidade a fim de melhor proteger os sistemas críticos.

Como se pode observar na analogia, cada nível necessita dos serviços da camada anterior e provê serviços para a camada superior, como na pilha TCP/IP. Desta forma, esta pode ser uma das formas de mensurar a maturidade em segurança de uma instituição, empresa ou pessoa física. Enfim, a maturidade em segurança de sistemas de informação parte da análise das possibilidades e proteções que podem ou não acarretar em um incidente de segurança. Para

melhor entendimento das proteções serão apresentadas, nas próximas seções, técnicas e procedimentos utilizados pelos atacantes.

2.4 FOOTPRINTING

O *footprinting* de uma organização permite que atacantes criem um perfil completo da postura de segurança da mesma. Usando combinações de ferramentas e técnicas, atacantes podem empregar um fator desconhecido (a conexão a Internet da Empresa X) e convertê-lo em um conjunto específico de nomes de domínio, blocos de rede e endereços IP individuais de sistemas conectados diretamente à Internet. Embora existam técnicas diferentes de *footprinting*, o seu objetivo principal é descobrir informações relacionadas a tecnologia de Internet, Intranet, acesso remoto e Extranet. A tabela 1 apresenta as tecnologias e informações críticas que um atacante tentará identificar.

Tecnologia	Identifica
Internet	- Nomes de domínios. - Blocos de rede. - Endereços IP específicos de sistemas atingíveis via Internet. - Serviços TCP e UDP executados nos sistema identificado. - Arquitetura do sistema (por exemplo, SPARC versus x86). - Mecanismos de controle de acesso e listas de controle de acesso (ACL, access control list) relacionadas. - Sistemas de detecção de intrusos (IDS). - Enumeração de sistemas (nomes de usuários e de grupos, faixas de sistemas, tabelas de roteamento, informações SNMP).
Intranet	- Protocolos de rede em uso (por exemplo IP, DecNET, etc). - Nomes de domínios internos. - Blocos de rede. - Endereços IP específicos de sistemas atingíveis via Internet. - Serviços TCP e UDP executados nos sistema identificado. - Arquitetura do sistema (por exemplo, SPARC versus x86). - Mecanismos de controle de acesso e listas de controle de acesso relacionadas. - Sistemas de detecção de intrusos (IDS). - Enumeração de sistemas (nomes de usuários e de grupos, faixas de sistemas, tabelas de roteamento, informações SNMP).
Acesso remoto	- Números de telefones analógicos/digitais. - Tipo de sistema remoto. - Mecanismo de autenticação.
Extranet	- Origem e destino de conexões. - Tipo de conexões. - Mecanismos de controle de acesso.

Tabela 1 - Tecnologias e as informações críticas que os atacantes podem identificar (McClure, 2000).

Para o atacante obter informações de contas, setores, organização, entre

outras; pode-se fazer consultas simples ao site da organização. Além disto, consulta a sites de registro de domínios também é uma boa fonte de informações. No Brasil pode-se utilizar http://registro.br.

Algumas ferramentas como *nslookup*, *dig*, *host* e *whois* podem obter informações valiosas como: lista de IPs, estado do funcionamento do servidor de nomes, IP do servidor de e-mail, IP e versão do servidor de nomes, etc.

2.5 ENUMERAÇÃO

Máquinas ativas.

Após a identificação dos IPs utilizados pela organização, geralmente o atacante identifica quais máquinas estão ativas. Para isto, são utilizadas consulta ICMP tipo ECHO REQUEST ou conexão, quando negado, envia-se um ACK para uma determinada porta de um serviço, o *host* que responder será considerado como ativo. Existem muitas ferramentas que podem ser utilizadas para esse fim, entre elas: *ping*, *Cheops*, *Nmap* e *Nessus*.

Sistemas operacionais.

Para se descobrir qual o sistema operacional e outras características do *host* remoto, pode-se utilizar o método *Passive Fingerprinting*. Este método é utilizado para aprender mais sobre o inimigo, sem que ele fique sabendo, usando nada mais do que traços de *sniffer*.

Tradicionalmente o *fingerprinting* (impressão digital) tem sido usado ativamente em ferramentas tais como *Nmap* ou *Queso*. Elas operam no princípio de que a pilha IP (IP STACK) do sistema operacional tem sua própria idiossincrasia, eles respondem de maneira diferente a uma variedade de pacotes fragmentados, sendo assim, o que se faz é construir um banco de dados de como cada sistema responde a diferentes tipos de pacotes, o programa (*scanner*) envia pacotes mal formados e compara a resposta do *host* com o banco de dados para verificar qual tipo de sistema operacional o alvo possui (Sptizner, Fyodor, 2003).

A impressão digital passiva segue o mesmo conceito, mas é implementada de outra forma. Ela é baseada em traços de *sniffer* de um *host* remoto. Ao invés de fazer requisições para um *host*, captura-se pacotes enviados do *host* remoto, então, baseado nas características do pacote, pode-se determinar o sistema operacional do sistema remoto. Baseado no mesmo princípio de que o IP *stack* de um sistema tem sua própria idiossincrasia.

Para esse fim pode-se utilizar ferramentas como: *Nmap* e *Nessus*.

Contas.

Consiste em basicamente listar contas que são utilizadas no sistema. Para isto pode-se:

- Fazer uma simples consulta ao site da empresa/organização;
- Utilizar engenharia social para obter contas e senhas;
- Fazer ataque de força bruta com ferramentas que automatizam a tarefa como *Brutus*;
- Usar programas que capturam e registram teclas digitas, como chamados *keylogs*, ou ainda telas do console;
- Capturar dados da rede local utilizando *Sniffers* que possibilitam a interface de rede trabalhar no modo promíscuo. Entre as mais utilizadas tem-se: *Tcpdump*, *Ethereal*, *Ethercap* e *Spy*;
- Obter arquivos das contas e utilizar programas que "quebrem" as senhas como o *John The Ripper* e *Crack*.

Serviços.

A identificação dos serviços consiste em listar quais as portas do *host* remoto que estão em escuta. Para listar esses serviços, também pode-se utilizar passive *fingerprinting* possibilitando um *port scan* furtivo.

Pode-se utilizar ferramentas como *Nmap* e *Nessus*.

Programas e versões dos servidores.

De posse da listagem das portas que estão em escuta, pode-se utilizar o

método *banner* para identificar o nome do programa e versões executadas no servidor. Este método consiste em gerar uma conexão na porta que está em escuta com determinados dados e aguardar pela resposta do servidor. Como exemplo para identificação de um servidor WEB pode-se utilizar os comandos:

telnet vitima.coitado.com.br 80 <ENTER>

GET/HTTP/1.0 ou HEAD/HTTP/1.0 <ENTER>

Também pode-se utilizar ferramentas que automatizam estas tarefas, por exemplo *Nessus* que pode ser obtido gratuitamente em http://www.nessus.org.

Com estas informações, geralmente, o atacante busca ou desenvolve uma ferramenta para promover o ataque.

2.6 ATAQUES MAIS COMUNS

2.6.1 DENIAL OF SERVICE

Os ataques de negação de serviço iniciaram-se através de acesso local na máquina atacada com a execução de programas para acabar com o espaço livre em disco de sistema, criação de *loop* de *fork* ou criação recursiva de diretórios, entre outros. A seguir surgiram os ataques remotos, tais como: SYN *flooding*, *smurfing*, UDP *flooding*, *finger flooding*, entre outros. Daí para ataques coordenados com vários *hackers* atacando uma vítima simultaneamente foi um pulo. Em seguida, surgiram os ataques combinados, onde um único programa apresenta várias vertentes de ataques (*Targa* – inclui *Bonk, Jolt, Nestea, Newtear, Syndrop, Teardrop* e *Winnuke* em um programa). Todos esses ataques anteriores tinham a característica de ser ponto-a-ponto (com relação do atacante para o atacado).

2.6.2 DISTRIBUITED DENIAL OF SERVICE

Em fevereiro de 2000, o mundo inteiro noticiou o ataque sofrido por grandes sites da Internet , como *Yahoo*, *Amazon*, cnn.com, *Zdnet* e outros. Um tipo de ataque já conhecido no meio de pesquisa da Internet foi o causador deste alvoroço, seu nome é DDoS.

O DOS (*Denial of Service*) já era velho conhecido, porém o DDoS (*Distributed Denial of Service*) multiplicou em muito o seu poder de ataque causando grande temor e apreensão num mundo com negócios cada vez mais dependentes de segurança na Internet.

O DDoS é um aperfeiçoamento do DoS. Seu objetivo principal é impedir o acesso de usuários legítimos a um grande provedor de Internet, colocando em risco, inclusive, a sua reputação e credibilidade.

Os ataques distribuídos caracterizam-se pela rede que é formada para sua execução. Um número variável de nós, que podem chegar aos milhares, podem ser facilmente utilizados para gerar um ataque a uma rede ou servidor vítima.

As redes DDoS são hierárquicas , divididas em 4 níveis. Os seguintes tipos de nós podem ser encontrados (Mariano, 2000):

1- Cliente – É o equipamento de onde o atacante monta toda a rede de ataque.

2- *Handler* ou *Master* – Controlados pelos clientes, enviam ordem de ataque e definem também o método de ataque.

3- Agente ou zumbi – Geram o tráfego de ataque a partir da ordem recebida pelo *handler*

4 - Vítima – É um servidor ou uma rede alvo do ataque.

A figura 6 apresenta o aspecto da topologia de rede para o ataque DDoS.

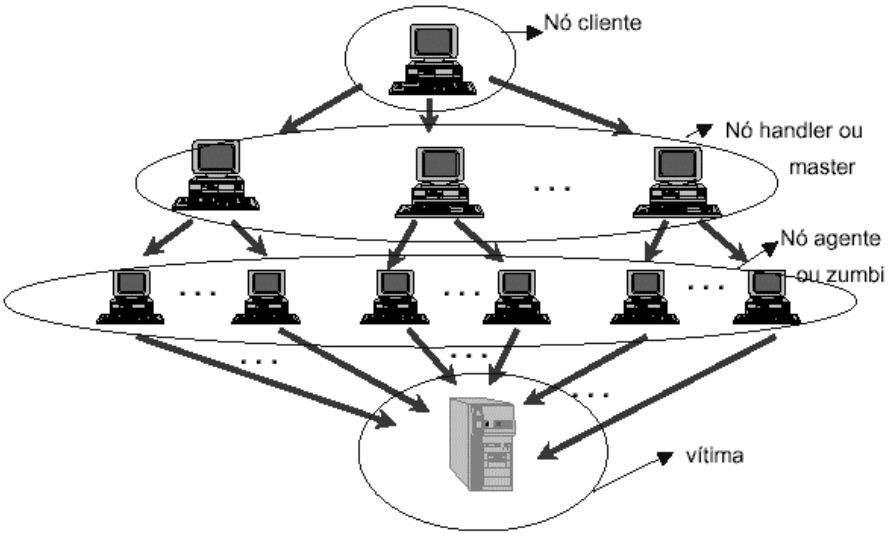


Figura 6 – Aspecto da topologia da rede DDoS (Mariano, 2000).

Nesse tipo de rede a comunicação se dá hierarquicamente na seguinte ordem:

CLIENTE ←→ HANDLER ←→ AGENTE ←→ VÍTIMA

Esta comunicação pode utilizar, inclusive, mensagens criptografadas.

A rede DDoS é utilizada pelo atacante para iniciar um ataque à vítima que normalmente caracteriza-se por um bombardeamento de mensagens que pode causar utilização de 100% da banda disponível ou também uma sobrecarga na UCP (unidade central de processamento) de um servidor.

As técnicas de ataque DDoS mais conhecidas são (Mariano, 2000):

- *Smurfing* – O atacante envia um pacote de ECHO REQUEST contendo endereço de destino um broadcast e endereço de origem a vítima, que receberá pacotes ECHO REPLY de todos os nós que compõem o endereço de broadcast.
- UDP *flooding* – Um grande número de nós agentes geram pacotes UDP ECHO REQUEST com endereço de destino da vítima e endereço fonte

aleatório. A vítima pode ter seus recursos consumidos rapidamente.

- TCP *flooding* – Aproveita-se do three-way-handshake. Vítima recebe diversos pedidos de conexão cujos endereços fonte são fictícios e aguarda resposta que nunca chegará.
- *Address spoofing* – O atacante altera o endereço fonte dos pacotes para poder penetrar numa rede.

Pode-se observar que um ataque DDoS pode apresentar-se de várias formas. Uma das formas de diminuir a ação do DDoS é tomar providências para dificultar a criação da rede DDoS. As vulnerabilidades conhecidas devem ser testadas, sistemas devem ser corrigidos, *patches* devem ser instalados e atualizados. O DDoS inicia-se a partir do momento que um atacante possa ter em mãos uma lista de nós vulneráveis a partir de onde ele possa iniciar os ataques (Mariano, 2000). Além disso, existem na Internet ferramentas de busca que podem detectar programas e aplicativos que compõem um tipo de ataque DDoS.

O FIND_DDOS é uma ferramenta de scan desenvolvida pelo *Network Infrastructure Protection Center* (NIPC) que detecta programas do TRINOO e TFN. Sua execução é local em cada máquina cliente, *handler* ou agente. Ele pode ser encontrado na página http://packetstorm.security.com/distributed.

O *Remote Intrusion Detector* (RID) é outra ferramenta de busca que também faz a detecção de programas do TRINOO e TFN. Sua execução, porém pode ser remota e a medida que novas assinaturas sejam descobertas (nova portas de comunicação, novas senhas, etc.) ele pode ser reconfigurado para sua detecção. Pode ser encontrado na página http://www.theorygroup.com/Software/RID.

Outra ferramenta útil para a detecção de DDoS são os *Network Intrusion Detection Systems* (NIDS). Eles possuem a capacidade de monitorar a rede e

instantaneamente detectar uma tentativa de comunicação entre nó TRINOO, por exemplo.

A maior dificuldade é que a medida que se descobrem os tipos de ataque, novas variações são criadas.(Mariano, 2000)

2.6.3 BUFFER OVERFLOW

O *buffer overflow* (Estouro de Buffer) tem sido a forma mais comum de exploração de vulnerabilidade de segurança nos últimos anos. Esta vulnerabilidade permite penetrações a partir de redes remotas, onde um usuário anônimo da Internet visa obter controle parcial ou total de um *host*. Devido ao fato desse tipo de ataque permitir que qualquer um obtenha controle total de um *host*, representa uma das ameaças mais sérias a segurança (Cync, 2003, DilDog, 1998, Crabb, 1997, Mudge, 1997, NathanP, 1997).

Um buffer é um simples bloco de memória que armazena múltiplas solicitações de um mesmo tipo de dado. O princípio do *buffer overflow* é bem simples, é o resultado de quando se põe mais dado num buffer do que ele pode suportar. Freqüentemente devido a erros na programação, pode-se tomar vantagem para execução de código arbitrário.

As principais funções da CPU são o processamento e movimentação de dados. Enquanto processa esses dados ela precisa de um lugar para rapidamente salvar informações importantes, devido ao espaço limitado dos registradores ("células de memória"). Essas informações são salvas no *stack*, uma parte muito especial da memória que pode ser acessada com algumas instruções em *Assembler*.

Para que um atacante tenha sucesso nesse tipo de ataque deve-se possuir habilidade para injetar e executar código malicioso. O código de ataque injetado é executado com os privilégios do programa vulnerável, e permite o atacante obter qualquer outra funcionalidade necessária para controlar o computador *host*.

Para que esse código seja injetado no *host* é necessário um *exploit*, programa especialmente desenvolvido para explorar vulnerabilidades de segurança. A maioria dos *exploits*, baseada em buffer *overflow*, consiste na sobrescrita do endereço de retorno na área de *stack*, apontando para execução de uma série de comandos geralmente em *Assembler* (*shellcode*). Se a área de *stack* é definida como não executável, vulnerabilidades do tipo *buffer overflow* tornam-se mais difíceis de serem exploradas. Outro modo de explorar um buffer *overflow* é apontar o endereço de retorno para uma função na *libc*, usualmente *system*().

Para que o administrador tenha conhecimento das vulnerabilidades do tipo buffer *overflow* é necessário que faça visitas periódicas a sites especializados ou esteja cadastrado em uma boa lista de discussão que trate desse assunto. Um dos sites mais completos que trata desse e de outros assuntos de segurança é http://www.securityfocus.com. Neste tipo de site pode-se encontrar informações sobre vulnerabilidades como: discussões, *exploits*, soluções, entre outras. Geralmente, para se evitar que a vulnerabilidade de buffer *overflow* seja explorada é necessário que os programas servidores, com esse tipo de vulnerabilidade, sejam atualizados para novas versões.

2.7 MEDIDAS PÓS-INCIDENTE

A detecção do incidente pode ser feita com os sistemas descritos nas camadas 1 e/ou 2 da figura 5, ou seja, através dos mecanismos de controle de conexão como, roteadores, sistemas de *firewalls* e/ou ferramentas de IDS. Além desses mecanismos o administrador pode ser alertado por usuários e/ou parceiros quanto ao mau funcionamento de determinadas aplicações e/ou suspeita de má utilização dos recursos.

Diversas são as formas de detecção de incidentes e o momento de ocorrência de um pode em alguns casos até ser prevista. Por exemplo, após a

notificação de uma vulnerabilidade em determinada aplicação anunciada pelo CERT® *Coordination Center Software Engineering Institute Carnegie Mellon University Pittsburgh* (CERT, 2003), o sistema pode ser alvo de tentativas de intrusão.

Após a identificação de um possível incidente deve-se:

1. Confirmar a ocorrência do mesmo, de forma a evitar esforço desnecessário, ou seja, distinguir entre falso-positivo (incidente falso reconhecido como real) e incidente real;
2. Registrar todas as ações tomadas;
3. Definir o nível de dano do incidente;
4. Identificar sistemas atingidos direta ou indiretamente;
5. Observar se o incidente continua em curso;
6. Acionar os especialistas necessários para a resposta ao incidente;
7. Notificar a gerência quanto ao estado do sistema, tempo estimado de recuperação e ações de contra-resposta;
8. Isolar os sistemas atingidos até a recuperação do mesmo e coleta das evidências.

As ações de contra-resposta não significam bombardear a origem do ataque, caso se tenha identificado a mesma, mas sim os procedimentos de preservação das evidências, recuperação do sistema. Esta questão é fundamental, pois realizar uma ação de "ataque" contra o agressor não é uma medida muito inteligente visto que:

1. O tempo será gasto com uma ação que poderá gerar efeitos inesperados;
2. O agressor ficará sabendo que foi descoberto;
3. Estará utilizando os recursos da organização de forma incorreta;
4. Estará contribuindo para a elevação do nível de lixo na Internet como um todo;

Os procedimentos mais recomendados são a discrição e ações para

identificação da técnica utilizada para comprometer o sistema. Somente as pessoas certas devem ser notificadas, com informações claras do que ocorreu. As ações que estão sendo tomadas e os tempos estimados para normalização do sistema. Tratando-se de uma organização com fins lucrativos, os gerentes estarão mais preocupados com a normalização dos negócios do que com a identificação das origens. Sabe-se que a normalização é apenas uma das fases do processo de resposta e, provavelmente, uma das mais simples e rápidas.

Todas as informações coletadas na fase inicial servirão de base para a elaboração da estratégia de acompanhamento e investigação do incidente. Esta estratégia deve contemplar aspectos técnicos e financeiros, a mesma deve ter a aprovação da direção da instituição, pois conforme o nível de gravidade do incidente não é justificável sua investigação. Algumas vezes, para o desespero dos especialistas em crime eletrônico, a organização decide simplesmente não continuar um processo de investigação com receio de manchar a imagem da mesma. Essa visão deve ser mudada, pois todas as redes estão sujeitas a intrusão, e esconder a ocorrência de uma não ajuda a comunidade em nada. Lógico que não significa que a mesma deva ser tornar pública, mas sim que deve ser analisada com seriedade antes de tomar uma decisão quanto ao abandono das investigações.

Para notificação de incidentes relacionados com *software* de rede, pode-se utilizar o endereço do CERT http://www.cert.org/contact_cert/contactinfo.html, desta forma a divulgação da vulnerabilidade pode ocasionar a prevenção de administradores atentos. Criado em 1988, o CERT Centro de Coordenação (CERT/CC) é um centro especializado em segurança na Internet, localizado no Instituto de Engenharia de *Software*, uma fundação federal de pesquisa e desenvolvimento da Universidade *Carnegie Mellon* (CERT, 2003).

No Brasil, o CAIS, Centro de Atendimento a Incidentes de Segurança

ligado a RNP Rede Nacional de Pesquisa, atua na detecção, resolução e prevenção de incidentes de segurança na rede acadêmica brasileira, além de elaborar, promover e disseminar práticas de segurança em redes. De forma geral, as informações disponíveis são aplicáveis às redes comerciais.

Entre as atividades do CAIS estão: atendimento a incidentes de segurança, coordenação com grupos de segurança já existentes, fomento à criação de novos grupos de segurança no país, disseminação de informações na área de segurança em redes, divulgação de recomendações e alertas, testes e recomendação de ferramentas de segurança, recomendação de políticas para a RNP, recomendação de políticas para os PoPs , recomendação de políticas para o *backbone* da RNP (CAIS, 2003).

2.8 MEDIDAS PRÉ INCIDENTE

A detecção de um incidente de segurança pode ser realizada de várias formas, mas uma das principais é através da monitoração contínua dos sistemas e conexões das redes à procura de desvio no padrão de funcionamento e/ou ações maliciosas.

2.8.1 PROCEDIMENTOS DE RESPOSTA AO INCIDENTE

Nesse tópico serão abordados alguns procedimentos fundamentais que irão facilitar e tornar mais ágil o processo de resposta. Para tanto é fundamental uma visão clara dos impactos que uma intrusão pode causar à continuidade do funcionamento da organização, seja ela financeira, comercial, governamental ou educacional.

Cada organização possui diferentes níveis de requisitos. Ações classificadas como graves para uma organização financeira podem ter menor impacto em uma organização acadêmica. Por exemplo, um servidor WWW que disponibilize informações sobre o resultado do vestibular. Antes da apuração dos

resultados, a indisponibilidade desse servidor pode até não ser percebida pela comunidade, mas na semana de divulgação dos resultados, torna-se crítica a indisponibilidade do mesmo. Já para uma organização financeira a qualquer momento a indisponibilidade dos servidores de Internet *Bank* torna-se crítico.

De modo a auxiliar no processo de mapeamento e pontuação do grau de criticidade dos recursos, segue alguns pontos que devem ser observados (Francisco, 2003):

- Identificação dos ativos a serem protegidos:
  - Dados - Quanto a confidencialidade, integridade e disponibilidade;
  - Recursos - Quanto à má utilização e indisponibilidade;
  - Reputação – Imagem e credibilidade.
- Identificação dos possíveis atacantes:
  - Concorrentes;
  - Funcionários;
  - Ex-funcionários;
  - Pessoal terceirizado;
  - Visitantes;
  - Vândalos;
  - Espiões;
- Identificação dos tipos de ataque:
  - DoS;
  - Roubo de Informação;
  - Erros;
  - Acidentes;
  - Engenharia social;
  - Etc.
- Classificação dos ativos quanto ao grau de impacto para o negócio, caso seja comprometido;

- Elaboração da documentação da rede e arquitetura atual, a documentação deve conter dados como:
    - Sistemas operacionais utilizados;
    - Versão do sistema operacional;
    - Fornecedor;
    - Serviços habilitados;
    - Responsáveis pelo sistema;
    - Outros dados.
- Definição dos ativos mais importantes para a organização;
- Definição do plano de contingência para o caso da ocorrência de um incidente de segurança;
- Definição de diretivas quanto ao monitoramento da rede e sistemas (e-mail, sites acessados, programas utilizados);
- Definição de diretivas referentes à propriedade intelectual;
- Identificar instituições federais, internacionais e os profissionais especializados, bem como estabelecer acordo de cooperação.

Este é um procedimento importante para ações coordenadas, pois durante uma situação de crise o tempo de resposta é um fator crucial e se existem contatos já estabelecidos com outras organizações sabe-se quem procurar e principalmente em quem confiar. Abaixo são apresentadas algumas instituições que possuem equipes de resposta a incidente:

- Contato com órgãos federais:
    - FAPESP;
    - RNP;
    - DPF.
- Contato com empresas privadas de *Backbone*:
    - EMBRATEL;

- Telecom em geral.

- Contato com profissionais de segurança. Preferencialmente profissionais especializados da organização e/ou de empresas de grande credibilidade e experiência;
- Instituições Internacionais:
  - CERT
  - FIRST
  - I-4 (só Estados Unidos)

Entretanto, não se deve limitar os contatos a estas organizações, deve-se procurar administradores de outras redes, de provedores, enfim, montar uma relação de confiança com várias organizações, definindo chaves de criptografia, canais seguros de comunicação, telefone de emergência entre outros.

Os pontos relacionados acima auxiliam na identificação e classificação dos ativos quanto ao grau de importância para o funcionamento da organização, bem como a identificação de possíveis grupos atacantes. Os fatores que levam a uma ação de intrusão são vários e os mesmos devem ser analisados de acordo com o ramo de trabalho da organização.

De acordo com o grau de criticidade, deve-se definir uma estratégia de proteção dos ativos. Uma regra universal para definição dos investimentos é "o valor investido na proteção do ativo não deve ser maior que o valor do mesmo", na definição dos valores deve-se levar em consideração fatores, tais como: tempo de recuperação, impacto no funcionamento da organização, entre outros.

Pode-se implementar uma estratégia de proteção baseada nos seguintes pontos:

- Camadas de proteção:
  - Nível de rede;
  - Nível de *host*;
- Definição de níveis de privilégios;

- Rastreamento / identificação e eliminação de pontos vulneráveis;
- Definir participação dos usuários:
  - Voluntária;
  - Obrigatória;

Outra característica da estratégia de proteção diz respeito à forma de implementação:

- Diversidade de soluções;
- Simplicidade da solução;

Cada abordagem possui pontos positivos e pontos negativos, e sempre se deve observar o grau de especialização da equipe, os investimentos em qualificação, nível de satisfação da equipe técnica, recursos disponíveis.

Montar uma equipe de resposta a incidente não é tarefa fácil, deve-se identificar dentro da organização profissionais com diferentes perfis e acioná-los no caso de ataque e esses profissionais devem antes passar por treinamento para familiarização dos procedimentos de resposta a incidentes e entendimentos das ações e hierarquias a serem respeitadas. A equipe deve contar com profissionais multidisciplinares e/ou com especialidade tais como: criptografia, banco de dados, TCP/IP, *Firewall*, IDS, elementos de conectividade, plataformas variadas, estenografia, estrutura de arquivos.

Em conjunto com outros setores a equipe de resposta a incidente deve classificar os mesmos quanto: gravidade, prioridade, valor do ativo e técnica utilizada.

Esta equipe deve ser responsável entre outras funções pela elaboração de matrizes de valores que devem conter dados como:

- Histórico de eventos;
- Estatísticas:
  - Por hora;

- Por dia;
- Por tipo de evento;
- Por número de ocorrência;
- Por origem - destino;

Também é importante que uma equipe de resposta a incidente deve ter direito de acesso a setores, recursos e dados de forma a poder executar seu trabalho.

Todos os envolvidos devem conhecer a hierarquia da chefia da organização e respeitá-la. Deve estar claro quem é o interlocutor entre a equipe de resposta e a direção da instituição, e em nenhum momento membros da equipe devem comentar as ações com outras pessoas, sobre o risco de comprometer a continuidade dos trabalhos, pois o incidente pode ter origem interna. Cada organização possui sua própria estrutura hierárquica de chefia e de acordo com a definição de gravidade do incidente, deve-se notificar ao superior sobre o incidente. Porém, não seria interessante notificar o presidente da organização sobre todo *port scan* que venha a ocorrer. Dependendo da gravidade do incidente de segurança, a “relação pública” deve passar para a direção de outros níveis o estado dos acontecimentos e elaborando memorandos internos e externos de esclarecimento, se for o caso.

3 MEDIDAS DE SEGURANÇA

Foram tratadas medidas basicamente procedimentais, mas faz-se necessário um conjunto de medidas físicas, tais como (NIC BR, 2003):

- Definição da política de segurança e política de uso aceitável;
- Elaboração de uma arquitetura de rede segura;
- Elevação do nível de segurança dos *hosts*;
- Monitoração contínua do tráfego da rede e dos serviços;
- Definição de testes periódicos à procura de vulnerabilidades.

A seguir serão apresentadas algumas medidas físicas possíveis e as adotadas no estudo de caso com resultados e comentários quando for o caso.

3.1 POLÍTICAS

3.1.1 POLÍTICAS DE SEGURANÇA

Uma política de segurança é um instrumento importante para proteger a sua organização contra ameaça à segurança da informação que a ela pertence ou que está sob sua responsabilidade. Uma ameaça a segurança é compreendida neste contexto como a quebra de uma ou mais de suas três propriedades fundamentais (confidencialidade, integridade e disponibilidade).

A política de segurança não define procedimentos específicos de manipulação e proteção da informação, mas atribui direitos e responsabilidades às pessoas (usuários, administradores de redes e sistemas, funcionários, gerentes, etc.) que lidam com essa informação. Desta forma, elas sabem quais as expectativas que podem ter e quais são as suas atribuições em relação à segurança dos recursos computacionais com os quais trabalham.

Além disso, a política de segurança também estipula as penalidades às quais estão sujeitos aqueles que a descumprem. Antes que a política de segurança seja escrita, é necessário definir as informações a serem protegidas. Usualmente, isso é feito através de uma análise de riscos, que identifica (NIC BR, 2003):

- Recursos protegidos pela política;
- Ameaças às quais estes recursos estão sujeitos;
- Vulnerabilidades que podem viabilizar a concretização destas ameaças, analisando-as individualmente.

Uma política de segurança deve cobrir os seguintes aspectos:

- Aspectos preliminares:

– abrangência e escopo de atuação da política;

– definições fundamentais;

– normas e regulamentos aos quais a política está subordinada;

– quem tem autoridade para sancionar, implementar e fiscalizar o cumprimento da política;

– meios de distribuições da política;

– como e com que freqüência a política é revisada.

- Política de senhas:

– requisitos para formação de senhas;

– período de validade das senhas;

– normas para proteção de senhas;

– reuso de senhas;

– senhas padrões.

- Direitos e responsabilidades dos usuários, tais como:

– utilização de contas de acesso;

– utilização de *software* e informações, incluindo questões de instalação,

licenciamento e copyright;

– proteção e uso de informações (sensíveis ou não), como senhas, dados de configuração de sistemas e dados confidenciais da organização;

– uso aceitável de recursos como *e-mail*, *news* e páginas *Web*;

– direito à privacidade, e condições nas quais esse direito pode ser violado pelo provedor dos recursos (a organização);

– uso de antivírus.

- Direitos e responsabilidades do provedor dos recursos, como:

– *backups*;

– diretrizes para configuração e instalação de sistemas e equipamentos de rede;

– autoridade para conceder e revogar autorizações de acesso, conectar e desconectar sistemas e equipamentos de rede, alocar e registrar endereços e nomes de sistemas e equipamentos;

– monitoramento de sistemas e equipamentos de rede;

– normas de segurança física.

- Ações previstas em caso de violação da política:

– diretrizes para tratamento e resposta de incidentes de segurança;

– penalidades cabíveis.

A lista de tópicos acima não é exaustiva nem tampouco se aplica a todos os casos. Cada organização possui um ambiente distinto e os seus próprios requisitos de segurança, e deve, portanto, desenvolver uma política de segurança que se molde a essas peculiaridades. É recomendável, por exemplo, que organizações que possuam uma rede *wireless* incorporem uma política específica para esse tipo de rede à sua política de segurança.

Alguns fatores importantes para o sucesso de uma política de segurança são:

- Apoio por parte da administração superior;

- A política deve ser ampla, cobrindo todos os aspectos que envolvem a segurança dos recursos computacionais e da informação sob responsabilidade da organização;
- A política deve ser periodicamente atualizada de forma a refletir as mudanças na organização;
- Deve haver um indivíduo ou grupo responsável por verificar se a política está sendo respeitada;
- Todos os usuários da organização devem tomar conhecimento da política e manifestar a sua concordância em submeter-se a ela antes de obter acesso aos recursos computacionais;
- A política deve estar disponível em um local de fácil acesso aos usuários, tal como a intranet da organização.

Dentre os itens acima, o apoio por parte da administração superior é essencial. Se a política de segurança não for encampada pela administração, ela rapidamente será deixada de lado pelos demais setores da organização. Além disso, é importante que os seus membros dêem o exemplo no que diz respeito à observância da política de segurança.

Entre os fatores que influem negativamente na aceitação de uma política de segurança e que podem levá-la ao fracasso estão: política demasiadamente detalhada ou restritiva; criação de exceções para indivíduos ou grupos e a política não deve estar atrelada a *software* e/ou *hardware* específicos.

3.1.2 POLÍTICAS DE USO

A política de uso aceitável (AUP—*Acceptable Use Policy*) é o documento que define como os recursos computacionais da organização podem ser utilizados. Ela deve ser pública e estar disponível a todos os que utilizam a infra-estrutura computacional da organização, sendo recomendável que a autorização para o uso dos recursos seja condicionada a uma concordância

expressa com os seus termos.

A AUP é geralmente parte integrante da política de segurança global. Para muitas organizações, ela será composta pelos itens da política que afetam diretamente os usuários de recursos computacionais, principalmente os que definem seus direitos e responsabilidades.

Por outro lado, organizações que oferecem acesso a usuários externos (tais como provedores de acesso Internet) devem definir uma política de uso aceitável para esses usuários que seja independente da AUP à qual estão sujeitos os seus usuários internos. É importante que os usuários externos tomem conhecimento dessa política e saibam que o uso dos recursos está condicionado ao seu cumprimento.

3.2 ARQUITETURA DE REDE SEGURA

A arquitetura de rede segura é constituída do projeto físico e implementação de técnicas que visam aumentar a segurança da rede interna, esse conjunto é chamado de *Firewall*. Entretanto *firewalls* não são mecanismos exclusivos para proteger a rede interna da rede externa (que pode ser qualquer rede, a Internet é apenas o exemplo mais significativo de redes). Atuantes como barreiras de segurança, *firewalls* são úteis em qualquer ponto estratégico às redes ou sub-redes. Em algumas situações as organizações podem necessitar proteger partes da rede interna de outras partes da mesma rede corporativa. Nesse caso, pode-se utilizar *firewalls* internos configurados de forma apropriada à segurança interna.

Existem componentes básicos com os quais se pode construir uma infinidade de arquiteturas de *firewall*. Serão apresentadas algumas arquiteturas e características de *firewall* mais usuais e no tópico de desenvolvimento será apresentada a arquitetura escolhida para o estudo de caso.

"*Antigamente, paredes de tijolos eram construídas entre construções em*

complexos de apartamentos de forma que se ocorresse um incêndio ele não poderia se espalhar de uma construção para a outra. De uma forma completamente natural, as paredes foram chamadas de firewall" (Siyan & Hare, 1995). Em redes de computadores, *firewalls* são barreiras interpostas entre a rede privada e a rede externa com a finalidade de reduzir os riscos de intrusos (ataques); ou seja, são mecanismos (dispositivos) de segurança que protegem os recursos de *hardware* e *software* da organização dos perigos (ameaças) aos quais o sistema está exposto. Estes mecanismos de segurança são baseados em *hardware* e *software* e seguem a política de segurança estabelecida pela organização.

3.2.1 ATRIBUIÇÕES DO FIREWALL

Entre as tarefas cabíveis a um *firewall*:

- *Checkpoint*; ou seja, ele é um foco para as decisões referentes à segurança, é o ponto de conexão com o mundo externo, tudo o que chega à rede interna passa pelo *firewall*;
- Aplicar a política de segurança;
- Logar eficientemente as atividades da e para Internet;
- Limitar a exposição da empresa ao mundo externo;
- Proteger a rede contra vírus.

Tarefas que um *firewall* não pode realizar:

- Proteger a empresa contra usuários internos mal intencionados;
- Proteger a empresa de conexões que não passam por ele;
- Proteger contra ameaças completamente novas como vulnerabilidade e vírus.

3.2.2 ESTRATÉGIAS DE SEGURANÇA

Basicamente, existem algumas estratégias de segurança. Estas estratégias não são exclusivas de ambientes de sistemas de computação, são estratégias de segurança de uma forma geral, mas são muito úteis quando consideradas em toda a sua extensão. As estratégias para isso serão abordadas nas próximas subseções.

3.2.2.1 LEAST PRIVILEGE

"Basicamente, o princípio do mínimo privilégio significa que qualquer objeto (usuário, administrador, programa, sistema, etc) deveria ter somente os privilégios que o objeto precisa para realizar as suas tarefas e nada mais". (Chapman et al, 1995)

Mínimo privilégio é um princípio importante para limitar a exposição aos ataques e para limitar os danos causados por ataques. Deve-se explorar meios para reduzir os privilégios requeridos para as operações.

3.2.2.2 CHOKE POINT

Esta estratégia força os atacantes a utilizarem um “canal estreito”, o qual pode ser monitorado e controlado. O *firewall* quando é o único ponto de acesso a rede privada, constitui-se em um *choke point* porque os atacantes necessariamente devem passar por ele.

Todos os *choke points* devem ser amplamente conhecidos e monitorados. Não adianta ter um ótimo *firewall* se ela permite que funcionários conectados a uma rede TCP/IP interna acessem a Internet via linhas discadas. Caso isto ocorra, os atacantes terão portas abertas sem precisar enfrentar a grande muralha

do *firewall*.

3.2.2.3 DEFENSE IN DEPTH

De acordo com este princípio, não se deve depender de apenas um mecanismo de segurança não importando quão forte ele pareça ser. Ao invés disso, recomenda-se que sejam utilizados múltiplos mecanismos de segurança e que estes estejam configurados no nível mais alto possível de segurança e redundância. A estratégia principal é fazer com que o ataque seja significativamente arriscado e caro ao atacante que se espera encontrar.

Um exemplo prático ocorre quando se utiliza dois roteadores, um externo conectado diretamente a Internet e um interno conectado diretamente a rede privada e entre eles um *bastion host*. Nesse caso, esta estratégia poderia ser empregada utilizando redundância em ambos roteadores, aplicando-se ao roteador interno também as regras de filtragem adotadas no roteador externo. Desta forma, caso um pacote que deveria ser barrado no primeiro roteador chegasse ao segundo, isso indicaria que o primeiro roteador foi atacado com sucesso e, conseqüentemente, um alarme poderia ser acionado a fim de que medidas sejam tomadas para solucionar o problema. Esta estratégia permite que o sistema tolere mais falhas na segurança.

3.2.2.4 WEAKEST LINK

"Uma corrente é tão segura quão a mais fraca de suas argolas e uma parede é tão forte quão o seu ponto mais fraco". Esta é uma noção fundamental à segurança dos sistemas de computação. Deve-se eliminar os pontos fracos, observando-se que o perigo ainda é maior quando uma ligação fraca também é um *choke point*.

3.2.2.5 FAIL SAFE

A Falha Segura estabelece que se um sistema falhar ele falha de tal forma que é negado o acesso ao atacante, não permitindo que ele entre. A falha também pode resultar que os usuários legítimos não tenham acesso até que os reparos sejam realizados, mas isso é bem melhor do que permitir que alguém (o intruso) entre e destrua tudo.

Por exemplo, se um programa *proxy* falhar devido a um ataque, apenas este programa falhará, impedindo que o atacante prossiga a sua investida. *Type enforcement* é uma técnica utilizada para implementar sistemas *fail safe*, permitindo programas *proxy* com tais características. Se um *packet filter* configurável em modo *fail safe* falhar, tanto os pacotes do intruso como os dos demais usuários não poderão trafegar, mas pelo menos o ataque não se alastrará para mais adiante.

A principal aplicação desse princípio em segurança de redes está em escolher a postura da organização com respeito a segurança. As duas principais posturas são:

- Postura padrão de negação: especificar apenas o que é permitido e proibir o resto;
- Postura padrão de permissão: especificar somente o que é proibido e permitir o resto.

Esta postura padrão de negação é uma postura *fail safe*. Entretanto, nem sempre a política de segurança agrada aos usuários do sistema. É necessário que se estabeleça uma relação amistosa entre o pessoal da administração da segurança e os usuários, esclarecendo esses acerca das medidas tomadas. Para determinar os serviços que serão permitidos, aconselha-se seguir os seguintes passos:

- Identificar os serviços que os usuários precisam;

- Considerar as implicações destes serviços na segurança e como se pode provê-los seguramente;
- Permitir somente os serviços que se compreendem, os quais possam ser providos seguramente e que se consiga visualizar uma necessidade legítima para eles.

Esta postura padrão de permissão certamente não se enquadra como *fail safe*. Os únicos que de fato se beneficiam desta postura são os intrusos, porque o administrador do *firewall* não pode tapar todos os buracos nos serviços disponíveis e naqueles que surgem no correr do tempo. Enquanto ele está ocupado tentando resolver alguns problemas com alguns serviços os atacantes se deliciam atacando em outros pontos.

3.2.2.6 UNIVERSAL PARTICIPATION

É necessário que os usuários da rede tenham consciência da necessidade dos mecanismos de segurança adotados. Esta consciência deve ser construída de uma forma polida, mostrando ao usuário que as medidas vêm em benefício deles e de toda a corporação. Para que haja participação, ou pelo menos não haja oposição, a compreensão dos usuários deve ser conquistada de forma voluntária e não involuntária. Qualquer inimigo dentro da empresa pode ser a grande brecha na segurança da corporação: por exemplo, um usuário revoltado com os meios adotados para impor a política de segurança, resolve se conectar à Internet via linha discada utilizando um protocolo tipo PPP (*Point to Point Protocol*) ou SLIP (*Serial Line Internet Protocol*).

3.2.2.7 DIVERSITY OF DEFENSE

Através da estratégia *defense in depth* se acrescenta segurança através de um certo número de diferentes sistemas (um *packet filter*, mais um *application host gateway*, mais outro *packet filter*, etc). Analogamente, com a estratégia de *diversity of defense*, adiciona-se segurança utilizando um número de diferentes tipos de sistemas.

A idéia por detrás desta estratégia é que utilizando sistemas de segurança de diferentes fornecedores pode reduzir as chances de existir um bug ou erro de configuração comum aos diferentes sistemas. No exemplo citado anteriormente, caso os dois *packet filters* utilizados sejam do mesmo tipo, fica mais fácil para o atacante, pois ele poderá explorar o mesmo problema da mesma forma em ambos.

É importante que se tenha cuidado com diferentes sistemas configurados pela mesma pessoa (ou grupo de pessoas), pois é provável que os mesmos erros cometidos em um sistema estejam presentes nos outros devido a compreensão conceitual errada sobre alguns aspectos.

3.2.2.8 SIMPLICIDADE

Simplicidade é uma estratégia de segurança por duas razões. Coisas simples são mais fáceis de serem compreendidas (se não se compreende alguma coisa, não se pode realmente saber se ela é ou não segura) e a complexidade proporciona esconderijos de todos os tipos de coisas (é mais fácil proteger uma sala do que todo um prédio).

3.2.2.9 TYPE ENFORCEMENT

"*Type enforcement* é um mecanismo de segurança que atribui a todo programa no sistema permissão para fazer somente aquelas coisas que ele precisa fazer para executar o seu trabalho. Isto é chamado *least privilege* e aplica a programas, arquivos e Sistemas Operacionais" (Thomsen ,1996). Entretanto, este mecanismo também aborda as estratégias *defense in depth* e *fail safe* porque caso algum programa seja comprometido proposital ou acidentalmente a falha não se propagará além do ambiente restrito onde o programa estiver executando; ou seja, além de se poder investir na segurança do próprio serviço, tem-se a redundância na segurança a nível de sistema operacional e se garante que a falha não permitirá maiores danos ao sistema.

3.2.3 ARQUITETURAS DE FIREWALL

O *Firewall* consiste em um conjunto de componentes organizados de uma forma a garantir certos requisitos de segurança. Os componentes básicos para a construção de um *firewall* são:

Packet Filters: são responsáveis pela filtragem (exame) dos pacotes que trafegam entre dois segmentos de rede.

Bastion Host: computador responsável pela segurança de um ou mais recursos (serviços) da rede.

Determinadas arquiteturas recebem denominações especiais e servem como referência para a construção de uma infinidade de variantes. As arquiteturas *screened subnet* e *screened host* podem ser consideradas clássicas. Destacam-se porque são resultantes de uma disposição básica dos componentes *packet filter* e *bastion host* (Spohn, 1996).

3.2.3.1 PACKET FILTERS

Como um primeiro passo ao se implementar uma barreira de segurança em uma rede de computadores, é fundamental que se conheça os detalhes dos protocolos de comunicação utilizados. Na Internet, a atenção deve ser voltada aos protocolos IP, TCP, ICMP e UDP. Esses são os principais protocolos em nível de rede e transporte que são considerados e examinados ao se estabelecer regras de filtragem em um *packet filter* para a Internet. Esse mecanismo de filtragem no nível de roteador possibilita que se controle o tipo de tráfego de rede que pode existir em qualquer segmento de rede; conseqüentemente, pode-se controlar o tipo de serviços que podem existir no segmento de rede. Serviços que comprometem a segurança da rede podem, portanto, ser restringidos.

Com o exposto acima, fica evidente que um *packet filter* não se encarrega de examinar nenhum protocolo de nível superior ao nível de transporte, como por exemplo, o nível de aplicação que fica como tarefa dos *application gateways* (*proxy servers*). Portanto, qualquer falha de segurança no nível de aplicação não pode ser evitada utilizando somente um *packet filter*.

O componente que realiza a filtragem de pacotes geralmente é um roteador dedicado, mas também pode ser um *host* de propósito geral configurado como roteador, e recebe a denominação de *screening router* (figura 7). Deve-se ressaltar que o processo de filtragem de pacotes acarreta um overhead ao sistema; portanto, para uma situação de alto tráfego é necessário que se utilize um roteador com uma velocidade de processamento compatível com as necessidades.

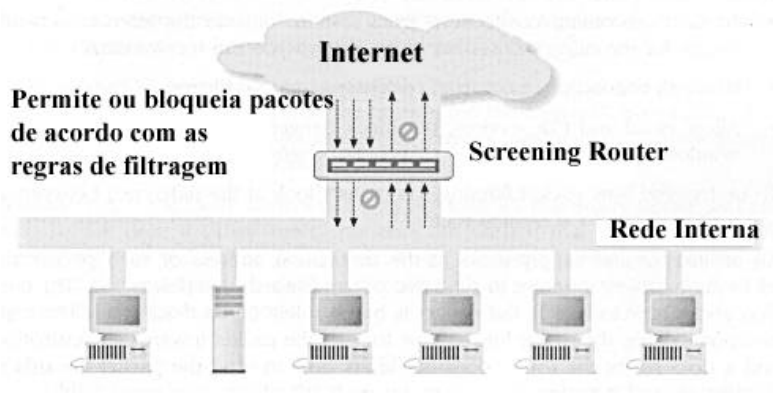


Figura 7 – Esquema de atuação de um *Screening Router*.

Frisando que a filtragem dos pacotes não considera protocolos acima do nível de transporte, não é tomada nenhuma decisão baseada no conteúdo dos pacotes; ou seja, nos dados dos pacotes propriamente ditos. A filtragem que a maioria dos *screening routers* realizam são baseadas nas seguintes informações:

- Endereço IP fonte;
- Endereço IP destino;
- Protocolo: Se o pacote é TCP, UDP ou ICMP;
- Portas TCP ou UDP fontes;
- Portas TCP ou UDP destino;
- Tipo de mensagem ICMP (se for o caso).

No protocolo TCP existe um *flag* denominado ACK que é utilizado para confirmação de pacotes e também pode ser utilizado para detectar se o pacote é o primeiro de uma solicitação de conexão. Quando o *flag* não estiver setado significa que o pacote se refere a uma solicitação de conexão e, caso contrário, o pacote corresponde a alguma conexão já existente (figura 8). Desta forma, o *packet filter* pode bloquear um serviço *inbound* (de fora para dentro; ou seja, o servidor está na rede interna) apenas não permitindo o fluxo de pacotes com o ACK setado destinado a um servidor interno associado a porta (por exemplo, a

port 23 do telnet) do serviço bloqueado. Em protocolos não orientados a conexão, por exemplo, o protocolo UDP, não é possível tomar nenhuma decisão desse tipo; ou seja, nestes protocolos, nunca se sabe se o pacote que está chegando é o primeiro que o servidor está recebendo. Para fazer uma filtragem correta dos pacotes, é importante saber se o protocolo é bidirecional (pacotes fluem nos dois sentidos, cliente para servidor e vice-versa) ou unidirecional. Não se pode confundir serviços *inbound* (a rede interna provendo algum serviço) e serviços *outbound* (o cliente está na rede interna e o servidor na Internet) com pacotes *inbound* (pacotes que chegam na rede interna) e pacotes *outbound* (pacotes que saem da rede interna); ou seja, ambos os serviços apresentam pacotes *inbound* e *outbound* caso o protocolo seja bidirecional.

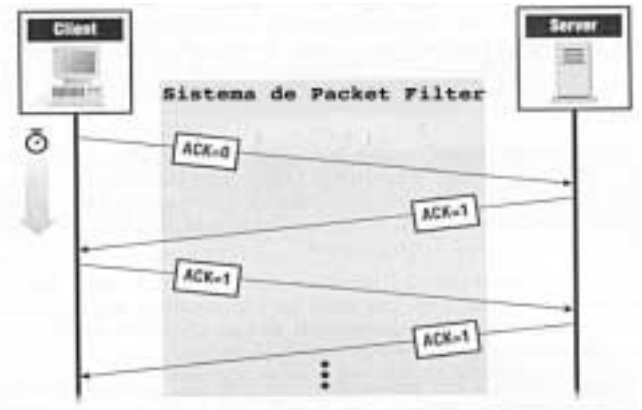


Figura 8 – Esquema do ACK bit no protocolo TCP entre cliente e servidor.

É importante que o roteador tenha facilidades de filtragem por interfaces de rede. Ou seja, todas as interfaces disponíveis no roteador são submetidas às regras de filtragem, possibilitando que as regras sejam aplicadas considerando as seguintes informações:

- A interface na qual o pacote chega;
- A interface pela qual o pacote sai.

O IP Spoofing (figura 9) é um ataque que pode ser evitado com a

aplicação do recurso exposto acima. Nesse ataque o intruso tenta se passar como um *host* interno (um *host* considerado confiável) utilizando o endereço IP desse como o endereço fonte. Se a filtragem é realizada por interface em ambos os sentidos, esse ataque não funciona porque jamais um pacote pode chegar do mundo externo (Internet) tendo como endereço fonte o endereço de uma máquina que está na rede interna; ou seja, só poderia chegar naquela interface no outro sentido. Com isto uma simples regra pode evitar muita dor de cabeça.

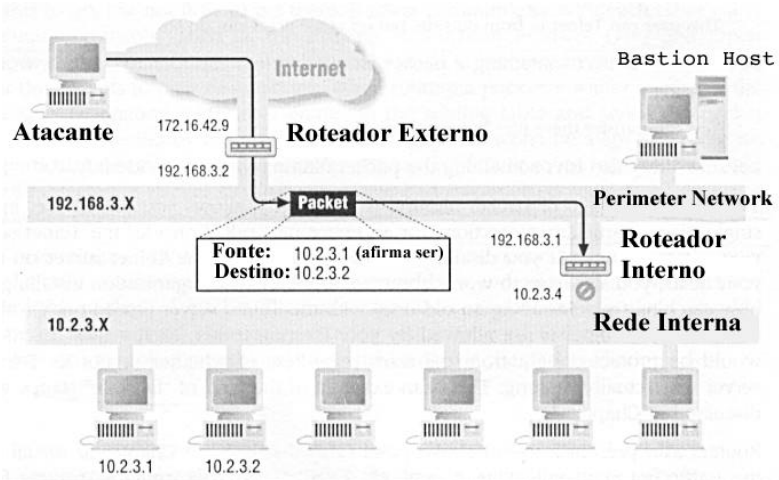


Figura 9 – Esquema do ataque do tipo IP *spoofing*.

A seguir são apresentados alguns exemplos de regras de filtragem que poderiam ser aplicadas em um *screening router* com postura padrão de negação:

- Permitir todas as solicitações de conexão de *hosts* da rede interna conectada a interface interna do roteador para *hosts* externos com conexões SMTP (Porta TCP número 25), HTTP (Porta TCP número 80), DNS (Porta UDP número 53), FTP (Portas TCP números 20 e 21), etc;
- Bloquear todas as conexões para e de certos sites considerados não confiáveis;
- Bloquear todas as solicitações de conexão de *hosts* da rede interna conectada a interface externa do roteador;

- Bloquear pacotes do protocolo ICMP;

A sintaxe dessas regras de filtragem depende do produto (roteador). É aconselhável montar uma tabela com as possibilidades de pacotes para cada serviço a ser provido. Veja um exemplo na tabela 2. Descrição dos campos da tabela:

Direção	End. Fonte	End. Dest.	Protocolo	Porta Fonte	Porta Dest.	ACK	Observação
In	Ext	Int	UDP	>1023	53	a	Consulta incoming via UDP, cliente para servidor interno.
Out	Int	Ext	UDP	53	>1023	a	Resposta para consulta incoming UDP, servidor para cliente
In	Ext	Int	TCP	53	>1023	b	Consulta incoming via TCP, cliente para servidor interno
Out	Int	Ext	TCP	53	>1023	sim	Resposta para consulta incoming TCP, servidor para cliente.
Out	Int	Ext	UDP	>1023	53	a	Consulta outgoing via UDP, cliente para servidor externo.
In	Ext	Int	UDP	53	>1023	a	Resposta para consulta outgoing UDP, cliente para servidor interno.
Out	Int	Ext	TCP	>1023	53	a	Consulta outgoing via TCP, cliente para servidor externo.
In	Ext	Int	TCP	53	>1023	sim	Resposta para consulta outgoing TCP, servidor para cliente interno.
In	Ext	Int	TCP	>1023	53	b	Consulta de servidor externo para servidor interno. Transferência de zona do servidor secundário externo.
Out	Int	Ext	TCP	53	>1023	sim	Resposta do servidor interno. Transferência de zona para servidor secundário externo.
Out	Int	Ext	TCP	>1023	53	b	Consulta do servidor interno para servidor externo.
In	Ext	Int	TCP	53	>1023	sim	Resposta do servidor externo para o servidor interno.

a = pacotes UDP que não têm ACK bit.
b = ACK não é setado no primeiro pacote deste tipo (estabelecendo conexão), mas pode ser setado no restante.

Tabela 2 – Exemplo da esquematização de Regras de filtragem (protocolo DNS)

3.2.3.1.1 OPERAÇÕES DE UM PACKET FILTER

Quase todos os dispositivos de filtragem de pacotes operam da seguinte maneira (Siyan, 1995):

1. Os critérios de filtragem de pacotes devem ser armazenados para as portas do dispositivo de filtragem de pacotes. Os critérios de filtragem de pacotes são chamados “regras de filtragem de pacotes”.
2. Quando o pacote chega em uma porta, os cabeçalhos do pacote são analisados. Muitos dispositivos examinam os campos somente nos cabeçalhos dos protocolos IP, TCP ou UDP.
3. As regras de filtragem são armazenadas em uma ordem específica. Cada regra é aplicada ao pacote na ordem em que as regras estão armazenadas.
4. Se uma regra bloqueia a transmissão ou recepção do pacote, o pacote é bloqueado.
5. Se uma regra permite a transmissão ou recepção do pacote, o pacote é aceito para prosseguir.
6. Se um pacote não satisfaz qualquer regra e tendo como política padrão bloqueio, então ele é bloqueado.

Pelas regras 4 e 5 fica evidente que a ordem das regras de filtragem é de fundamental importância. Uma ordenação incorreta das regras pode acarretar em bloqueio de serviços válidos e em permissão de serviços que deveriam ser negados. Da regra 6 segue a filosofia "O que não é expressamente permitido é proibido".

3.2.3.1.2 VANTAGENS E DESVANTAGENS DO PACKET FILTER

Algumas vantagens dos *packet filters* são:

- Pode ajudar a proteger toda uma rede, principalmente se este é o único

roteador que conecta a rede interna à Internet;

- A filtragem de pacotes é transparente e não requer conhecimento nem cooperação dos usuários;
- Está disponível em muitos roteadores.
- Algumas desvantagens são:
- Alguns protocolos não são bem adaptados para a filtragem;
- Algumas políticas não podem ser aplicadas somente com a filtragem de pacotes.

Quando se aplica alguma restrição em algum protocolo de mais alto nível, através de números de portas, espera-se que nada além do próprio serviço esteja associado àquela porta; entretanto, usuários internos mal intencionados podem subverter este tipo de controle colocando outro programa (desenvolvido ou alterado por ele) associado a essa porta. Como citado anteriormente, um *firewall* não é apropriado para se defender de ameaças internas.

3.2.3.1.3 AÇÕES DO SCREENING ROUTER

O roteador encarregado da filtragem dos pacotes pode executar uma série de atividades que servem, entre outras coisas, para monitorar o sistema. Algumas atividades são:

- Realizar *logs* de acordo com a configuração especificada pelo administrador. Dessa forma, é possível analisar eventuais tentativas de ataque, bem como verificar a correta operação do sistema;
- Retorno de mensagens de erros ICMP: caso um pacote seja barrado existe a possibilidade de se enviar ao endereço fonte alguma mensagem com o código de erro ICMP do tipo *host unreachable* ou *host administratively unreachable*. Entretanto, tais mensagens, além de causar um *overhead*, podem fornecer algumas informações sobre o *packet filter* ao atacante, pois

dessa forma, ele poderia descobrir quais os protocolos que são barrados e quais estão disponíveis; portanto, recomenda-se que não se retorne nenhum código ICMP de erro para *hosts* na rede externa.

3.2.3.1.4 RISCOS NA FILTRAGEM

A filtragem por endereço fonte apresenta alguns riscos. Há dois tipos de ataques possíveis:

- *Source address*: o atacante forja o endereço fonte utilizando o endereço de uma máquina (externa ou interna) considerada confiável (*trusted*) pelo *firewall* (figura 9). Este ataque pode ter sucesso principalmente quando o atacante não precisa capturar (ou seja, estar em um caminho entre o *firewall* e o *host* forjado) nenhum pacote e quando, caso a máquina forjada seja interna, não houver mecanismos de filtragem que impeçam o IP *spoofing* (citado anteriormente);
- *Man in the middle*: além de forjar o endereço, nesse ataque o atacante deve estar no caminho entre o *firewall* e o *host* confiável porque ele tem de capturar os pacotes que são, na realidade, enviados ao *host* confiável (daí a denominação do ataque).

Muitos desses ataques só funcionam quando o *host* confiável (aquele cujo endereço é utilizado pelo atacante) estiver fora de operação, porque assim que ele receber algum pacote que não esteja relacionado com nenhuma conexão que ele tenha iniciado ele solicitará que a conexão forjada seja encerrada. Existem várias formas de se evitar que o *host* confiável tome conhecimento da conexão forjada pelo atacante, eis algumas maneiras:

- Confundindo o roteamento entre a máquina real (*host* confiável) e a máquina alvo;
- Utilizando um ataque onde somente a primeira resposta é requerida, de tal

forma que o *reset* solicitado pela máquina real não importará;

- Inundando a máquina real com pacotes lixo (por exemplo, pacotes ICMP) enquanto o ataque ocorre, de forma que a máquina real ficará ocupada tentando processar os pacotes lixo que ela recebe;
- Utilizando *source routing*.

A filtragem baseada na porta fonte apresenta um problema semelhante àquele da filtragem pelo endereço fonte. Assume-se que a uma determinada porta um determinado serviço esteja associado, mas nada impede que alguém com os devidos direitos substitua o servidor por outro. Para evitar esse tipo de ataque, deve-se garantir que o servidor seja confiável e execute somente o permitido; impedindo, de outra forma, que o cliente devidamente modificado possa solicitar alguma facilidade que comprometa o servidor. Portanto, é fundamental que tanto o servidor como também os clientes utilizados não sejam passíveis de serem alterados indevidamente por pessoas mal intencionadas. Vale salientar que uma solução plausível é a técnica de *type enforcement* (abordada anteriormente) implementada no nível de sistema operacional.

3.2.3.1.5 CARACTERÍSTICAS DESEJÁVEIS EM UM SCREENING ROUTER

Eis algumas características altamente desejáveis a fim de que se possa realizar uma filtragem de pacotes bem apurada:

- Ter uma boa performance na filtragem dos pacotes: um overhead aceitável de acordo com as necessidades;
- Pode ser um roteador dedicado ou um computador de propósito geral executando algum sistema de roteamento e filtragem;
- Permitir uma especificação de regras de forma simples;
- Permitir regras baseadas em qualquer cabeçalho ou critério *meta-packet* (por exemplo, em qual interface o pacote chegou ou está saindo);

- Aplicar as regras na ordem especificada;
- Aplicar as regras separadamente para pacotes que chegam e partem em e de cada interface de rede;
- Registrar informações sobre pacotes aceitos e rejeitados;
- Ter capacidade de teste e validação.

3.2.3.1.6 MÚLTIPLOS ROTEADORES

Em muitas configurações mais seguras, como a *screened subnet*, constatam-se que há pelo menos dois roteadores no *firewall*: um interno (entre a rede interna e a rede perimetral) e outro externo (entre a Internet e a rede perimetral). Esta é uma aplicação das estratégias *defense in depth* e *multiple defense* (caso os roteadores sejam produtos diferentes). Há também a situação em que se utiliza um único roteador, mas com múltiplas interfaces de rede (por exemplo: uma interface conectada na Internet, outra com a rede perimetral onde estão os *bastion hosts* e a outra conectada à rede interna).

Para cada roteador existente no *firewall* devem ser elaboradas as regras de filtragem baseadas na sua posição relativa dentro do *firewall* e, quando possível, não se deve poupar na redundância (caso um roteador falhe, o outro impedirá que a falha se propague além dos seus domínios).

3.2.4 BASTION HOST

Bastion host é qualquer máquina configurada para desempenhar algum papel crítico na segurança da rede interna; constituindo-se na presença pública na Internet, provendo os serviços permitidos segundo a política de segurança da organização.

Marcus Ranum é um dos responsáveis pela popularidade deste termo na comunidade profissional de *firewall*. Segundo ele "*bastions* são áreas críticas de

defesa, geralmente apresentando paredes fortes, salas para tropas extras, e o ocasional útil repositório de óleo quente para desencorajar os atacantes" (Chapman, 1995).

Um *bastion host* deve ter uma estrutura simples, de forma que seja fácil de garantir a segurança. É importante que se esteja preparado para o fato de que o *bastion host* seja comprometido, considerando que ele provavelmente (dependendo do site) será alvo de ataques.

O *bastion host* tem responsabilidades diferentes do *packet filter*, dependendo do seu tipo. Alguns autores enfatizam que enquanto o *packet filter* atua em um nível mais baixo, o *bastion host* se encarrega de todos os níveis (referentes ao modelo OSI). Na realidade, um *host* pode acumular tanto as funções de filtragem de pacotes como também pode prover alguns serviços; neste caso, ele seria um *packet filter* e *bastion host*, simultaneamente. Independentemente de qual seja a nomenclatura adotada, o que se deve ter em mente é o papel que estes dois componentes desempenham: filtragem e provedor de serviços.

Este tipo de máquina também recebe a denominação de application gateway porque funciona como um gateway no nível de aplicação. Os servidores disponíveis nos *bastion host* são denominados de *proxy servers*; ou seja, servidores por procuração que atuam como intermediários entre o cliente e o servidor. Neste caso, os serviços só podem ser providos via *bastion host*, obrigando o cliente (por exemplo, via regras de filtragem nos roteadores) a acessar estas máquinas; portanto, esse é um mecanismo que garante que o serviço será provido de forma segura para usuários internos e externos e impede que o *bastion host* seja desviado (a não ser que o roteador seja comprometido e as regras de filtragem alteradas).

3.2.4.1.1 TIPOS ESPECIAIS DE BASTION HOSTS

Dependendo da localização do *bastion host* dentro do *firewall*, tem-se alguns tipos de máquinas com funções diferenciadas na segurança, os quais são (Chapman, 1995):

- *Dual homed host*: trata-se de um computador com duas interfaces de rede conectadas cada uma a segmentos diferentes de rede. Uma das características fundamentais dessa configuração é que o roteamento direto (IP *forwarding*) é desabilitado e, portanto, todo o roteamento é realizado a nível de aplicação. Neste caso, todos os serviços segurados podem ser fornecidos via procuração (*proxy servers*) e somente o tráfego, referente aos serviços habilitados, via *proxy* e aqueles especificados pelas regras de filtragem circulam entre os dois segmentos de rede conectados ao *bastion host* (figura 10);
- *Victim machines*: estas máquinas abrigam serviços que não são considerados fáceis de serem segurados. A máquina é configurada basicamente somente com os serviços fornecidos para garantir que nada mais significativo esteja a disposição do atacante caso a máquina seja comprometida. Geralmente uma *screened subnet* possui uma ou mais máquinas deste tipo;
- *Internal bastion hosts*: são aquelas máquinas com maior interação com as máquinas internas (por exemplo, uma máquina que recebe o e-mail e o reenvia a um servidor de correio eletrônico residente na rede interna).

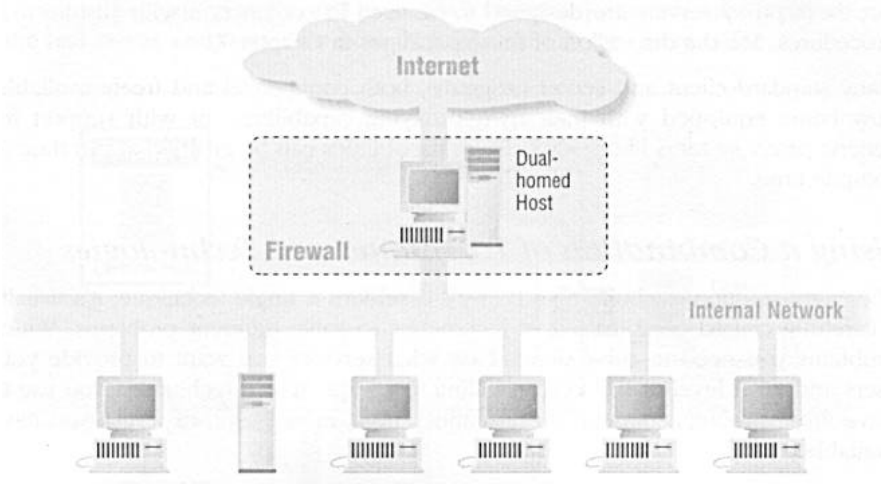


Figura 10 – Esquema da localização da *Dual-homed Host*.

Caso o *bastion host* esteja localizado junto a sub-rede interna (como na arquitetura *screened host*), ele pode ser facilmente desviado (*bypassed*) na ocorrência do roteador externo ser comprometido. O mesmo não ocorreria caso esse *bastion host* fosse do tipo *dual homed* e estivesse entre o roteador externo e a rede interna. Em Chapman, 1995, há a recomendação para não se utilizar a configuração do *dual homed host* fundida com o roteador interno porque todo o tráfego interno pode ser capturado caso o *bastion host* seja comprometido. Em contrapartida, a fusão do *bastion* com o roteador externo pode ser realizada sem que ocorram grandes problemas (caso exista um roteador interno).

3.2.4.1.2 CONSTRUINDO UM BASTION HOST

Uma vez definidos os serviços a serem providos pelo *bastion host*, configura-se a máquina de acordo com o tipo de *bastion host* necessário. Caso o serviço seja extremamente seguro, poder-se-ia provê-lo utilizando apenas filtragem de pacotes, mas é importante observar que abrir uma passagem para uma máquina interna para prover tal serviço significa tornar todas as

demais máquinas da sub-rede igualmente atingíveis caso o roteador seja comprometido. A solução é manter todos os serviços disponíveis à Internet em um ou mais *bastion hosts* e, caso o serviço seja considerado altamente inseguro, deve-se escolher uma máquina vítima para provê-lo.

Há alguns passos básicos para construir um *bastion host* (Chapman, 1995):

1. Tornar a máquina segura;
2. Desabilitar todos os serviços não desejados;
3. Instalar ou modificar os serviços que se deseja prover;
4. Rodar uma auditoria de segurança para estabelecer uma linha base;
5. Conectar a máquina na rede na qual ela será utilizada.

Uma máquina *dual homed* deve ter o roteamento direto desabilitado entre as duas interfaces. Para que isso seja feito, algumas vezes é necessário que o *kernel* do sistema operacional seja recompilado; entretanto, há algumas plataformas que permitem que essa configuração seja realizada de uma forma mais flexível sem que precise recompilar o sistema. Infelizmente, desabilitar o roteamento direto nem sempre é suficiente para desabilitar todos os recursos de roteamento da máquina. Em algumas plataformas como, por exemplo, as baseadas no Unix BSD (SunOS, Ultrix,etc), é possível desabilitar o roteamento direto mas geralmente permanece a opção de *source routing* habilitada. *Source routing* é um mecanismo de roteamento que consiste no seguinte: junto ao pacote, além das informações convencionais, envia-se a rota (as máquinas pelas quais deve passar) que o pacote deve seguir até o destino, de forma que as máquinas intermediárias por onde o pacote trafegar não utilizarão as suas potencialidades de roteador para definir a rota, seguindo estritamente a rota especificada no pacote. Se o *screening router* permitir, deve-se desabilitar pacotes *source routed*, aliviando o *bastion host* deste problema. Utilizando IP *spoofing* e *source routing* fica fácil ao atacante obter sucesso em suas investidas

(além de forjar o endereço ele garante que os pacotes seguirão direto para a sua máquina). Considerando que o mecanismo de *source routing* não é considerado necessário a nenhum serviço usual fornecido via Internet, a melhor solução é rejeitar qualquer pacote deste tipo no *choke point* do *firewall*; ou seja, no roteador externo.

Proxy Systems

Uma maneira de tornar seguro um serviço é não permitir que cliente e servidor interajam diretamente. *Proxy systems* são sistemas que atuam em nome do cliente de uma forma transparente. Os serviços *proxies* são implementações mais seguras que as convencionais, provendo apenas as facilidades necessárias para fornecer o serviço. Estes procuradores residem em algum *bastion host* no *firewall*. Para que esses *hosts* não sejam desviados (*bypassed*) é necessário que seja utilizado em conjunto um *packet filter*, de forma que esse force o tráfego (dos serviços via procuração) através do *bastion host*, ou então se utiliza um *dual homed host* como servidor porque desta forma ele funciona como um *choke point* de fato sem que para isso seja necessário um roteador (figura 11).

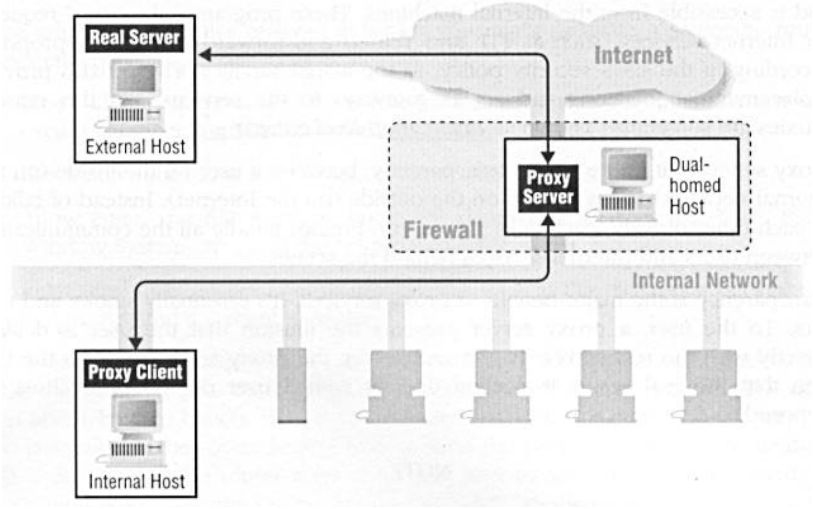


Figura 11 – Esquema de um *Dua homed host* atuando como *proxy server*.

Alguns serviços, denominados *store-and-forward*, tais como SMTP, NNTP e NTP, suportam *proxying* de uma forma natural. Estes serviços são projetados de tal forma que as mensagens (*e-mail*, *news*, *clock settings*) são recebidos por um servidor e então armazenados até que eles possam ser enviados adiante para um outro servidor apropriado. Portanto, cada *host* intermediário atua como uma espécie de procurador.

O *proxy server* atua como um procurador que aceita as chamadas que chegam e checa se é uma operação válida. Após receber a chamada e verificar que a solicitação é permitida, o servidor procurador envia adiante a solicitação para o servidor real. A procuração atua como servidor para receber a solicitação que chega e como um cliente, quando envia adiante a solicitação (figura 12). Depois que a sessão é estabelecida, a aplicação procuradora atua como uma retransmissora e copia os dados entre o cliente que iniciou a aplicação e o servidor. Devido ao fato de todos os dados entre o cliente e o servidor serem interceptados pelo application *proxy* ele tem controle total sobre a sessão e pode realizar um *logging* tão detalhado quanto se desejar.

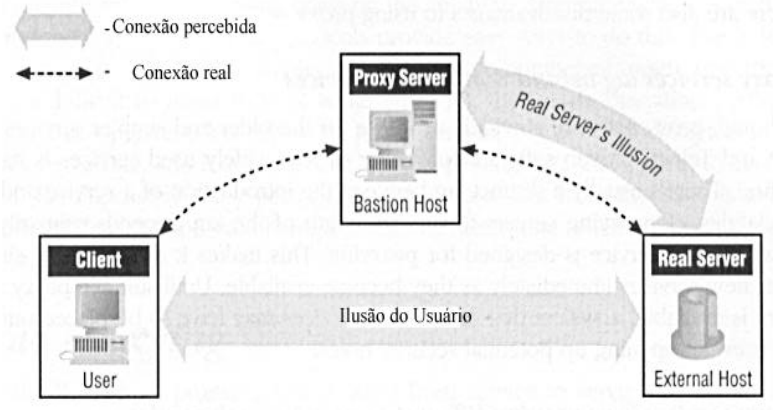


Figura 12 – Esquema do funcionamento genérico de um *Proxy Server*.

Para serviços que não apresentam características originais de *proxying*, os detalhes do funcionamento do procurador dependem de cada protocolo

(serviço) em questão. Em contrapartida, a comunicação do programa cliente com o servidor *proxy* pode ser realizada de três formas distintas:

- *Custom client software*: trata-se de um cliente modificado. Nessa situação o programa cliente deve saber como o servidor *proxy* opera, como contactá-lo e como passar as informações sobre o servidor real solicitado. Para o usuário tudo se passa de uma forma completamente transparente;
- *Custom user procedures*: nesse caso o usuário utiliza um cliente convencional, sem alterações, para contactar o servidor *proxy*. O processo ocorre da seguinte forma: o usuário contacta o servidor *proxy* da mesma forma como um servidor qualquer. Após, utilizando procedimentos diferentes (comandos do *proxy server*), ele fornece as informações acerca do servidor real a ser contactado; o servidor então realiza a conexão com o referido servidor e, feito isso, o usuário estará numa interface igual àquela que estaria caso estivesse acessado diretamente o servidor remoto. A desvantagem desta alternativa é a falta de transparência; entretanto, tem-se como vantagem a reutilização dos mesmos programas clientes.
- *Custom server software*: trata-se do redirecionamento do serviço para o *proxy server*. Nessa situação o programa cliente faz a solicitação ao servidor real, porém uma regra de redirecionamento altera a porta e *host* destino para os do *proxy server* e esse faz a conexão real. Para o usuário tudo se passa de uma forma completamente transparente.

Vantagens e Desvantagens.

Algumas vantagens de sistemas procuradores são:

- Permitem aos usuários acesso direto aos serviços na Internet: apesar de haver um procurador atuando em nome do cliente, esse mantém a ilusão de estar ser comunicando diretamente com o servidor remoto;
- Bons mecanismos de *log*: como todo o tráfego dos serviços procurados passa pelo servidor procurador, e tudo até o nível de aplicação, uma grande

quantidade de informações podem ser registradas de acordo com as necessidades de auditoria e segurança;

- Podem ser negados acessos a sites que contenham conteúdo indesejado para a organização.

Algumas desvantagens dos servidores procuradores são:

- Dependendo da situação pode ocorrer degradação no desempenho (tempo de resposta) no acesso ao um serviço;
- Pode ser necessário utilizar diferentes servidores procuradores para cada serviço;
- Alguns serviços não são viáveis para operar via procuradores;
- Um serviço por procuração não protege contra todas as fraquezas dos protocolos, depende da habilidade de se determinar que operações são seguras em um determinado protocolo.

3.2.4.2 SCREENED SUBNET

Composta por componentes mais básicos (*packet filters* e *bastion hosts*) esta é uma arquitetura que apresenta múltiplos níveis de redundância e provê um bom esquema de segurança, constituindo-se em um exemplo clássico de arquiteturas de *firewall*.

Os componentes deste tipo de *firewall* incluem os seguintes (figura 13):

1. Rede perimetral (zona desmilitarizada): citado anteriormente constitui-se numa sub-rede situada entre a rede interna e a rede externa (Internet);
2. Roteador externo: diretamente conectado à Internet e à rede perimetral;
3. Roteador interno: diretamente conectado à rede interna e à rede perimetral;
4. *Bastion hosts* residentes na rede perimetral (pode ser o próprio roteador).

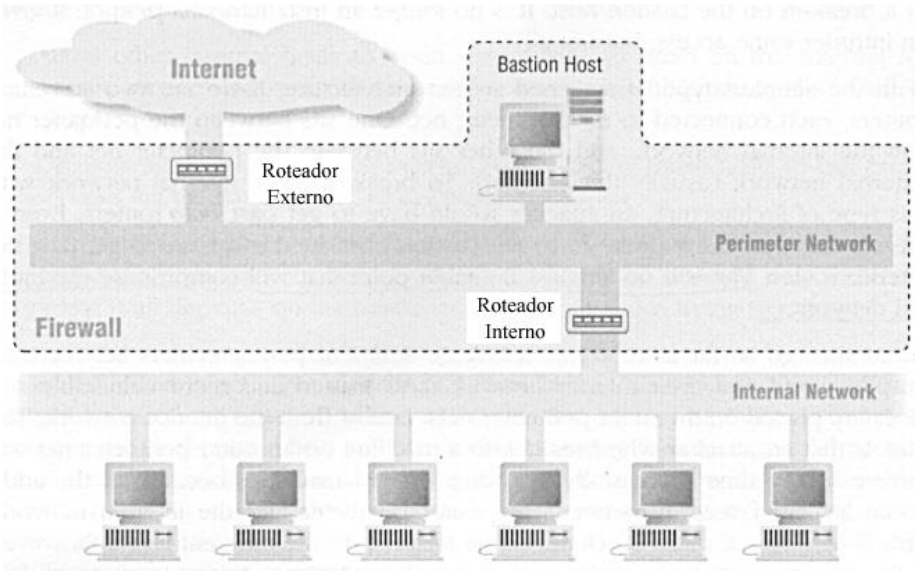


Figura 13 – Esquema de uma arquitetura Sreened Subnet.

Ao invés de se utilizar dois roteadores, o que pode ser muito caro dependendo dos recursos disponíveis, pode-se utilizar um único roteador com três interfaces de rede, mas que possibilite aplicar as regras de filtragem em cada interface em ambos os sentidos (figura 14). Nesse caso, segundo Chapman (1995), "a solução seria um pouco mais complexa porque teria de se executar um *merge* dos dois conjuntos de filtragem adotados no roteador externo e interno, agora substituídos por um único roteador".

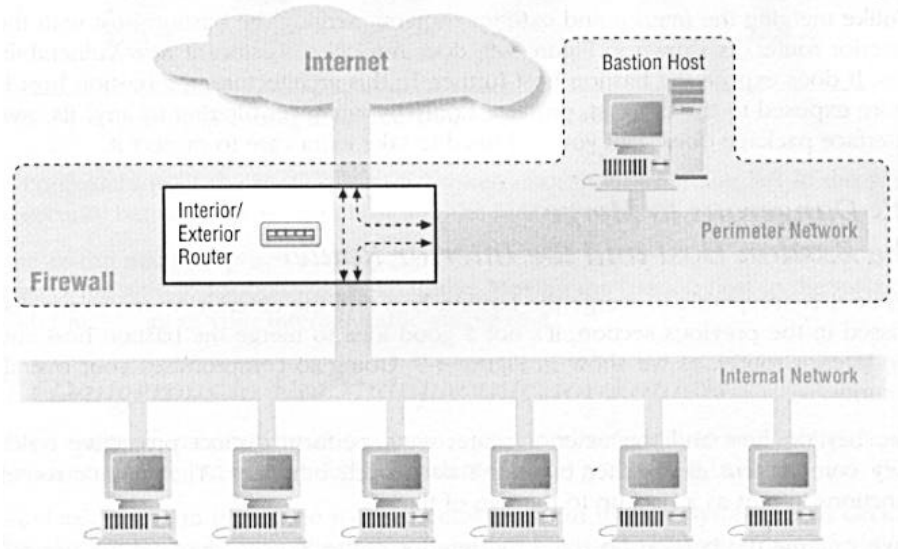


Figura 14 – Esquema da arquitetura *Screened Subnet* utilizando um único roteador.

Podem existir alguns servidores secundários localizados na rede interna, fazendo o serviço para com alguns servidores primários localizados nos *bastion hosts* (MAIL *servers*, NEWS *servers*, DNS *servers*, etc).

Definidos os serviços a serem providos, pode-se elaborar as regras de filtragem para os roteadores externo e interno. Vale ressaltar que um pouco de redundância sempre ajuda e, como alternativa, repete-se alguma das regras de filtragem adotada no roteador externo também no roteador interno, de forma que o interno ainda se configure como uma barreira caso o externo seja atacado com sucesso.

Com relação aos serviços disponíveis nos *bastion hosts*, há uma grande gama de alternativas de boas ferramentas (*proxy servers*) disponíveis na Internet entre os mais utilizados tem-se o Squid.

Análise frente as Estratégias de Segurança.

Para melhor definir quão seguro uma arquitetura de *firewall* pode ser, uma alternativa é verificando de que maneiras tal estrutura pode satisfazer as estratégias de segurança. eis alguns pontos básicos:

- *Least privilege*: há uma série de possibilidades, pois todos os serviços que são fornecidos exclusivamente via procuradores asseguram que os clientes internos terão essa única possibilidade de acessar um servidor externo na Internet;
- *Defense in depth*: os *hosts* internos estão protegidos tanto pelo roteador externo como pelo interno, assim como os *bastion hosts* estão protegidos tanto pelo roteador externo como também pela sua própria configuração;
- *Choke point*: a *perimeter network* é o *choke point* nessa arquitetura. Caso todos os serviços sejam fornecidos via procuração, os *bastion hosts* serão os *choke points* em particular;
- *Weakest link*: não há nenhuma *weakest link* óbvia nessa configuração. Tudo depende de quão bem configurados estão os procuradores; mesmo assim, caso o *bastion host* seja comprometido, o prosseguimento do ataque dependerá de como esse serviço pode servir como meio de propagação de um ataque para a rede interna;
- Fail safe: o princípio "Tudo que não é expressamente permitido é proibido" assegura que qualquer serviço novo que se queira prover só será permitido caso o *screening router* seja apropriadamente configurado;
- *Universal participation*: pode ser voluntária ou involuntária (caso o *firewall* seja o único *choke point*). Também depende de quão transparente o *firewall* pode ser, o que está relacionado com as características dos procuradores e das facilidades disponíveis nos programas clientes. De qualquer forma, a educação dos usuários é fundamental para que se tenha compreensão das medidas de segurança adotadas. Se não houver cooperação, o *firewall* pode

deixar de ser o único ponto de acesso à Internet bastando para isso que um usuário descontente acesse um provedor via linha discada.

- *Diversity of Defense*: esta estratégia encontra aplicação em diversos pontos, eis alguns exemplos: utilizando roteadores de diferentes marcas ou então, caso sejam utilizados computadores configurados como *screening routers*, adotando diferentes ferramentas de filtragem de pacotes nos dois roteadores.

3.2.4.3 SCREENED HOST

Nesta arquitetura não há uma sub-rede de segurança (*perimeter network*) entre a Internet e a rede Interna. Existem apenas um *screening router* e um *bastion host* situado junto à rede interna (figura 15).

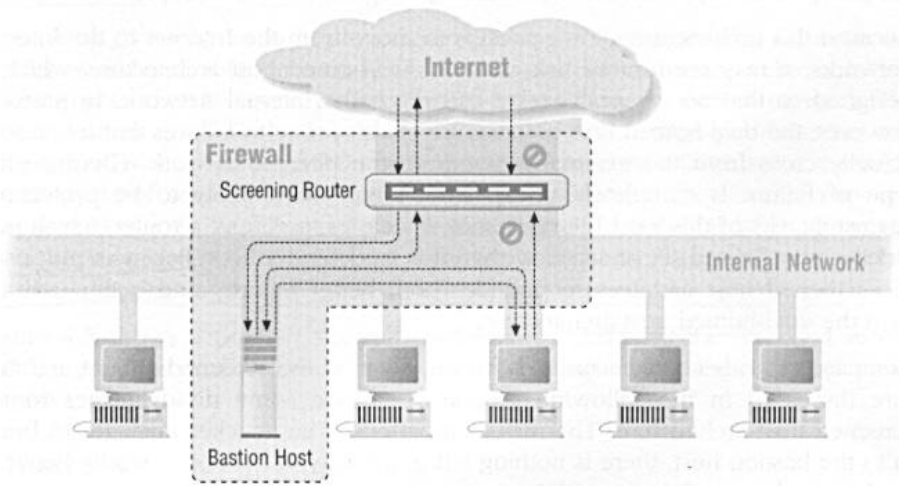


Figura 15 – Esquema da arquitetura *Screened Host*.

Análise frente as Estratégias de Segurança

Esta não é uma arquitetura muito segura, eis algumas observações sobre como ela satisfaz ou não as estratégias de segurança:

- *Least privilege*: pode ser observada quando os serviços são fornecidos exclusivamente via procuradores; entretanto, o fato do

bastion host estar situado na rede interna e, além disso, acumular privilégios de vários servidores pode ser um fator que vá contra o princípio do mínimo privilégio devido a sua posição crítica.

- *Defense in depth*: esta estratégia não é satisfeita principalmente porque basta que um dos componentes, roteador ou servidor, seja comprometido para que toda a rede interna esteja ao alcance do atacante.
- *Choke point*: o roteador é o *choke point* nesta arquitetura.
- *Weakest link*: o *bastion host* é o alvo mais visado pelos atacantes porque ele é o servidor de diversos serviços e está localizado junto à rede interna.
- *Fail safe*: esta não é uma arquitetura que permite falhas seguras de uma forma geral. Visto que basta comprometer o *bastion host* pra se ter acesso a rede interna. Um certo nível de falha segura é possível da mesma forma como foi exposto para a arquitetura *screened subnet* caso o *screening router* seja configurado segundo a filosofia "o que não é expressamente permitido é proibido".
- *Universal participation*: caso o roteador seja a única via de acesso à Internet, tem-se participação involuntária dos usuários da rede interna. Entretanto, valem as mesmas observações feitas para a arquitetura *screened subnet*.
- *Diversity of defense*: pouca ou nenhuma oportunidade de se aplicar esta estratégia, pois há um único roteador e *bastion host*.

3.3 FILTRAGEM DE PACOTES COM LINUX

Como citado anteriormente, um filtro de pacotes é um *software* que analisa o cabeçalho (*header*) dos pacotes enquanto eles passam, e decide o seu destino (Russel, 2003). Ele pode decidir entre:

- descartar (DROP) o pacote (como se nunca o tivesse recebido);
- aceitar (ACCEPT) o pacote (deixando-o seguir ao seu destino);
- gerar *logs* (LOG) ou

- outros alvos *(targets)* especificados pelo usuário.

No Linux, a filtragem de pacotes está implementada diretamente no *kerne*l. A principal utilização desse recurso é a realização da análise do cabeçalho dos pacotes e a eventual decisão sobre o seu destino.

3.3.1 A EVOLUÇÃO DOS MECANISMOS DE FILTRAGEM

Conforme relatado por Russel 2003, os *kernels* do Linux têm tido filtros de pacotes desde a série 1.1. A primeira geração, baseada no *ipfw* do BSD, foi portada por Alan Cox no final de 1994. Essa implementação foi melhorada por Jos Vos e outros voluntários para o *kernel* 2.0, com a ferramenta *ipfwadm*, que controlava as regras de filtragem do *kernel*.

Em meados de 1998, Rust Russel e Michael Neuling, voltaram a fazer alterações no *kernel* para implementar novos mecanismos de filtragem. Este esforço culminou no lançamento da ferramenta *ipchains* para o Linux *kernel* 2.2. Finalmente, a ferramenta da quarta geração, o *iptables*, foi lançada em meados de 1999 para o Linux *kernel* 2.4.

Além das funcionalidades dos seus antecessores, o *iptables* trouxe níveis sofisticados de filtragem, modularização de recursos, inspeção e controle dos estados dos pacotes (*statefull inspection*). A infraestrutura para funcionamento do *iptables* é implementada pelo Netfilter, um *framework* adicionado ao novo *kernel* do Linux (2.4) via módulo ou compilado (para suportar o módulo Netfilter, na compilação a opção CONFIG_NETFILTER deve ser habilitada.) diretamente nesse *kernel*. A ferramenta *iptables* comunica-se diretamente com o *kernel* (via Netfilter), controlando o destino de cada pacote analisado.

3.3.2 A FILTRAGEM TRADICIONAL

A filtragem de pacotes implementada desde o *kernel* 2.0 dispõe de três listas de regras na tabela *filter* (módulo específico do *kernel* para filtragem de

pacotes); tais listas são denominadas *firewall chains* (ou simplesmente *chains*). As três *chains* básicas são INPUT, OUTPUT e FORWARD, apresentadas graficamente na Figura 16.

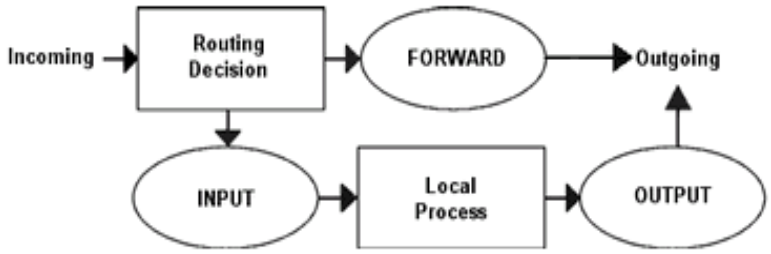


Figura 16 - Esquema de filtragem tradicional do *iptables*.

Os três "balões" em destaque representam as *chains* mencionadas acima. Quando o pacote atinge um balão no diagrama, a *chain* é examinada a fim de que seja decidido o destino do pacote. Se a *chain* aponta para descartar (DROP) o pacote, ele é descartado, mas se a *chain* aponta para aceitar o pacote (ACCEPT), ele segue como no diagrama.

Uma *chain* é uma lista de regras. Cada regra pode ser interpretada como uma condição do tipo: *"se o cabeçalho do pacote se parece com isso, aqui está o que deve ser feito com o pacote"*. Se a regra não se associa com o pacote, então a próxima regra na *chain* é consultada. Não havendo mais regras a consultar, o *kernel* analisa a política da *chain* para decidir o que fazer. Como apresentado anteriormente a *chain* pode ter 2 tipos de política: "Tudo o que não for expressamente proibido é permitido" ou "Tudo o que não for expressamente permitido é proibido".

Conforme ilustrado na figura 16, quando o pacote chega ao computador, pela placa *ethernet*, por exemplo, o *kernel* analisa o seu destino, num processo denominado roteamento *(routin*g). Segue-se então a decisão de roteamento, onde é realizado o repasse (FORWARD), ou, caso o pacote se destine à própria máquina, o encaminhamento à *chain* INPUT. Ao chegar nestas *chain*s, o pacote

passa por regras específicas que decidirão o seu destino (aceitar ou descartar).

3.3.3 NOVAS IMPLEMENTAÇÕES DO IPTABLES

Além dos recursos das soluções de filtragem de pacotes que o antecederam, o *iptables* (com o suporte do Netfilter) é extensível, isto é, dispõe de suporte a módulos adicionais, proporcionando várias outras funcionalidades. Estes módulos podem ser carregados "por demanda", de acordo com a necessidade de filtragem.

Estas extensões são classificadas em três tipos:

- Extensões de controle de protocolos (-p)
- Novos alvos (-j)
- Novas associações (-m)

3.3.3.1 EXTENSÕES DE CONTROLE DE PROTOCOLOS

Extensões TCP

As extensões TCP são automaticamente carregadas quando especificada a opção '-p tcp' e dispõe das seguintes opções:

--tcp-flags - Permite que sejam filtradas *flags* TCP específicas.

--source-port - Indica uma porta ou conjunto *(rang*e) de portas TCP de origem.

--destination-port- Indica uma porta ou conjunto *(rang*e) de portas TCP de destino.

--tcp-option - Utiliza-se de números específicos para controlar (e eventualmente descartar) pacotes com a opção TCP igual ao do número informado. Um pacote que não tem um cabeçalho TCP completo é automaticamente descartado se há uma tentativa de examinar suas opções TCP.

Extensões UDP

Essas extensões são automaticamente carregadas se a opção ‘-p udp' é especificada. Permite as opções **'--destination-port'** e **'--source-port'**.

Extensões ICMP

Essas extensões são automaticamente carregadas se a opção '-p icmp' é especificada.

Possui uma só opção diferente das demais extensões:

--icmp-type – Controla o tipo de conexão ICMP baseado no nome de tipo ou tipo numérico comumente utilizado em conexões deste tipo.

3.3.3.2 NOVOS ALVOS

Os alvos (*targets*) indicam “o que fazer” com o pacote caso coincida com as condições da regra. Os alvos padrões embutidos no *iptables* são: DROP (descartar) e ACCEPT (aceitar). Se a regra se associa com o pacote e seu alvo é um desses dois, nenhuma outra regra é consultada: o destino do pacote já foi decidido. Há dois tipos de alvos diferentes dos descritos acima: as *chains* definidas por usuários e as extensões.

Chains definidas por usuários.

Uma funcionalidade que o *iptables* herdou do *ipchains* é a possibilidade da criação de novas *chain*s, além das três disponíveis (INPUT, FORWARD e OUTPUT).

Quando um pacote associa-se com uma regra cujo alvo é uma *chain* definida pelo usuário, o pacote passa a ser analisado pelas regras dessa nova *chai*n.

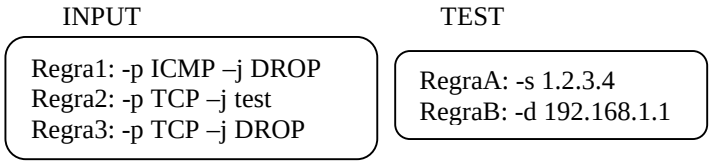


Figura 17 - Exemplo de *chains* definidas pelo usuário.

A figura 17 mostra graficamente duas *chain*s: INPUT (a *chain* padrão) e test (uma *chain* definida pelo usuário). No modelo da figura 17, considerando um pacote TCP vindo de 192.168.1.1, com destino a 1.2.3.4, este pacote submete-se à *chain* INPUT e é testado pela Regra1, com a qual não se associa.

Porém, quando submetida à Regra2 ocorre a associação, tendo como alvo test. Logo, a próxima regra examinada será a primeira da *chain* test. A RegraA na *chain* test não se associa ao pacote por não coincidir com a origem, passando o pacote à próxima regra (RegraB). Como também não ocorre a associação, já que apresenta um destino diferente, e o pacote chega ao final da *chain* sem se associar a nenhuma regra, a análise retorna à *chain* INPUT. Já que a última regra examinada desta *chain* foi a Regra2, a regra a ser examinada agora é a Regra3, que também não se associa com o pacote.

Graficamente, o caminho percorrido pelo pacote é como ilustrado na figura 18.

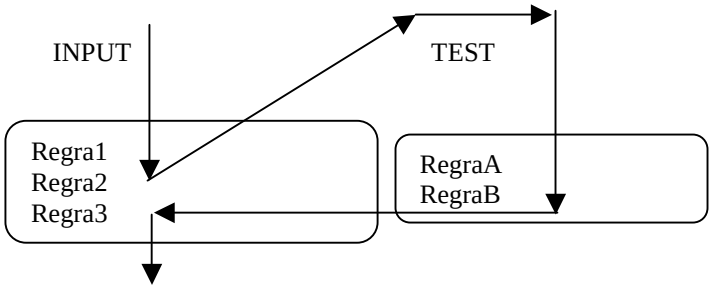


Figura 18 – Diagrama do caminho percorrido pelo pacote na nova *chain.*

Extensões ao iptables: Novos alvos (targets)

O outro tipo de alvo é a extensão. Uma extensão-alvo consiste em um módulo do *kernel*, opcional ao *iptables*, para prover opções de linha de comando. As extensões na distribuição padrão do netfilter são:

LOG - Esse módulo provê o registro em *logs* dos pacotes submetidos, possuindo as seguintes opções adicionais:

--log-level - Seguido de um número de nível ou nome.

--log-prefix - Seguido de uma *string* de até 29 caracteres, que será adicionada no início da linha de registro de *log (syslo*g), permitindo melhor identificação da mesma.

REJECT - Esse módulo tem o mesmo efeito do alvo 'DROP', porém, retorna uma mensagem de erro ao remetente, rejeitando o pacote. O REJECT era um alvo nativo no *ipchains*. Porém, pela semelhança com o DROP, (além da pouca utilização), foi disponibilizado no *iptables* como uma extensão opcional.

3.3.3.3 NOVAS ASSOCIAÇÕES

Estes tipos de extensões, no pacote netfilter, podem ser habilitadas com a opção ‘-m’.

Estas funcionalidades adicionais permitem um controle mais detalhado dos pacotes, não se detendo ao cabeçalho, analisando informações em seu conteúdo.

mac - Utilizado para associar a regra com o endereço *Ethernet* (MAC) da máquina de origem do pacote (--mac-source).

limit - Utilizado para restringir a taxa de pacotes, e para suprimir mensagens de *lo*g.

owner - Associa a regra a várias características do criador do pacote gerado localmente.

state - Interpreta a análise do controle da conexão feita pelo módulo *ip_conntrack.*

3.4 HONEYNET E HONEYPOT

Além do *Packet Filter* e *Bastion Host*, uso de mecanismos para observação das atividades de invasores em redes conectadas à Internet vem sendo utilizado na prática há um bom tempo no mundo da tecnologia da informação.

Um pouco da história desses mecanismos está descrita no artigo "Honeynet.BR: desenvolvimento e implantação de um sistema para avaliação de atividades hostis na Internet brasileira", escrito por especialistas do NIC BR *Security Office* (NBSO) e Instituto Nacional de Pesquisas Espaciais (INPE).

As primeiras experiências na área datam de 1988, quando o especialista Clifford Stoll faz um relato completo sobre a história da invasão (origem do ataque, motivos e redes-alvo) nos sistemas do Lawrence Berkeley Laboratory (LBL).

Quatro anos depois, em 1992, seria a vez do especialista Bill Cheswick explicar no artigo "*An Evening with Berferd In Which a Cracker is Lured, Endured, and Studied*" os resultados do acompanhamento de invasões em um dos sistemas da AT&T, projetado especialmente para este fim (Rocha, 2003).

O termo *honeypot* só surgiria em meados de 1998, quando Fred Cohen desenvolveu a ferramenta *Detection Toolkit* (DTK), a primeira utilizada para "emulação de diversas vulnerabilidades e coleta de informações sobre os ataques sofridos".

Mas é em 1999, quando um grupo de especialistas em segurança da informação liderado por Lance Spitzner lança o *Honeynet Project*, uma rede projetada exclusivamente para ser comprometida por ataques, que o conceito de *honeynets* ganha repercussão mundial e demonstra a importância do estudo do

comportamento dos invasores de uma rede para o desenvolvimento de novas ferramentas e sistemas de defesa (Rocha, 2003).

Segundo Cristine Hoepers, Analista de Segurança Sênior do NIC BR *Security* Office (NBSO), "*Honeypot é um recurso de segurança preparado especificamente para ser sondado, atacado ou comprometido e para registrar essas atividades. Já Honeynet é uma rede projetada especificamente para ser comprometida e utilizada para observar os invasores. Essa rede normalmente é composta por sistemas reais e necessita de mecanismos de contenção eficientes e transparentes, para que não seja usada como origem de ataques e também não alertar o invasor do fato de estar em uma honeynet*".

3.4.1 HONEYNETS NO BRASIL

O projeto Honeynet.BR (http://www.honeynet.org.br/), foi criado e é mantido em parceria por especialistas do INPE (http://www.lac.inpe.br/) e do grupo brasileiro de resposta a incidentes de segurança NBSO (http://www.nbso.nic.br/).

Em março de 2002 começaram as operações do Honeynet.BR. Em junho de 2002 o Projeto Honeynet.BR tornou-se membro da *Honeynet Research Alliance*, que reúne diversos grupos de várias partes do mundo, todos empenhados em desenvolver a tecnologia de *honeynets*.

Depois do pioneirismo do Honeynet.BR, a expectativa é que, em pouco tempo, o número de projetos nessa área ganhe novos adeptos no Brasil. Por exemplo, o Laboratório de Redes de Alta Velocidade (RAVEL), ligado à COPPE e contando com o apoio da FAPERJ, colocou em operação um projeto envolvendo uma *honeynet* (Rocha,2003).

3.4.2 TIPOS DE HONEYPOTS

Segundo Cristine Hoepers, o *Honeynet Project* utiliza comumente dois

tipos principais de *Honeypot* (Rocha, 2003):

Honeypots de baixa interação (*Low-interaction Honeypots*): normalmente apenas emulam serviços e sistemas operacionais, não permitindo que o atacante interaja com o sistema.

Honeypots de alta interação (*High-interaction Honeypots*): são compostos por sistemas operacionais e serviços reais e permitem que o atacante interaja com o sistema.

O projeto brasileiro de *honeynet* utiliza *honeypots* de alta interação.

Os *Honeypots* também são considerados como Sistemas de Detecção de Intruso (IDS).

3.5 SISTEMAS DE DETECÇÃO DE INTRUSO

O IDS compreende uma série de medidas que envolvem: monitorar a rede ou sistema para prevenir-se de ataques que venham a comprometer a segurança, restringir acesso a área de *stack* na memória evitando assim ataques do tipo buffer *overflow*, construção de uma política de cuidados específicos com os *logs* do sistema, periodicamente rodar ferramentas que verifiquem a integridade dos arquivos, criação de "aquários" ou "*Honeypots*" e uso de porta em escuta. Tudo isso com o objetivo de se antecipar, saber de antemão quando a rede foi alvo de interesse de um possível invasor e assim tomar as medidas necessárias para a total segurança do seu sistema. (Bejtlich, 2003).

A tabela 3 apresentada os principais eventos do sistema que podem ser monitorados por um sistema IDS.

CONEXÕES	LOGIN	FILE SYSTEM	ROOT
Tráfego TCP/UDP	Atividade não usual de *log in* e *log out* do sistemas	Checagem da integridade dos arquivos periodicamente	Alteração do *Kernel* para logar atividades específicas
Tentativas frustadas de conexão			
Port-Scanners			

Tabela 3 - Principais eventos do sistema que pode ser monitorados em um IDS.

Uma política de *Firewall* exemplar é fundamental e necessária, porém há inúmeras formas de ataque e até mesmo o *firewall* não é capaz de conter todas elas. Geralmente implementado entre o *firewall* e o sistema a ser protegido um IDS fornece uma camada extra de proteção dando informações preciosas ao administrador.

3.5.1 TIPOS DE IDS

3.5.1.1 NETWORK BASED SYSTEMS - NIDS

Operam monitorando o tráfego do segmento de rede inteiro. O NIC (*Network Interface Card*) – a placa de rede; Pode operar de dois modos:

Modo normal: Os pacotes com destino MAC ADDRESS específico da NIC de um computador são capturados e processados por este.

Modo Promíscuo: Todos os pacotes que chegam a NIC, independente do MAC ADDRESS, são capturados e processados.

No UNIX, o administrador pode definir na linha de comando o modo de operação (normal ou promíscuo) da sua *eth0* (Interface de rede). A maioria dos sistemas IDS baseados em rede operam em modo promíscuo.

Sistemas de Detecção de Intrusos em rede podem também realizar tarefas como:

Monitorar a rede por *port scans* comuns como TCP connect() ou SYN scans (“*stealth*”). Antes de realizar qualquer ação mais evasiva é comum o levantamento de informação com *port scanners* que podem verificar as portas abertas no *host* ou *Scanners* de Vulnerabilidades como Nessus, por exemplo, que já fornece as fraquezas do sistema para quem realiza o *scanning*. Dependendo do risco da sondagem detectada pelo IDS, o administrador poderá tomar as medidas necessárias.

Monitorar conexões válidas que tentem por em ação ataques conhecidos. Acessar o servidor WEB na porta (80) pode parecer uma atividade inofensiva, mas algumas tentativas de ataque tipo CGI-attacks podem ser fatais em alguns casos. Por exemplo, um ataque como este:

telnet vitima.com.br 80 <ENTER>

GET /../../../etc/passwd HTTP/1.0 <ENTER>

Deve ser devidamente bloqueado e encarado como de alto risco.

Saber detectar atividade incomum é essencial. Grandes redes acumulam grande volume de tráfego, desse modo para que o administrador possa saber o que realmente merece sua atenção existe uma série de programas que analisam alguns parâmetros do sistema procurando por anormalidades. Entre os mais utilizados tem-se o Snort. Já o Portsentry é uma excelente ferramenta para se ter rodando já que evidencia tentativas de *stealth scans* ocorridas e automaticamente jogam o IP do atacante no *hosts.deny* do sistema operacional UNIX.

3.5.1.2 INTEGRIDADE DOS ARQUIVOS

A importância de se manter um banco de dados com as informações de seus arquivos é muito importante. *Hackers* costumam “*trojanar*” ou modificar certos arquivos (*rootkit*) vitais no sistema. Para o administrador saber que algo mudou é altamente recomendado que se tenha ferramentas como o Tripware

(Bejtlich, 2003), que fazem o database dos principais arquivos do sistema guardando as informações para posteriores checagens, se estiver tudo certo, o programa envia uma mensagem que o sistema esta intacto, caso contrario diz quais arquivos forma alterados. Não se deve esquecer de fazer um *backup* do database dos arquivos.

3.6 COMENTÁRIOS FINAIS

A partir revisão de literatura pode-se concluir que existem diversas técnicas, ferramentas e políticas que podem ser adotadas para implementação da segurança em redes. Também foi observado que o conhecimento e maturidade em segurança, do pessoal envolvido, é fundamental para o sucesso das políticas e recursos aplicados visando a segurança da informação.

Considerando as informações apresentadas, no capítulo estudo de caso serão apresentadas as políticas, técnicas e ferramentas adotadas para implementação da segurança em redes. Podendo servir como proposta para implementação de segurança redes com características semelhantes.

4 ESTUDO DE CASO

A apresentação do estudo de caso foi baseada no capítulo 3 “Medidas de Segurança” apresentado anteriormente.

No estudo de caso foi utilizada a rede do Centro Federal de Educação Tecnológica de Campos. Atualmente esta rede possui cerca de 600 clientes conectados a Internet, *link* com banda de 2Mbps, diversos servidores conforme exemplificados na tabela 4.

Servidores (serviços)	**Sistema operacional**	
	Linux	**Windows**
E-mail	3	0
WEB	3	2
FTP	4	1
DNS	3	1
Boot remoto	1	0
Terminal *server*	1	1
Proxy WWW	2	0
Banco de dados	1	2
Impressão	3	2
SSH	10	
Arquivos	2	1
Domínio	1	1

Tabela 4 – Quantidade de servidores por sistema operacional da rede do CEFET Campos.

A situação anterior à implementação da segurança em redes, entre outros, era:

- Único roteador (CISCO 2511) respondendo por quatro redes lógicas classe C;
- Única rede física com aproximadamente 400 *hosts*;
- Inexistência regra de filtragem no roteador;

- Inexistência de *Firewall*;
- Todos os *hosts* (clientes e servidores) com IP Público;
- Inexistência de políticas;
- Inexistência de qualquer tipo de monitoração (*logs*, gráficos, etc);
- *Backup* deficiente e com política inadequada;
- Servidores desatualizados;
- Banda de 256 Kbps de acesso a Internet;
- Inexistência de *proxy servers*.

A partir das informações apresentadas, conclui-se que a rede do CEFET Campos era um "prato feito" para atacantes internos, invasores externos, vírus, entre outros.

4.1 POLÍTICA DE SEGURANÇA E POLÍTICA DE USO

Toda instituição/empresa deve possuir documento que descreva a política de segurança e política de uso. Este último deve ser de conhecimento de todos que fazem uso da rede interna. Em geral, tal política é um documento a ser assinado pelo usuário em questão (Uchoa, 2003).

O CEFET Campos, por ser uma autarquia federal, está sujeito a decretos, leis e medidas provisórias de esfera federal. Considerando estas informações e a revisão de literatura, a política de segurança foi elaborada com base em:

- Decreto nº 3.505, de 13/06/2000, que institui a Política de Segurança da Informação nos órgãos e entidades da Administração Pública Federal;
- Portaria nº 316, de 12/07/2000, do CEFET CAMPOS, que estabelece normas de conteúdo disponível na rede WEB do CEFET CAMPOS;
- Medida Provisória nº 2.200-2, de 24/08/2001, que institui a Infra-Estrutura de Chaves Públicas Brasileira – ICP-Brasil, transforma o Instituto Nacional de Tecnologia da Informação em autarquia, e dá outras providências;
- Recomendação nº 01, de 09/12/2002, da Secretaria de Logística e

Tecnologia da Informação do Ministério do Planejamento, Orçamento e Gestão.

No anexo A tem-se uma cópia da portaria número 267 de 20 de outubro de 2003 do Cefet Campos, publicada no diário oficial, a qual foi resultado da elaboração da política de segurança. Esse procedimento visou tornar pública e dar validade legal a política de segurança. Além da publicação, foi feita divulgação por correio eletrônico para os usuários da rede interna. Cada instituição/empresa deve buscar a forma mais adequada de divulgar e validar legalmente a sua política de segurança.

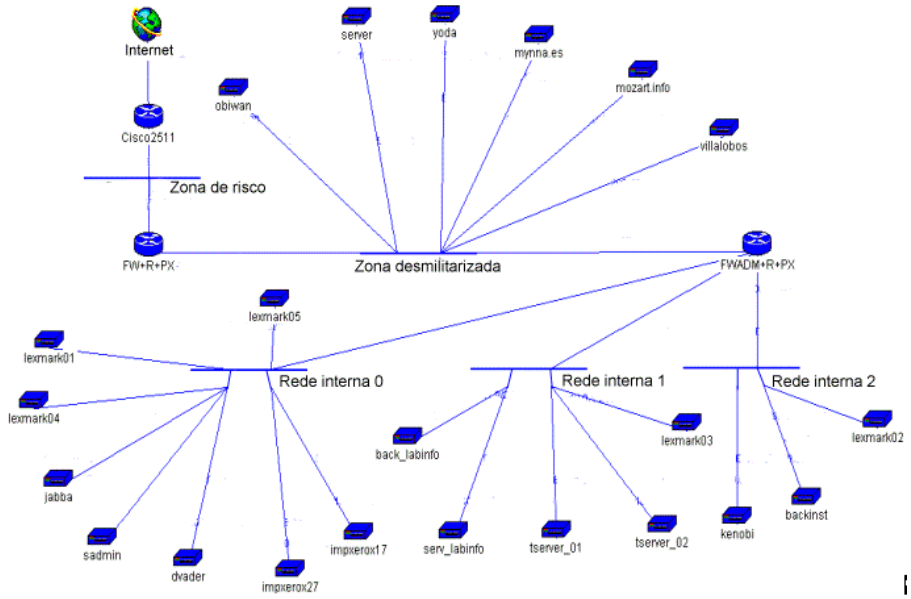
A partir da leitura do anexo A, observa-se que a política de segurança elaborada definiu: acessos restritos e liberados; itens preservados; níveis de acessos dos usuários e regras da política de uso.

4.2 ARQUITETURA DE REDE SEGURA

A arquitetura de rede adotada foi uma modificação da *Screened Subnet* com 5 redes lógicas e físicas distintas: rede externa (zona de risco), rede perimetral (desmilitarizada - DMZ) e redes internas (administrativa, educacional oficinas e informática). Considerou-se que a instituição possui, basicamente, tráfegos de rede com perfis distintos que podem ser classificados como: administrativo e educacional. Entretanto devido ao número de máquinas existentes na rede educacional, optou-se por dividir a rede educacional em duas: oficinas e informática. Logo as redes internas foram classificadas como administrativa (0), educacional informática (1) e educacional oficina (2). Estas três redes utilizam endereços IPs privados 10.0.X.0, onde X representa o número da rede. Embora a rede privada 10.0.0.0 seja classe A, optou-se por utilizar máscara de rede classe C para evitar que máquinas de uma determinada rede fossem alocadas para outra sem o conhecimento do pessoal responsável.

Na figura 19, pode-se observar um esquema da arquitetura de rede

adotada com a respectiva localização dos servidores.



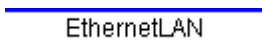
Legenda	
Símbolo	**Descrição**
	Internet
	Rede local - ethernet
	Servidor
	*Screening Router* (Roteador e/ou *proxy* e/ou filtro de pacotes e/ou IDS)

Figura 19 – Esquema da arquitetura de rede adotada e localização dos servidores.

Entre dois *screening router*, existe a rede perimetral (desmilitarizada) com IPs públicos, na qual foram instalados os servidores de Internet. Nas redes internas, atrás do *screening router* interno foram instalados os servidores de Intranet e Extranet.

O *screening router* interno faz o roteamento entre as redes internas e a perimetral, este é composto de um PC-AT Atlhon 2.0 GHz com 512MB de

RAM DDR *Front Size Bus* (FSB) 266 MHz, 18 GB de disco rígido SCSI II, placa principal ASUS, 4 placas de rede 3Com e sistema operacional Linux da distribuição Astaro. Esta máquina faz filtragem de pacotes, mascaramento (SNAT), *proxy* transparente WEB e FTP, IDS com detecção de *port scan* e DNAT (*Destination Network Andress Translator*) para os servidores internos (Extranet).

O *screening router* externo faz o roteamento entre a rede perimetral e a rede externa, este é constituído de um PC-AT Atlhon 2.0 GHz com 512 MB de RAM DDR FSB Bus 266 MHz, 18 GB de disco rígido SCSI II, 2 placas de rede 3Com e sistema operacional Linux da Distribuição RedHat. Esta máquina faz filtragem de pacotes, *proxy* transparente WEB, lista de controle de acesso WEB, IDS com detecção de *port scan* e IDS com *faker server* (Portsentry). O roteamento com o link de Internet foi feito com roteador Cisco série 2500 modelo 2511.

Pode ser observado que, entre os dois *screening router*, com filtragem de pacotes, tem-se a rede perimetral ou rede desmilitarizada (DMZ). Analisando a configuração, pode-se concluir que a estratégia de segurança adotada permite:

Least privilege: os serviços de acesso WEB e FTP são fornecidos exclusivamente por procuradores (*proxy*) com controle de acesso. *Proxy* seguro, *screening router* interno permite apenas conexões a partir da rede internas e *screening router* externo permite apenas conexões a partir da rede desmilitarizada. Filtragem de pacotes permitindo os serviços considerados básicos (FTP, SMTP, HTTP, HTTPS, DNS, POP, IMAP) e necessários, todos os demais foram negados;

Defense in depth: os *hosts* internos estão protegidos tanto pelo *screening router* externo como pelo interno, assim como os servidores localizados na rede desmilitarizada estão protegidos tanto pelo *screening router* externo como também pela sua própria configuração com IDS e filtragem de

pacotes local;

Choke point: a rede perimetral é o *choke point* nessa arquitetura. Os serviços WEB, FTP são fornecidos via *proxy* e filtragem de pacotes, nesse caso os *screening routers* são os *choke points* em particular. Além disto os servidores de e-mail possuem verificação de vírus no conteúdo do anexo e bloqueio de anexos executáveis;

Weakest link: não foi observada nenhuma *weakest link* óbvia nessa configuração;

Fail safe: o princípio "Tudo que não é expressamente permitido é proibido" foi assegurado visto que, qualquer serviço novo que se queira prover só será permitido caso o *screening router* externo e interno sejam apropriadamente configurados;

Universal participation: é involuntária, pois todo o tráfego é obrigado a passar pelo *firewall* e pode ser voluntária, dependendo da educação dos usuários. Infelizmente foi observado que alguns usuários concordam, mas não aceitam as políticas de restrições adotadas;

Diversity of Defense: esta estratégia é atendida a medida que: os *screening router* são de distribuições distintas, existe detecção de *port scan* interno e externo, existe IDS com *faker server* nos servidores internos e *screening router* externo, existe filtragem de pacotes internos e externos, pacotes ICMP externos são negados. Máquinas internas são protegidas de vírus em dois níveis, nos servidores de e-mail e localmente através de ferramenta para instalação e atualização remota de antivírus.

4.2.1 REGRAS DE FILTRAGEM DE PACOTES

O *screening router* interno da distribuição Astaro versão 2.016, específica para segurança em redes, pode ser obtida imagem isso em http://www.astaro.com. Possui interface WEB (HTTPS) que facilitou a

monitoração e configuração da regras de filtragem de pacotes, detecção de *port scan*, uso de CPU, RAM, disco, etc. A figura 20 apresenta a imagem da interface utilizada para configuração das regras de filtragem de pacotes utilizando a distribuição Linux Astaro.

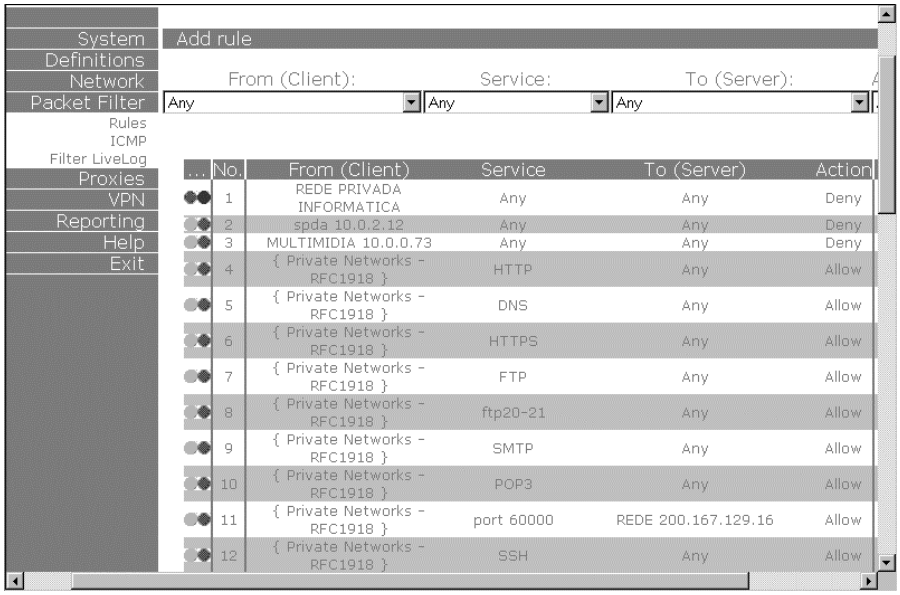


Figura 20 - Imagem da interface de configuração das regras de filtragem do *screening router* interno.

Esta interface é um *frontend* via HTTPS que utiliza o *iptables* para configuração da regras de filtragem de pacotes.

Na configuração da filtragem de pacotes do *screening router* externo (distribuição RedHat), optou-se por utilizar o Firewall Builder versão 1.0.6 para criar as regras de *iptables* versão 1.2.7. O Firewall Builder promove uma interface GUI (figura 21) para criação de regras, após é gravado um arquivo .xml que pode ser “compilado” criando um *script iptables* entre outros.

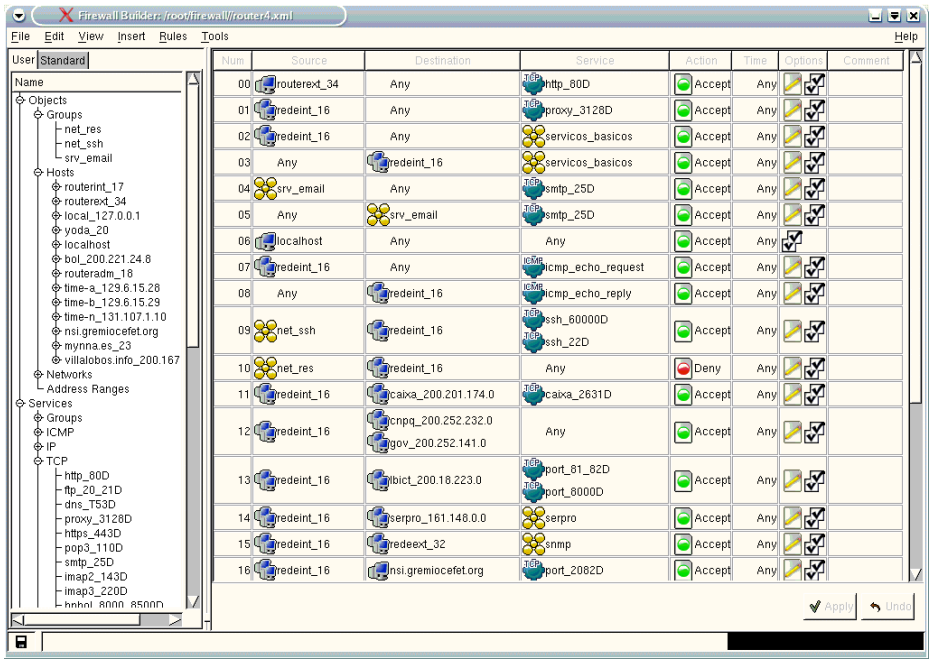


Figura 21 - Interface para configuração da regras de filtragem de pacotes do *screening router* externo.

Na figura 21 tem-se do lado esquerda definição de redes, *hosts*, serviços, grupos de redes, grupos de serviços, entre outros. No lado direito os campos das configurações incluindo as regras de filtragem, conforme apresentadas e comentadas na tabela 5. O uso da ferramenta é relativamente simples, pois utiliza o conceito arraste e solte.

Regra	Cometário.
00	Permite o *screening router* externo acessar o serviço HTTP dos servidores WEB da Internet. Regra devido ao DNAT do *proxy* transparente.
01	Permite a rede perimetral acessar a porta do *proxy* (Squid) no *screening router* externo.
02	Permite acesso da rede perimetral aos serviços considerados básicos de Internet (HTTPS, HTTP, DNS, FTP, POP e IMAP).
03	Permite o acesso da Internet aos serviços básicos (HTTPS, HTTP,, DNS, FTP, POP e IMAP) da rede perimetral.
04	Permite apenas os servidores internos de e-mail fazerem uso do serviço SMTP. Regra para evitar SPAM de origem interna.
05	Permite que apenas os servidores internos de e-mail receber conexão no serviço SMTP.
06	Libera *localhost*
07	Permite ICMP tipo ECHO REQUEST para Internet.
08	Permite ICMP tipo ECHO REPLY oriundo da Internet.
09	Permite redes específicas acessarem serviço ssh na rede perimetral.
10	Nega redes reservadas o acesso a rede perimetral e gera *log*. Esta medida é para proteção de ataques de IP *spoof*.
11	Permite acesso a serviço específico da Caixa Econômica Federal.
12	Permite acesso a qualquer serviço da rede do CNPQ.
13	Permite acesso a serviços específicos para funcionamento da biblioteca.
14	Permite acesso a serviços específicos da rede do SERPRO.
15	Permite consulta SNMP os roteador Cisco para geração de relatórios de uso de banda.
16	Porta para uso do grêmio acadêmico.
17	Nega e gera *logs* de detecção de *port scan*.
18	Nega e gera logs de IP fragmentados.
19	Permite que redes específicas façam acesso ao serviço ssh do *screening router* externo.
20	Permite ICMP tipo ECHO REPLY oriundo da Internet para *screening router* externo.
21	Permite ICMP tipo ECHO REQUEST para Internet com origem *screening router* externo.
22	Permite que máquina interna faça montagem de compartilhamento de arquivos via nfs para geração de relatório de acessos utilizando o Sarg.
23	Regra para registrar todos os pacotes descartados.

Tabela 5 - Comentários das regras de filtragem de pacotes do *screening router* externo.

Entre outras funcionalidades, o Firewall Builder permitiu a criação de

regras específicas para as interfaces. No anexo D podem ser analisadas todas as regras de filtragem de pacotes, do *screening router* externo, criadas para o *iptables* com auxílio do Firewall Builder. Também pode ser observado, a partir da figura 21 (lado direito) e anexo, que todos os pacotes são registrados por *logs*.

4.3 ELEVAÇÃO DO NÍVEL DE SEGURANÇA DOS HOSTS

4.3.1 ANTIVÍRUS NOS SERVIDORES DE E-MAIL

Para reduzir a possibilidade de infecção das estações por vírus, *trojans* ou *worns* instalou-se antivírus nos servidores de e-mail.

O servidor de e-mail utilizado foi o Sendmail, o procedimento constou basicamente de:

1 – Instalação do antivírus;

2 – Instalação do *script* Amavis que pode ser obtido em http://www.amavis.org;

3 – Re-configuração do Sendmail;

4 – Teste do conjunto utilizando assinatura de vírus.

O Amavis não é um antivírus e sim um *script* que, depois de instalado, utilizada um antivírus para verificações nos anexos dos e-mails *incoming* e *outcoming* a presença de vírus.

Após instalação do *Script* Amavis e do antivívus adotado, o Uvscan da *Networks Associates* para Linux versão 4.16.0, foram feitas alterações no arquivo /etc/mail/sendmail.cf conforme apresentadas a seguir.

Linhas originais comentadas.

#Mlocal, P=/usr/bin/procmail, F=lsDFMAw5:/|@qSPfhn9, S=10/30, R=20/40,

T=DNS/RFC822/X-Unix,

A=procmail -Y -a $h -d $u

Linhas após alterações.

Mlocal, P=/usr/sbin/scanmails, F=lsDFMAw5:/|@qSPfhn9, S=10/30, R=20/40,

T=DNS/RFC822/X-Unix,

A=scanmails -Y -a $h -d $u

Após todas as configurações, foram feitos testes de funcionalidade que constaram basicamente de envio de e-mails (*incoming* e *outcoming*) com anexo contendo assinatura de vírus do **eicar**, organização que combina universidades, indústria, pessoas físicas, etc. Alguns dos objetivos são: unir esforços contra a proliferação de código malicioso, contra crime e fraude de computador. Esta assinatura que pode ser obtida em http://www.eicar.com. Sendo detectado vírus no anexo, o e-mail foi bloqueado e, automaticamente, enviada uma notificação, por e-mail, para o remetente e outra para o destinatário.

Devido a grande quantidade de novos vírus que surgem diariamente, foi necessário desenvolver um *script* para manter as assinaturas de vírus do antivírus atualizadas. Diariamente o *script* de atualização do antivírus, conforme cópia no anexo B, foi executado por agendamento no Linux.

A avaliação da funcionalidade do antivírus no servidor de e-mails foi feita utilizando um *script* (anexo C) desenvolvido para criar e enviar, por e-mail, um relatório conforme exemplificado no anexo E. A partir dos dados do relatório foi construído o gráfico da figura 22 e concluiu-se que 1 de janeiro e 25 de dezembro de 2003 foram encontrados, no servidor yoda.cefetcampos.br com 548 contas, um total de 5685 e-mails contendo 20 diferentes tipos de vírus.

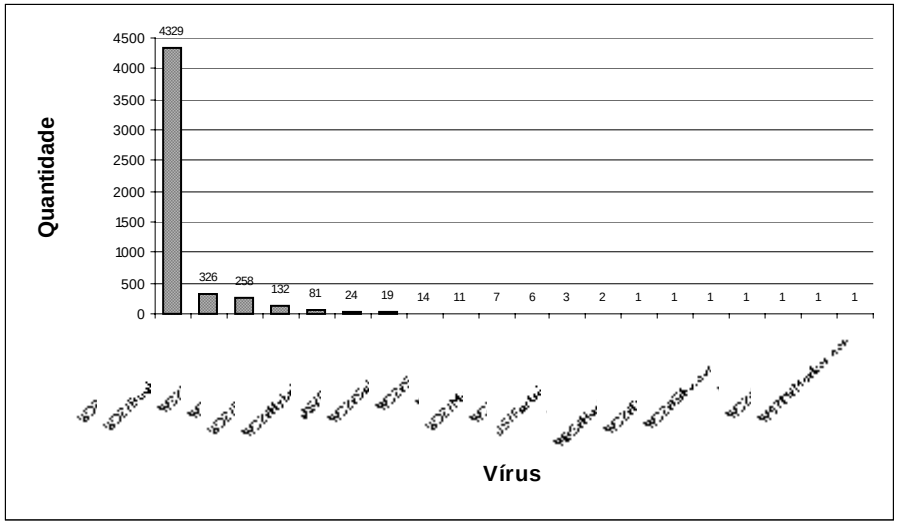


Figura 22 - Gráfico da quantidade de vírus por nome encontrados nos e-mail do CEFET Campos no ano 2003.

O gráfico da figura 22 apresenta a quantidade de vírus por nome durante esse período. Não foram totalizados e-mails com anexos executáveis com extensões exe, com, vbs, bat, pif, js, shs, scr, chm, dll, hta, bas, lnk, isn, ade, adp, cmd, cpl, crt, hlp, inf, ins, isp, jse, mdb, mde, msc, msi, msp, mst, pcd, reg, sct, url, vb, vbe, wsc, wsf, wsh e htt. Observa-se que 4329 vírus encontrados foram W32/Sobig.f@MM, destes 2326 foram detectados no dia 2 de agosto de 2003. Esta quantidade foi completamente atípica, considerando que a grande maioria das estações clientes possui sistema operacional Windows, pode-se concluir que os danos poderiam ser enormes para o funcionamento destas máquinas e redes.

4.3.2 BLOQUEIO DE E-MAILS COM ANEXOS EXECUTÁVEIS

Além do antivírus nos servidores de e-mail, os correios eletrônicos contendo anexos executáveis com extensões exe, com, vbs, bat, pif, js, shs, scr, chm, dll, hta, bas, lnk, isn, ade, adp, cmd, cpl, crt, hlp, inf, ins, isp, jse, mdb, mde, msc, msi, msp, mst, pcd, reg, sct, url, vb, vbe, wsc, wsf, wsh e htt foram

bloqueados. Esta medida visou proteger os usuários de arquivos com código malicioso (cavalo de tróia, vírus, trojans, entre outros) e prevenir a presença de e-mail com novos vírus na janela de infecção (entre atualizações).

A solução adotada foi desenvolvida por Mauro Borchardt e Carlos Maziero "Filtro de anexos executáveis de e-mail" (PPGIA/PUCPR http://www.ppgia.pucpr.br). Utilizou-se a versão 1.2 distribuída de acordo com a licença GNU.

A solução consistiu em duas partes: uma regra de Procmail (no arquivo /etc/procmailrc), para detectar anexos com certas extensões, e um *script* Perl (/etc/procmail.d/recusanexo.pl). Se um arquivo em anexo contendo determinada extensão fosse detectado, o e-mail era repassado ao *script* Perl . Nesse *script*, o remetente e o destinatário do e-mail foram notificados do ocorrido, o e-mail era descartado e gerada uma entrada no arquivo de log (/var/log/procmail.log).

4.3.3 ANTIVÍRUS NOS CLIENTES DA REDE

Os clientes da rede do Cefet Campos utilizam o antivírus McaFee licenciado. Visando atualização das assinaturas de vírus, automaticamente uma vez por dia é feito o download das novas verões para um servidor de FTP (obiwan.cefetcampos.br) em seguida é gerado um e-mail de notificação para todos os usuários. O anexo B apresenta o *script* que realiza esta tarefa.

Além disto, para tornar a atualização mais rápida e funcional, optou-se por utilizar a ferramenta EPo para Windows (suíte AVD Mcafee) da *Network Associate*. Esta ferramenta possibilitou remotamente instalação, atualização, agendamento de atualização e geração de relatórios dos antivírus. A figura 23 apresenta uma das telas do servidor EPo para configuração dos antivírus nos clientes, já a figura 24 apresenta um exemplo de relatório, gráfico das 10 máquinas mais infectadas.

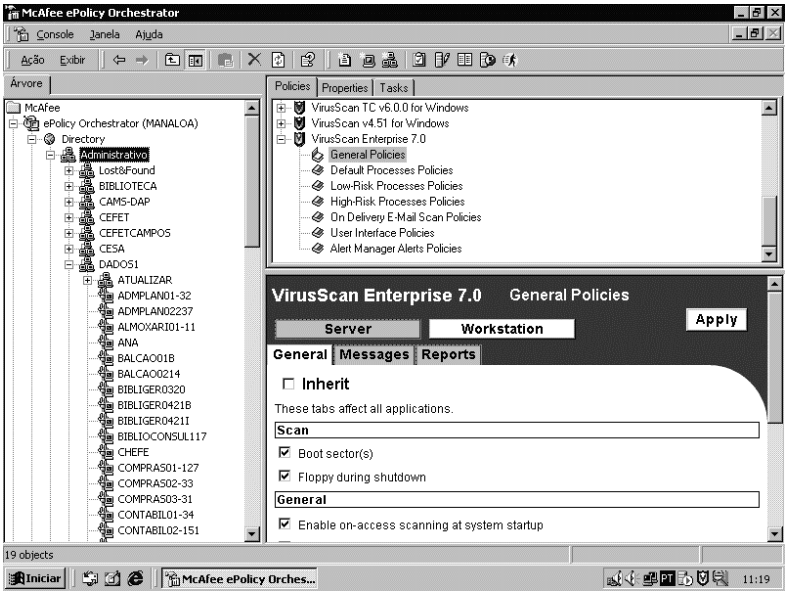


Figura 23 - Tela de configuração dos antivírus nos clientes da rede gerenciados pelo Epo.

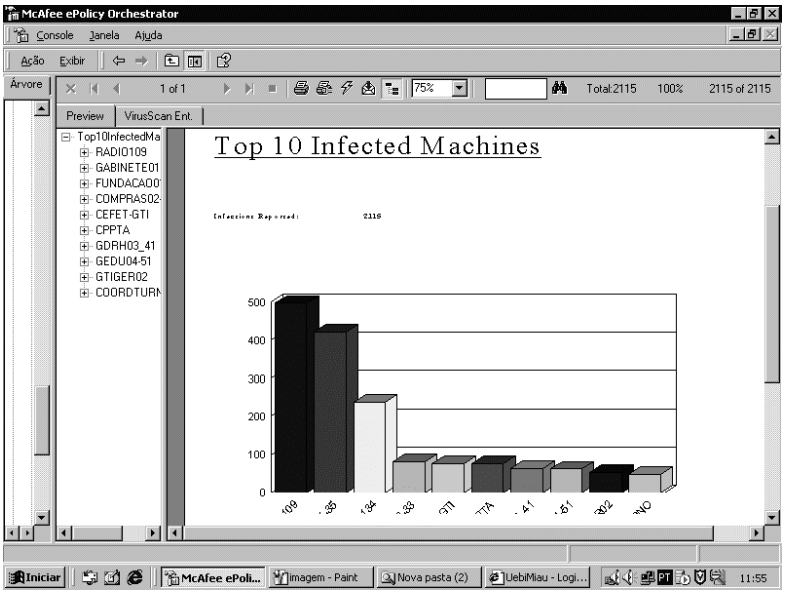


Figura 24 - Tela com um dos possíveis relatórios gerados pelo Epo.

4.3.4 DESATIVAÇÃO DE SERVIÇOS DESNECESSÁRIOS

Após a instalação dos servidores, por padrão alguns serviços são ativados, os considerados desnecessários foram desativados. O procedimento constou de:

1 - Identificação das portas em escuta. Comandos *netstat –nat* e *netstat –nau* para portas TCP e UDP respectivamente. Na dúvida foi utilizado o comando *fuser –v port/tcp* (onde port é o número da porta) para verificar qual o processo é responsável pela porta em escuta.

2 - Desativação do serviço. Utilizando o comando: service serviço stop.

3 - Desativação da inicialização automática. Ferramenta *ntsysv* ou comando *chkconfig* ou removendo o link do nível de inicialização desejado.

4.3.5 DEFINIÇÕES DE SENHAS

Todas as senhas para administração foram elaboradas com pelo menos 8 caracteres, compostos de letras números e símbolos, estas são trocadas freqüentemente e, as senhas dos usuários utilizados para administração remota são distintas.

Foi feita verificação a procura de senhas simples dos usuários. Foram encontradas senhas do tipo: nome, sobrenome, números de documentos, placas de carros, números de telefones, data de nascimento e palavras constantes em dicionários. Após foram feitas solicitações para que os usuários com estas senhas simples fizessem as substituições.

4.3.6 ATUALIZAÇÕES DOS SERVIDORES

A tarefa de atualização do servidor não é simples, pelo contrário, é trabalhosa. Entretanto algumas distribuições já utilizam a tecnologia apt desenvolvida pela equipe Debian. Por questões de agilidade e escassez de mão

de obra na instituição, optou-se por utilizar este recurso.

A distribuição utilizada nos servidores foi a Conectiva Linux 8 e 9 que já possui este recurso, e a RedHat com instalação posterior do recurso apt. Para configuração e atualização basta:

1 - Configurar os repositórios no arquivo /etc/apt/source.list;

2 - Digitar apt-get update para atualizar lista de arquivos do repositório;

3 - Digitar apt-get upgrade –y para fazer download e instalar as atualizações.

Os itens 2 e 3 são realizados pelo menos uma vez por semana.

É importante observar que o recurso apt não possibilita atualização do *kernel*, para isto, deve-se fazer download da nova versão e compilar na máquina.

4.3.7 BACKUP

Para o *backup* foram utilizadas duas unidades de gravação de fita dat de 24 GB e uma de 4 GB.

A política de *backup* consta de uma fita para cada dia da semana durante uma semana, uma fita para cada semana durante um mês e uma fita para cada mês durante um ano. As fitas mensais são guardadas em local acondicionado, de acesso restrito e com segurança física.

No caso dos servidores Linux, utilizou-se *scripts* agendados como exemplificado a seguir. No final da quinta linha pode ser observado que a cópia dos *logs* também foi feita. Isto é muito importante, pois mediante a algum incidente os dados podem estar preservados.

```
#!/bin/sh
case $1 in
   b)
      echo "Iniciando backup"
      tar --preserve-permission --atime-preserve -cvf /dev/st0 /var/named
/var/spool /etc /var/www /home /root /var/log
   ;;
   r)
      echo -e "\a Informe a pasta para a restauracao dos arquivos
```

```
(ex.:/tmp) \c"
            sleep 1
            read PASTA
            PASTAATUAL=$(pwd)
            cd /$PASTA
            tar -xvpf /dev/st0
            cd /$PASTAATUAL
          ;;
          *)
          echo Parametros validos b = backup e r = restore
          ;;
        esac
```

4.3.8 LOGS

4.3.8.1 LEITURA DE LOGS

A leitura de *logs* é uma tarefa extremamente trabalhosa e devido a falta de pessoal, optou-se por utilizar a ferramenta Logwatch versão 1.88 (ftp://ftp.logwatch.org/pub/) para leitura diária dos logs do sistema. Esta ferramenta pode gerar relatórios dos *logs* em três níveis de detalhamento e enviar por e-mail. Optou-se por utilizar o menor nível de detalhes devido a quantidade de informações e servidores. A seguir, tem-se um exemplo simples de um dos relatórios gerados diariamente em um dos servidores.

```
################## LogWatch 4.3.2 (02/18/03) ####################
Processing Initiated: Mon Dec 8 00:00:01 2003
Date Range Processed: yesterday
Detail Level of Output: 0
Logfiles for Host: mynna.es.cefetcampos.br
################################################################
--------------------- ipop3d Begin ------------------------
**Unmatched Entries**
connect from 200.167.129.23: 1 Time(s)
--------------------- ipop3d End ------------------------
-------------------- Connections (secure-log) Begin ----------------------
Connections:
```

Service ipop3d:
: 1 Time(s)
---------- Connections (secure-log) End ----------
---------- SSHD Begin ----------
Users logging in through sshd:
manut logged in from 200.167.209.68 using password: 1 Time(s)
---------- SSHD End ----------
---------- Disk Space ----------
Filesystem Size Used Avail Use% Mounted on
/dev/sda2 486M 200M 261M 44% /
/dev/sda1 23M 3.3M 18M 15% /boot
/dev/sda3 1.9G 361M 1.4G 20% /usr
/dev/sdb1 9.6G 497M 8.6G 6% /var
/dev/sdb2 3.8G 49M 3.6G 2% /tmp
/dev/sdb3 980M 37M 894M 4% /home

A utilização do Logwatch facilitou a tarefa de leitura de *logs*, pois os relatórios são resumidos e organizados e seções de acordo com o serviço. Sendo identificada uma operação ou ação anômala, os *logs* originais são analisados.

A identificação de conexões aos consoles foi feita utilizando os comandos *last* e *lastlog*.

4.3.9 CONFIGURAÇÃO DE FILTRAGEM DE PACOTES NOS SERVIDORES

Além da filtragem de pacotes feitas pelos *screening router* interno e externo, nos servidores foi configurada filtragem de pacotes utilizando o *iptables*. Basicamente a filtragem constou de:

1 - Liberação de ICMP ECHO REPLY para rede local;

2 - Liberação de ICMP ECHO REQUEST para qualquer;

3 - Liberação dos serviços ativos;

4 - Negação dos demais serviços e protocolos.

4.3.10 PARTICIONAMENTO DOS SERVIDORES

Conforme a finalidade do servidor, foram criadas partições considerando que um usuário ou um programa mal-comportado pode lotar uma partição na

qual tenha permissões de escrita (áreas temporárias e de armazenamento de *logs* são suscetíveis a este problema). Se os programas do sistema estiverem em outra partição eles provavelmente não serão afetados, evitando que o sistema trave. Caso uma partição seja corrompida por alguma razão, as outras partições provavelmente não serão afetadas. Em alguns sistemas (notadamente sistemas Unix), é possível definir algumas características individuais para cada partição. Por exemplo, algumas partições podem ser usadas em modo *read-only*, o que é útil para partições que contenham binários que são modificados com pouca freqüência.

Em alguns casos a existência de várias partições permite múltiplas operações de disco em paralelo e/ou o uso de otimizações individuais para cada partição, o que pode aumentar significativamente o desempenho do sistema. O uso de várias partições geralmente facilita o procedimento de *backup* do sistema, pois simplifica funções como:

- copiar partições inteiras de uma só vez;
- excluir partições individuais do procedimento;
- fazer *backups* em intervalos diferentes para cada partição.

Um exemplo do esquema de particionamento pode ser observado no final do relatório do item 4.3.8.1.

4.3.11 ADMINISTRAÇÃO REMOTA

O serviço SSH foi configurado para permitir a administração remota, entretanto através do arquivo /etc/ssh/sshd_config a porta padrão foi alterado de 22 para uma porta alta. Esta medida visa evitar que esse serviços seja explorado pelo atacante, além disto, normalmente o *port scan* é feito em portas baixas. Além disto o acesso remoto como root foi bloqueado, desta forma é necessário logar com usuário de administração e fazer um relogin (su) para a conta de root.

No screening router externo foi feita filtragem de pacotes permitindo

acesso ao serviço ssh a partir de determinadas redes externas normalmente utilizadas para administração remota. Esta medida visou reduzir a quantidade de redes que poderiam ser utilizadas pelo atacante para explorar o serviço ssh nos servidores do CEFET Campos.

4.3.12 CONTROLE DE ACESSO A *PROXIES* WEB

Um *proxy* mal configurado pode ser usado pelo atacante como um "trampolim" para acessar recursos de forma anônima. Esta anonimidade pode ser usada para cometer crimes, tais como envio de mensagens caluniosas, difamatórias ou ameaçadoras e divulgação de pornografia envolvendo crianças.

A configuração correta para um *proxy* WEB é aquela que libera o acesso somente aos endereços IP de usuários autorizados (pertencentes à sua rede).

O *proxy* WEB utilizado foi o Squid. A configuração do acesso foi implementada no arquivo /etc/squid/squid.conf incluindo as seguintes linhas na sessão ACL:

```
...
acl redelocal src ip_rede/mascara_rede
...
http_access allow redelocal
http_access deny all
```

4.3.13 CONTROLE DE ACESSO A SITES "INDESEJADOS"

Considerado a política de segurança e uso da instituição (Anexo A), a negação dos acessos aos sites "indesejados" foi feita através do /etc/squid/squid.conf, as linhas da configuração são listada a seguir. Os arquivos txt contêm listas de palavras da url.

```
...
```

```
acl godurl url_regex “/etc/squid/goodurl.txt”
acl badurl url_regex “/etc/squid/badurl.txt”
...
http_access allow goodurl
http_access deny badurl
```

Pela ordem do http_access a permissão tem prioridade sobre a negação, logo uma url com palavra “temp3” em goodurl terá prioridade de permissão mesmo que exista a palavra “mp3” no arquivo badurl.txt.

4.3.14 IMPLEMENTAÇÃO DE IDS NOS SERVIDORES

Além das portas que respondem aos serviços normais, utilizando o sistema de detecção de intruso (IDS) Portsentry versão 1.0.4, foram configuradas mais 19 portas TCP e 18 UDP em quatro diferentes servidores: yoda.cefetcampos.br, obiwan.cefetcampos.br, mynna.cefetcampos.br e router.cefetcampos.br (*screening router* externo). Estas portas funcionaram como isca. Durante a tentativa de conexão ou *port scan*, o *host* atacante era bloqueado com regras de *firewall* e *tcpwrappers*, e gerado e-mail de notificação.

Para fins de testes, entre o dia 25/11/2003 a 25/12/2003, foram removidas as regras de filtragem de pacotes (*screening router* externo) para se observar o comportamento do sistema utilizando o Portsentry. O gráfico da figura 25 apresenta o número de conexões e/ou *port scan* em três servidores.

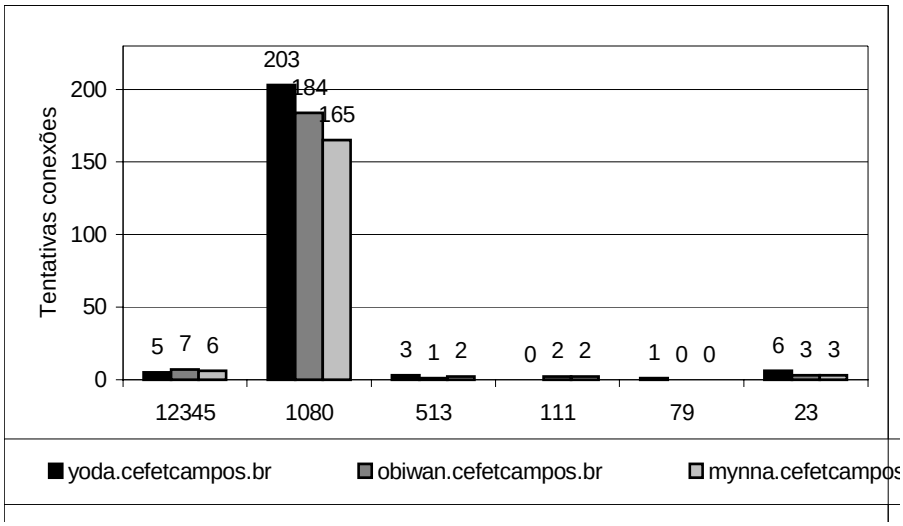


Figura 25 - Gráfico comparativo dos pedidos de conexões e/ou *port scan* por porta de três servidores do domínio cefetcampos.br

Foram registradas conexões em 6 das 47 portas configuradas nos três servidores. As portas exploradas foram: 513 *login*, *remote login*; 12345 NetBus 1.x (*avoiding Netbuster*); 1080 Wingate (*Socks-Proxy*); 111 *Sun Rpc*; 79 *Finger server* e 23 *Telnet.*

O servidor yoda.cefetcampos.br (DNS primário, e-mail e site da instituição) recebeu o maior número pedido de conexões, principalmente na porta 1080 (Wingate - *Socks-Proxy*), a seguir o obiwan.cefetcampos.br (FTP anônimo, DNS secundário) e mynna.cefetcampos.br (e-mail curso superior). Pelo número das portas, pode-se concluir que as conexões não são aleatórias e pela quantidade, conclui-se que o principal servidor do domínio necessita de maior atenção com a segurança.

4.3.15 REMOÇÃO DE SHELL

Para aumentar o nível de segurança, em todos os usuários do sistema,

exceto administradores e o usuário de administração, os *Shell* foram substituídos de /bin/bash para /bin/false no arquivo /etc/passwd.

4.3.16 SEGURANÇA FÍSICA DOS SERVIDORES

Todas os principais *hosts* e equipamentos localizados na Gerência da Tecnologia da Informação (GTI) são mantidos trancados e somente o pessoal autorizado tem acesso físico as máquinas. Além disto, estes equipamentos são alimentados por sistema de fornecimento de energia (*no-break*) mantido por 36 baterias de 12 Vcc, sendo 18 automotivas de 40 A/h e 18 de 16 A/h. A autonomia total é dependente da carga, utilizando *software* de monitoração e com monitores desligados, foi estimada uma autonomia de 10 horas.

4.4 MONITORAÇÃO CONTÍNUA DO TRÁFEGO DA REDE E DOS SERVIÇOS

Para monitoração do tráfego de rede e dos serviços, alguns procedimentos foram adotados:

1 - Geração de gráfico de uso de CPU do servidores e *screening router* do *firewall*.

Utilizou-se gráficos MRTG do *screening router* interno utilizando Astaro. O MRTG é ferramenta parar monitorar o tráfego de *links* de rede, mas pode ser utilizada para monitorar outros recursos como: uso de memória ram, uso de disco, uso de CPU, entre outros. A figura 26 apresenta um exemplo de um gráfico de utilização de CPU gerado com o MRTG da distribuição Linux Astaro.

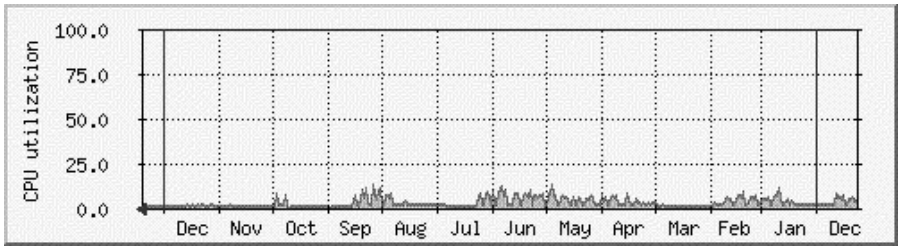
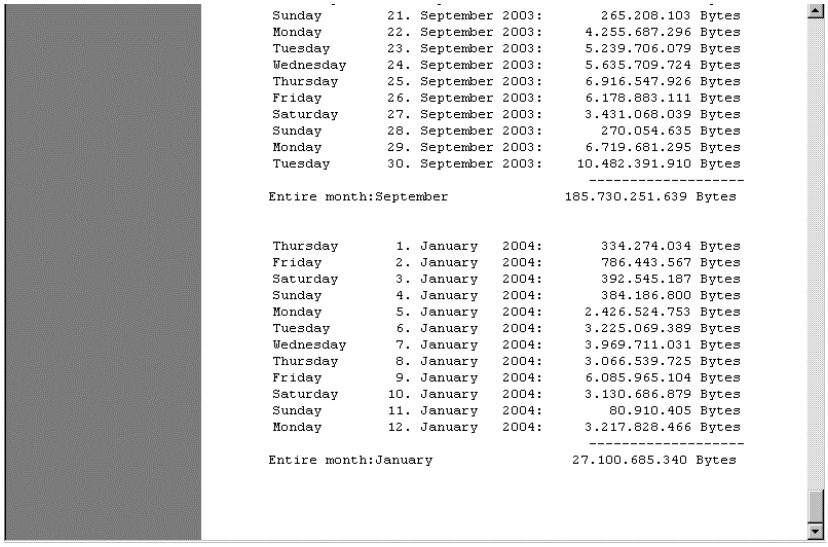


Figura 26 - Gráfico de utilização de CPU do *screening router* interno.

Utilizando o kdebase-ksysguard versão 3.1.2, software com interface GUI, foi possível monitorar nível de utilização de CPU (carga do usuário e carga do sistema), taxa de utilização da interface de rede, taxa de utilização das partições, etc.

2 - Total de bytes *incoming* e *outcoming* de Internet;

Recurso disponível no *screening router* interno utilizando Linux Astaro. A figura 27 apresenta o relatório do total de bytes por dia transferidos pelos *screening router*.



Sunday	21. September 2003:	265.208.103	Bytes
Monday	22. September 2003:	4.255.687.296	Bytes
Tuesday	23. September 2003:	5.239.706.079	Bytes
Wednesday	24. September 2003:	5.635.709.724	Bytes
Thursday	25. September 2003:	6.916.547.926	Bytes
Friday	26. September 2003:	6.178.883.111	Bytes
Saturday	27. September 2003:	3.431.068.039	Bytes
Sunday	28. September 2003:	270.054.635	Bytes
Monday	29. September 2003:	6.719.681.295	Bytes
Tuesday	30. September 2003:	10.482.391.910	Bytes
Entire month:September		185.730.251.639	Bytes
Thursday	1. January 2004:	334.274.034	Bytes
Friday	2. January 2004:	786.443.567	Bytes
Saturday	3. January 2004:	392.545.187	Bytes
Sunday	4. January 2004:	384.186.800	Bytes
Monday	5. January 2004:	2.426.524.753	Bytes
Tuesday	6. January 2004:	3.225.069.389	Bytes
Wednesday	7. January 2004:	3.969.711.031	Bytes
Thursday	8. January 2004:	3.066.539.725	Bytes
Friday	9. January 2004:	6.085.965.104	Bytes
Saturday	10. January 2004:	3.130.686.879	Bytes
Sunday	11. January 2004:	80.910.405	Bytes
Monday	12. January 2004:	3.217.828.466	Bytes
Entire month:January		27.100.685.340	Bytes

Figura 27 – Relatório de bytes por dia *incoming* e *outcoming* transferidos pelo

screening router.

3 - Gráfico de utilização de memória ram do servidor *proxy*;

Recurso disponível no *screening router* interno utilizando Linux Astaro.

4 – Gráfico de acertos do *proxy*;

Recurso disponível no *screening router* interno utilizando Linux Astaro.

5 - Gráficos de utilização de todas as portas do principal *switch* da instituição;

Utilizando MRTG e net-snmp versão 5.0.7, derivado da implementação do protocolo simples de gerenciamento de redes versão 2 (SNMPv2) da Universidade *Carnegie Mellon*. Pode ser visualizado *on-line* utilizando a url http://www.cefetcampos.br/mrtg/index2.html.

Estes gráficos possibilitaram visualizar o uso de banda de rede local para cada uma das redes e servidores com IP público.

6 - Gráficos de utilização de banda de acesso a Internet;

Gráfico montado via MRTG e net-snmp (conforme descrito anteriormente). A figura 28 apresenta um exemplo dos gráficos de uso de banda de acesso a Internet. Pode ser visualizado on-line via url http://www.cefetcampos.br/mrtg.

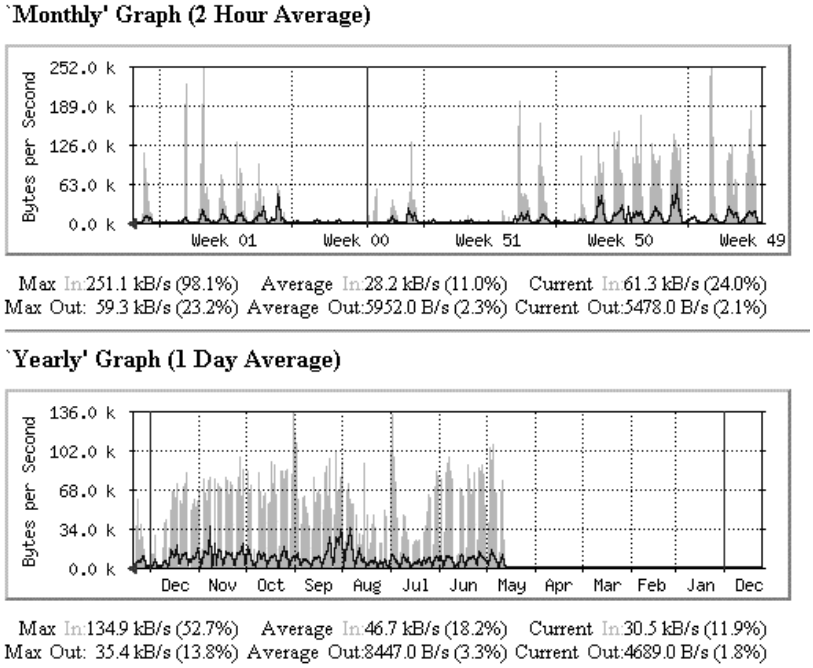


Figura 28 – Gráfico de uso de banda de acesso a Internet pelas redes do CEFET Campos.

7 - Relatório de sites acessados;

Utilizou-se a ferramenta Sarg versão 1.2.2.1. O Sarg (antigo Sqmgrlog) gera relatórios por usuário/ip/nome baseando-se no arquivo de log do SQUID (access.log). Os relatórios são gerados em HTML ou por e-mail. Os relatórios podem ser visualizados através da url http://www.cefetcampos.br/logs.

Estes relatórios possibilitaram efetuar um controle mais eficiente dos sites acessados, pois ao se identificar um site "indesejado", de acordo com a política de uso, a palavra chave era inserida no arquivo badurl.txt para controle do acesso feito pelo *proxy* WEB. Além disto, o relatório possibilitou identificar quais *hosts* da Internet tentaram fazer conexão não autorizada no *proxy* WEB.

8 - Acessos feitos ao principal site do Cefet Campos;

Foi utilizada a ferramenta Webalizer versão 2.01.10. Esta ferramenta é analisador de arquivos de *log* de servidores WWW. Os acessos podem ser visualizados através do relatório na url http://www.cefetcampos.br/acessos. Através deste relatório foi possível identificar ataques de IP *spoofing*.

4.5 DEFINIÇÃO DE TESTES PERIÓDICOS À PROCURA DE VULNERABILIDADES

Os testes periódicos para detecção de vulnerabilidades foram realizados com o Nessus-server e Nessus-lciente versão 1.2.0. O Nessus é um Scanner de vulnerabilidades de segurança gratuito, atualizável e completo para a plataforma Linux. É multitarefa, baseado em *plugins* e possui uma interface agradável em GTK. É capaz de gerar bons relatórios (HTML, LaTeX, texto ASCII) e não somente aponta as vulnerabilidades, como também sugere como corrigí-las. A atualização dos *plugins* de vulnerabilidades foi feita utilizando o comando *nessus-update-plugins.*

O medida que novas vulnerabilidades eram encontradas (por configuração inadequada ou erro de código), os aplicativos servidores eram reconfigurados ou atualizados para novas versões.

5 CONCLUSÕES

Considerando a revisão de literatura e desenvolvimento da proposta utilizada no estudo de caso, pode-se concluir que:

1- Existem muitas técnicas para se implementar ataques a segurança de redes e conseqüente informação. O nível de conhecimento, necessário para realizar estes ataques, é cada vez menor em função das facilidades das ferramentas desenvolvidas;

2 - A organização deve possuir uma ou mais pessoas responsáveis pela segurança da informação. O número de pessoas envolvidas dependerá basicamente do nível de dependência dos sistemas de informação e dos recursos disponíveis;

3 - Deve haver uma conscientização por parte dos usuários, pois sem eles não é possível se implementar segurança;

4 - A política de segurança e uso deve ser elaborada em conformidade com as características e particularidades da organização. Estas não devem ser dinâmicas e alteradas sempre que necessário. Os chefes e gerentes devem estar de acordo e respaldar o cumprimentos destas políticas;

5 - A arquitetura de rede deve ser elaborada visando segurança, desempenho e funcionalidade. Na medida do possível, deve-se adotar uma arquitetura que atenda as estratégias: *least privilege*, *defense in depth*, *choke point*, *weakest link*, *fail safe*, *universal participation* e *diversity of defense*;

6 - A arquitetura adotada deve ser documentada com informações úteis e relevantes para a manutenção;

7 - As regras de filtragem de pacotes devem ser implementadas considerando a estratégia do menor privilégio. Pode-se implementar regras que reduzam a

possibilidade de ataques tipo IP *spoofing* e conseqüente DOS utilizando esta técnica. A ferramenta Firewall Builder mostrou-se amigável e de simples utilização para construção das regras de filtragem de pacotes;

8 - As regras de filtragem de pacotes devem atender as necessidade da organização, não existem regras genéricas, para cada caso deve ser feita uma avaliação dos serviços necessários;

9 - Pode-se utilizar várias técnicas para aumentar o nível de segurança dos *hosts* da rede. Entres estes estão: utilização de antivírus nos servidores de e-mail, configuração IDS no servidores (atendendo *diversity of defense* e *defense in depth*), política e mídia de *backup* confiáveis, utilização e atualização dos antivírus nos clientes da rede, particionamento adequado com os finalidade dos servidores, senhas fortes, atualização dos *hosts*, leitura e análise de *logs*, controle de acesso a Internet e ao servidor *proxy*, alteração da porta padrão utilizada nos serviços de administração remota;

10 – A utilização de antivírus nos servidores de e-mail mostrou-se ser uma boa estratégia para conter esta praga virtual. Além disto o servidor de antivírus (EPo) apresentou ótimas funcionalidades, tendo reduzido significativamente a quantidade de vírus na rede interna e facilitado o trabalho de atualização das estações;

11 – O IDS Portsentry mostrou-se eficiente na detecção e negação de conexões indesejáveis. A partir dos registros das conexões feitas pode-se concluir que na Internet existem muitas solicitações de conexão não autorizadas ou inesperadas;

12 - Devem ser implementadas ferramentas para fazer a monitoração contínua dos recursos, estas devem ser utilizadas de forma eficiente e lógica visando resolver os problemas;

13 - Os testes periódicos devem ser realizados visando identificar vulnerabilidades não conhecidas pelo administrador;

Este documento não visa exaurir os assunto e sim apresentar uma

proposta, através de estudo de caso, do caminho que pode ser seguido para implementar a segurança da informação;

5.1 SUGESTÕES PARA TRABALHOS FUTUROS

1 - Estudar e descrever técnicas e ferramentas para investigação *forencci* de incidentes em redes;

2 - Estudar, comparar e avaliar ferramentas para identificação de e-mail SPAM;

3 - Estudar técnicas, implementar, testar e avaliar de *honeypot* e *honeynet*;

4 – Estudar, comparar e implementar de NIDS como por exemplo o Snot;

5 – Estudas e implementar a estratégia mais adequada para um *loghost* centralizado e sincronização de horários dos *logs*. Um *LogHost* centralizado é um sistema dedicado à coleta e ao armazenamento de *logs* de outros sistemas em uma rede, servindo como um repositório redundante de *log*s.

6 REFERÊNCIAS BIBLIOGRÁFICAS

"DILDOG". The Tao of Windows Buffer Overflow. [on-line]. Disponível na Internet via www url:http://www.cultdeadcow.com/cDc_files/cDc-351/. Arquivo capturado em abril de 1998.

"HACKER". [on-line] Disponível na Internet via www url: http://www.geocities.com/surtus/hackers.txt. Documento capturado em 20 outubro de 2003.

"MUDGE". How to Write Buffer Overflows. [on-line]. Disponível na Internet via www url:http://l0pht.com/advisories/bufero.html. Arquivo capturado em de 1997.

ABSOLUTA. Resposta a Incidentes de Segurança. [on-line]. Disponível na Internet via www url:http://www.networkdesigners.com.br/Artigos/artigos.html. Artigo capturado em de 10 de novembro de 2003.

ALEPH ONE. Bugtraq Mailing List. [on-line]. Disponível na Internet via www. url:http://geek-girl.com/bugtraq/. Dados capturado em de 1997.

BEJTLICH, RICHARD. Interpreting Network Traffic: A Network Intrusion Detector's Look at Suspicious Events. [on-line]. Disponível na Internet via www. url:http://packetstorm.securify.com/papers/intv2.txt . Arquivo capturado em 10 de novembro de 2003.

CAIS, Centro de Atendimento a Incidentes de Segurança. [on-line] Disponível na Internet via www url: http://www.rnp.br/cais/. Informações capturadas em 28/11/2003.

CERT® Coordination Center Software Engineering Institute Carnegie Mellon University Pittsburgh. PA 15213-3890 U.S.A. [on-line] Disponível na Internet via www url: http://www.cert.org. Dados capturado em 28/11/2003.

CHAPMAN, D. BRENT; ZWICKY, ELIZABETH, D. Building Internet Firewalls. O'Reilly Associates, Inc. ed. 1, Setembro 1995.

CRABB , MICHELE, ET AL. Curmudgeon's Executive Summary. Sans, The

SANS Network Security Digest. 1997.

CYNC. Inside to Buffers Overflows. Mbc Security Labs. [on-line]. Disponível na Internet via www url:http://www.inet-sec.org/docs/bufferoverflow/englishbuff.txt. Arquivo capturado em 26 novembro de 2003.

FRANCISCO, ALEXANDRE JOSÉ. IEEE 802.1x. [on-line]. Disponível na Internet via www url:http://www.networkdesigners.com.br/Artigos/artigos.html. Artigo capturado em de 10 de novembro de 2003.

FYODOR. Remote detection of O.S IP STACK FINGERPRINTING [on-line]. Disponível na Internet via www. url:http://www.insecure.org Dados capturado em 10 de novembro de 2003.

KEATING, CHARLES. MILKEN, MICHAEL. HANSSEN, ROBERT. Incident Response: Investigating Computer Crime. ISBN 0-07-213182-9. Berkeley, California, 2001.

MARCELO, ANTONIO; CERQUEIRA, FELIPE, SARAIVA, FELIPE. Linux: ferramentas anti-hacker. Rio de Janeiro. Brasport, 2000. 7 - 12p.

MARIANO, ISMAEL DA SILVA. IPSec e DDoS, Aspectos de Segurança em Redes Tcp/Ip. Dissertação de metrado. COPPE – Coordenação do Programa de Pós-Graduação da UFRJ Mestrado em Engenharia de Sistemas. 2000. 56 – 65p.

MCCLURE, STUART, SCAMBRAY, JOEL, KURTZ, GEORGE. Hackers Expostos. São Paulo. Makron Books, 2000. 5 – 6p.

NATHANP, SMITH. Stack Smashing vulnerabilities in the UNIX Operating System. [on-line]. Disponível na Internet via www url:http://millcomm.com/nate/machines/security/stack-smashing/nate-buffer. Arquivo capturado em de 1997.

NIC BR SECURITY OFFICE. Práticas de Segurança para Administradores de Redes Internet. Versão 1.2. [on-line]. Disponível na Internet via www url: http://www.nbso.nic.br/ Arquivo capturado em 10 de dezembro de 2003.

NORTHCUTT, STEPHEN, NOVAK, JUDY, MCLACHLN, DONALD. Segurança e Prevenção em Redes. São Paulo. Berkeley, 2001.

RANUM, MARCUS. Taxonomy of Internet Attacks: What you can expect.Information Warehouse Inc, 1995. Disponível na Internet via www url:http://www.iwi.com. Arquivo capturado em novembro de 2003.

ROCHA, LUIS FERNANDO. Honeynet: eficácia no mapeamento das ameaças virtuais. Módulo Security Magazine. [on-line]. Disponível na Internet via www url:http://www.modulo.com.br/index.jsp?page=3&catid=7&objid=2056&pagenumber=0&idiom=0 Arquivo capturado em 5 de dezembro de 2003.

RUFINO, NELSON MURILO. Segurança Nacional: Técnicas e Ferramentas de Ataque e Defesa de Redes de Computadores. São Paulo. Novatec, 2002. 15 - 24p.

RUSSEL, RUSTY. 2.4 Packet Filtering HOWTO. [on-line]. Disponível na Internet via www url:http://netfilter.samba.org/documentation/HOWTO/pt/packet-filtering-HOWTO.html. Arquivo capturado em novembro de 2003

RUSSEL, RUSTY. Linux Iptables HOWTO. [on-line]. Disponível na Internet via www url:http://www.telematik.informatik.uni-karlsruhe.de/lehre/seminare/LinuxSem/downloads/netfilter/iptables-HOWTO.html. Arquivo capturado em 25 novembro de 2003.

SCHNEIDER, FREDB., et al. Trust in Cyberspace. Committee on Information Systems Trustworthiness, National Research Council. National Acadamy , 1999.

SIYAN, KARANJIT; HARE, CHRIS. Internet Firewalls and Network Security. New Riders Publishing. 1995. ISBN 1-56205-437-6.

SPOHN, M. A. Internet Firewalls. 1996. Trabalho Individual I n 554, Curso de Pós-Graduação em Ciências da Computação, Universidade Federal do Rio Grande do Sul.

SPTIZNER, LANCE. Passive Fingerprinting. [on-line]. Disponível na Internet via www url:http://www.securityfocus.com. Arquivo capturado em 11 de novembro de 2003.

THOMSEN, D.; SCHWARTAU, W. “Is Your Network Secure?”. Byte International Edition, V.21, n.1, p.155, Janeiro 1996.

UCHÔA, JOAQUIM QUINTEIRO. Segurança em redes e criptografia. Lavras: UFLA/FAEPE, Lavras – MG. 2003. 10 –12p.

ANEXO A

SERVIÇO PÚBLICO FEDERAL

PORTARIA N.º 267 de 20 DE OUTUBRO de 2003.

O **DIRETOR GERAL** do CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE CAMPOS, no uso de suas atribuições legais, que lhe confere a Lei 8.948 de 8/12/94, o decreto Presidencial de 18/01/1999, a portaria MEC nº 887 de 04/06/1999 e a Portaria MEC nº 1744 de 15/12/1999,

Considerando a necessidade de estabelecer normas do uso da REDE WEB do CEFET Campos,

Considerando a necessidade de melhorar o gerenciamento dos equipamentos e serviços de rede oferecidos, bem como evitar o mal-uso dos recursos disponíveis,

Considerando o Decreto nº 3.505, de 13/06/2000, que institui a Política de Segurança da Informação nos órgãos e entidades da Administração Pública Federal,

Considerando a Portaria nº 316, de 12/07/2000, do CEFET CAMPOS, que estabelece normas de conteúdo disponível na REDE WEB do CEFET CAMPOS,

Considerando a Medida Provisória nº 2.200-2, de 24/08/2001, que institui a Infra-Estrutura de Chaves Públicas Brasileira – ICP-Brasil, transforma o Instituto Nacional de Tecnologia da Informação em autarquia, e dá outras providências,

Considerando a Recomendação nº 01, de 09/12/2002, da Secretaria de Logística e Tecnologia da Informação do Ministério do Planejamento, Orçamento e Gestão,

Resolve:

Estabelecer normas internas para utilização de correio eletrônico, utilização do acesso à Internet e utilização da Rede interna do CEFET Campos.

Normas para Utilização de Correio Eletrônico no CEFETCampos.

Art 1º A presente Norma tem como objetivo estabelecer regras para disponibilização e utilização dos serviços de Correio Eletrônico providos pelo CEFETCampos, visando disciplinar a troca de mensagens eletrônicas e estabelecer critérios para que os mesmos sejam utilizados em conformidade com a legislação brasileira aplicável.

Art. 2º Os seguintes conceitos se aplicam a essa norm

I. Considera-se Serviço de Correio Ele ı Sistema de mensageira utilizado para criar, enviar, encaminhar, responder, transmitir, arquivar, manter, copiar, mostrar, ler ou imprimir informações com o propósito de comunicação entre redes de computadores ou entre pessoas ou grupos;
II. Considera-se Mensagem de Correio Eletrônico um ou mais registros eletrônicos de computador ou mensagens criadas, enviadas, encaminhadas, respondidas, transmitidas, arquivadas, mantidas, copiadas, mostradas, lidas ou impressas por um ou vários sistemas ou serviços de correio eletrônicos;
III. Considera-se Usuário a pessoa física, seja servidor, empregado ou prestador de serviços;
IV. Considera-se Identificação do Usuário ou Nome do Usuário a forma com que o usuário é conhecido junto ao ambiente de informática do Órgão, onde o conjunto Identificação e Senha permitem que ações e ferramentas sejam utilizadas de acordo com o perfil desse usuário;
V. Considera-se Caixa Postal a área de armazenamento que contém todas as pastas do Correio Eletrônico, dentre as quais podem ser: Caixa de Entrada - Área predefinida que armazena mensagens recebidas; Caixa de Saída - Área predefinida que armazena as mensagens enviadas, até que elas sejam entregues.

Da Utilização do Correio Eletrônico

Art. 3º O acesso ao correio eletrônico se dá pelo conjunto Identificação do Usuário, Caixa Postal e Senha que é pessoal e intransferível, não podendo um mesmo usuário ter mais de uma conta de correio eletrônico.

Art. 4° Parágrafo único. Quando for criada conta de correio eletrônico para unidades administrativas, grupos de trabalho e outros usuários despersonalizados, deverá ser identificada junto ao administrador do serviço a pessoa responsável pelo uso do correio destes usuários.

Art. 5º É vedada tentativa de acesso não autorizado às caixas postais de terceiros.

Art. 6º Prestadores de serviços terceirizados e estagiários poderão durante o período de prestação dos serviços, a critério do responsável pela área onde está sendo prestado o serviço terceirizado ou estágio e no interesse do serviço, ter acesso ao correio eletrônico

institucional, observando as normas aqui enumeradas. Devendo o responsável encaminhar por escrito tal solicitação.

Art. 7º O remetente deve se identificar de forma clara e evidente em todas as suas comunicações eletrônicas, não sendo permitidas alterações ou manipulações da origem das postagens.
Parágrafo único. As mensagens deverão ser redigidas de forma clara e sucinta, devendo conter o grau de formalidade compatível com o destinatário e o assunto tratado.

Art. 8º É vedado o envio intencional e o armazenamento de mensagens contendo:

I. - material obsceno, ilegal ou antiético;
II. - anúncios publicitários;
III. - listas de endereços eletrônicos dos usuários do Correio Eletrônico do CEFETCampos;
IV. - vírus ou qualquer outro tipo de programa danoso;
V. - material protegido por leis de propriedade intelectual;
VI. - entretenimentos e "correntes";
VII. -material preconceituoso ou discriminatório;
VIII. - material de natureza político-partidária ou sindical, que promova a eleição de candidatos para cargos públicos eletivos, clubes, associações e sindicatos; e
IX. - assuntos ofensivos.

Art. 9º Não é permitida a transmissão, recebimento e/ou armazenamento de mensagens contendo:

I - músicas, vídeos ou animações que não interesse específico do trabalho; e

II - programas de computador que não sejam destinados ao desempenho de suas funções ou que possam ser considerados nocivos ao ambiente de rede do CEFETCampos.

Das Competências

Art. 10 Os administradores de rede da GTI, lotados na Coordenadoria de Administração de Rede, são os responsáveis pela administração de seu Serviço de Correio Eletrônico.

Art. 11 A administração do correio eletrônico deve estabelecer e manter um processo sistemático para gravação e retenção de arquivos de registro de mensagens de correio eletrônico. Estes arquivos deverão ser mantidos por um prazo mínimo de 12 meses e os conteúdos de caixas postais por um período de, no mínimo 05 dias e no máximo 45 dias.
Parágrafo único. A eliminação dos arquivos de registro de mensagens e de caixas postais deverá ser adiada em caso de auditoria, ou qualquer outro tipo de notificação administrativa ou judicial.

Art. 12 A Administração do Correio Eletrônico deverá manter ferramenta para atualização de dados cadastrais dos usuários, bem como, estabelecer a periodicidade em

que deverá promover campanhas neste sentido.

Art. 13 Compete ao usuário:

I. gerenciar compromissos, contatos, tarefas, arquivos e atividades;
II. utilizar o correio eletrônico institucional para os objetivos e funções próprios e inerentes às suas atribuições funcionais;
III. eliminar periodicamente as mensagens contidas nas caixas postais;
IV. não permitir acesso de terceiros ao correio eletrônico através de sua senha; e
V. atualizar seus dados cadastrais utilizando os meios disponíveis.
VI. se responsabilizar por todo e qualquer uso de suas contas. Isto inclui escolher senhas seguras e garantir que proteções de arquivos sejam configuradas corretamente.

Parágrafo único. O usuário deve estar ciente que os administradores dos recursos fazem checagens periódicas de segurança, incluindo a verificação de senhas. Qualquer usuário detectado com uma "senha ruim" será notificado via e-mail. O usuário deverá trocar sua senha em no máximo cinco dias úteis, data a partir da qual sua conta será bloqueada. Para desbloqueio será necessária sua presença junto aos administradores para que troque a referida senha.

Art. 14 Compete à administração do serviço de correio eletrônico:

I. - garantir a disponibilidade do serviço de correio eletrônico em níveis de serviço adequados à necessidade do trabalho;
II. - garantir a recuperação de mensagens em caso de danos ao ambiente, observando o prazo especificado no Art. 11º.

Art. 15 Para poder utilizar o serviço de correio eletrônico institucional, o usuário deve tomar conhecimento, por meio eletrônico ou impresso, de termo de responsabilidade, tomando ciência e concordando com os termos desta Norma.

Art. 16 Os usuários deverão notificar a administração do correio eletrônico e sua chefia imediata ou superior, quando do recebimento de mensagens que contrariem o disposto nesta Norma.

Da Apuração de Responsabilidades

Art. 17 Havendo indícios de que mensagens veiculadas pelo correio eletrônico possam ocasionar quebra de segurança ou violação de quaisquer das vedações constantes deste ou outro ato normativo, a administração do correio eletrônico adotará, imediatamente, medidas para a sua apuração, utilizando-se, para tanto, dos meios e procedimentos legalmente previstos.

Art. 18 Caracterizado o descumprimento de qualquer dos itens desta Norma, caberá à administração do correio eletrônico informar a chefia imediata ou superior do usuário, apresentando o ocorrido a fim de encaminhar as providências de apuração de responsabilidades.

Disposições Gerais Relativas ao Correio Eletrônico

Art. 19 As solicitações de novas caixas postais deverão ser encaminhadas a Gerência da Tecnologia da Informação, pela chefia imediata ou superior com os respectivos dados cadastrais.
Parágrafo único. No caso de afastamento definitivo, a GTI providenciará a exclusão da caixa postal, após aviso via e-mail.

Art. 20 Cabe à chefia imediata ou superior comunicar a GTI o desligamento de empregados terceirizados, temporários e estagiários sob sua responsabilidade para a exclusão definitiva da caixa postal, sob pena de exclusão pela GTI sem aviso prévio.

Art. 21 A caixa postal sem movimentação por um período igual ou superior a três meses será bloqueada automaticamente pela GTI.

Art. 22 Os usuários são responsáveis por utilizar o mínimo de recursos possível, limitando-se a atividades acadêmicas e tendo em mente que os recursos são compartilhados entre vários usuários. Isto inclui que o usuário não deve tentar "derrubar" servidores ou qualquer estação de trabalho, bem como manter um efetivo gerenciamento de sua quota de espaço em disco. Além disto, o espaço reservado no servidor não deve ser usado para distribuição de programas de terceiros, principalmente aquele de fácil obtenção (clientes de ftp, icq, e-mail, irc, etc). Caso haja qualquer tentativa de “invasão” dos servidores de rede, a GTI tomará as medidas cabíveis junto à Direção Geral da Instituição.

Norma para Utilização do acesso a Internet

A presente Norma tem como objetivo estabelecer regras para a utilização do serviço de acesso a Internet utilizando a rede do CEFETCampos.
Partindo do princípio de que todo usuário, seja servidor ou aluno, pode utilizar este recurso, acrescenta que somente o uso indevido fará com que o usuário não possa mais acessar este serviço.

Art. 23 Considera-se Serviço de Acesso a Internet, aquele que prove e disponibiliza o acesso a qualquer informação disponibilizada na rede mundial.

Art. 24 Não é permitido utilizar este recurso através de acesso discado não-autorizado.

Art. 25 Não é permitido utilizar estes recursos como ponto de partida para tentar invadir outros sistemas.

Art. 26 A principio, todo e qualquer usuário pode usufruir dos recursos de Internet disponíveis no CEFETCampos, desde que não usem para acesso a sites do tipo : pedófilo, pornográfico, de acesso a material de hakers e itens do gênero. Caso seja

constatado o uso da internet para estes fins, a máquina do usuário entrara em quarentena onde será bloqueada nela o acesso a recursos que possam fomentar estas atividades. Conhecendo o usuário será bloqueado o acesso do usuário ao recurso usado indevidamente.

Art. 27 Não é permitido tentar obter acesso não-autorizado de contas de administradores ou de qualquer outra conta não pertencente ao usuário dentro destes recursos.

Art. 28 Não é permitido o uso de softwares do tipo chat, ICQ, servidores de arquivos (MP3, AVI, MPG, etc), bem como outros que venham a degradar o desempenho total do acesso a Internet.

Norma para Utilização da Rede Interna do CEFETCampos

A rede do CEFETCampos está dividida em Educacional e Administrativa não sendo permitido desta forma o acesso de alunos a Rede Administrativa, salvo os em estágio ou trabalho, com o consentimento do responsável pelo setor.
Ao utilizar o acesso a rede LAN/WAN, o usuário destes recursos (equipamentos e serviços) deverá conhecer e estar de acordo com os seguintes itens:

Art. 29 Não é permitido tentar obter acesso não-autorizado de contas de administradores ou de qualquer outra conta não pertencente ao usuário ` ` tes recursos.

Art. 30 Qualquer compartilhamento de pastas e ou arquıvos deve ser feito com senha.

Art. 31 Qualquer usuário que encontrar um possível problema de segurança nestes recursos é obrigado a reportar isto aos administradores dos mesmos.

Art. 32 Os usuários são responsáveis por todo e qualquer uso de suas contas. Isto inclui escolher senhas seguras.

Art. 33 Os setores que solicitarem a criação de conta e senha para usuários temporários (bolsista e estagiários), o responsável devera fazê-lo por escrito informando o prazo de duração deste trabalho e em que período ele ocorre, devendo ainda comunicar o desligamento do mesmo para que sua conta seja desativada.

Art. 34 A chefia imediata deverá comunicar o nome do servidor que mudar de setor ou que não pertence mais ao quadro do CEFETCampos, sob pena de exclusão do registro pela GTI sem aviso prévio, no segundo caso.

Art. 35 O usuário deve estar ciente que suas ações poderão ser monitoradas caso haja suspeitas de mal-uso dos recursos.

Art. 36 As pastas compartilhadas em servidores para fins de cópia de segurança são exclusivamente para conteúdo pertinentes a instituição, sendo vedado o seu uso para fazer copia de segurança de conteúdo particular.

Das Competências:

Art. 37 Compete a Coordenadoria de Administração de Rede da GTI manter um sistema de copia dos arquivos da rede, como proteção de segurança, desde que solicitado pela Coordenação e que não haja impedimento técnico para tal tarefa.

Das Disposições Gerais
Art. 38 A presente norma não é de caráter definitivo devendo ser periodicamente atualizada de forma a refletir as mudanças na organização.
Art. 39 Caberá a GTI esclarecer os casos omissos a esta Norma.

Luiz Augusto Caldas Pereira
Diretor Geral do CEFET Campos

ANEXO B

Script para atualizar antivívus no servidor de e-mail, no ftp para os clientes e enviar e-mail para todos os usuários.

```
#!/bin/sh
# by WSV
#echo "apagando dats"
rm -rf /tmp/dat-*.tar
rm -rf /tmp/sdat*.exe

#echo "pegando dir atual"
DIRATUAL=$(pwd)

#echo "pegando datafiles"
. /etc/links/pegar_dat

#echo "indo para o dir atual"
cd /$DIRATUAL

#echo "vefificando se existe o dat.tar"
if [ -f /tmp/dat-????.tar ]; then

#echo "pegando a versao do data atual"
DATATUAL=$(/usr/local/uvscan/uvscan --version|grep "Virus data"|cut -d" " -f4|cut -dv
-f2)
DATNOVO=$(ls /tmp/dat-*.tar|cut -d- -f2|cut -d. -f1)

#    echo "verificando se sao diferentes"
   if [ "$DATATUAL" -lt "$DATNOVO" ]; then
      mkdir /tmp/dat-updates 2> /dev/null
      cd /tmp/dat-updates

#       echo "descompactando"
      tar -xf /tmp/dat-*.tar

#       echo "movendo para a pasta do uvscan"
      mv /tmp/dat-updates/*.dat /usr/local/uvscan
      cd /$DIRATUAL

#       echo "removendo pasta dat-updates"
      rm -rf /tmp/dat-updates
      rm -rf /tmp/dat-updates
      DATATUAL=$(/usr/local/uvscan/uvscan --version|grep "Virus data"|cut -d" " -
f4|cut -dv -f2)
      cd /etc/links
```

```
      ASSUNTO="ATUALIZAÇÃO DO ANTIVIRUS - SDAT $DATNOVO"
      MENSAGEM="NOVA ATUALIZAÇÃO DO AV DA MCAFEE, VERSÃO $DATNOVO. FAVOR FAZER DOWNLOAD ARQUIVO COM .EXE E EM SEGUIDA EXECUTAR. O ENDEREÇO É ftp://ftp.cefetcampos.br/softwares/AntiVirus/Update/mcafee/versao_4 "
      DESTINATARIO="todos@cefetcampos.br"
      . /etc/links/aviso_av

      SDATNOVO=$(ls sdat????.exe)
   fi
else
   echo "não há data file novo o atual é $DATATUAL, e o novo é $DATNOVO"
fi
```

SCRIPT PEGAR_DAT

```
#!/bin/sh
#pegar_dat
#by WSV
ftp -in <<EOF
open ftp.nai.com
user anonymous eu@eu
bin
cd /pub/antivirus/datfiles/4.x
lcd /tmp
mget dat-*.tar
mget sdat*.exe
bye
EOF
```

SCRIPT AVISO_AV

```
#!/bin/sh
#  aviso_av
# script para envio de mensagens
mail -s "$ASSUNTO" "$DESTINATARIO"<<EOF
########################################
MENSAGEM AUTOMáTICA DO YODA
########################################

$MENSAGEM

OBS.:
- VERIFIQUE O ARQUIVO DE LOG DE % DE PERDAS EM /var/log/perda.log
- TELEFONES:    EMERGÊNCIA = 190    WILLIAM = 98219095
EOF
```

ANEXO C

Script para elaboração do relatório via e-mail da estatística de e-mails com vírus.

```
#!/bin/sh
# by WSV
destinatario=gti_redes@cefetcampos.br
pasta=/var/virusmails/root
ano=`date|cut -d" " -f 6`
cd /tmp
ls -l $pasta|grep "\-2003"|awk '{print $6 $7}' > /tmp/stat_virus.txt
dias=`cat /tmp/stat_virus.txt|uniq`
> /tmp/mail_stat_virus.txt
echo "  " >> /tmp/mail_stat_virus.txt
echo "ESTATÍSTICA DE VÍRUS NO EMAIL - CEFET " >> /tmp/mail_stat_virus.txt
echo "  " >> /tmp/mail_stat_virus.txt
/usr/local/uvscan/uvscan --version >> /tmp/mail_stat_virus.txt
echo "  " >> /tmp/mail_stat_virus.txt
echo " Total de e-mail com vírus `cat /tmp/stat_virus.txt|wc -l` " >>
/tmp/mail_stat_virus.txt
echo "  " >> /tmp/mail_stat_virus.txt

cat /var/log/scanmails/logfile | grep Found |sort|awk '{print $3}'>
/tmp/stat_virus_nome.txt
virus=`cat /tmp/stat_virus_nome.txt|uniq`

echo " Quantidade e nome dos vírus" >> /tmp/mail_stat_virus.txt
echo "  " >> /tmp/mail_stat_virus.txt
for v in $virus
 do
  echo "`cat /tmp/stat_virus_nome.txt|grep $v|wc -l` $v" >> /tmp/mail_stat_virus.txt
 done
echo "  " >> /tmp/mail_stat_virus.txt

echo " Quantidade de vírus por dia em email encontrados no ano de $ano" >>
/tmp/mail_stat_virus.txt
echo "  " >> /tmp/mail_stat_virus.txt
for d in $dias
 do
  echo "`cat /tmp/stat_virus.txt|grep $d|wc -l` $d" >> /tmp/mail_stat_virus.txt
 done
echo " ###############  FIM  ################ " >> /tmp/mail_stat_virus.txt
cat /tmp/mail_stat_virus.txt|mail -s "Estatística de vírus em email" "$destinatario"
```

ANEXO D

```
#!/bin/sh -x
#
#  This is automatically generated file. DO NOT MODIFY !
#  Firewall Builder  fwb_ipt v1.0.6-
#  Generated Mon Feb 23 21:51:58 2004 BRT by root
#
check() {
  if test ! -x "$1"; then
    echo "$1 not found or is not executable"
    exit 1
  fi
}

log() {
  if test -x "$LOGGER"; then
    logger -p info "$1"
  fi
}

MODPROBE="/sbin/modprobe"
IPTABLES="/sbin/iptables"
IP="/sbin/ip"
LOGGER="/usr/bin/logger"

check $MODPROBE
check $IPTABLES
check $IP

cd /etc || exit 1

log "Activating firewall script generated Mon Feb 23
21:51:58 2004 BRT by root"

MODULE_DIR="/lib/modules/`uname -
r`/kernel/net/ipv4/netfilter/"
MODULES="ipt_conntrack ipt_conntrack_ftp ipt_nat_ftp
ipt_conntrack_irc ipt_nat_irc"
for module in $(echo $MODULES); do
  if [ -e "${MODULE_DIR}/${module}.o" -o -e
"${MODULE_DIR}/${module}.o.gz" ]; then
    $MODPROBE -k ${module} ||  exit 1
  fi
done
```

```
FWD=`cat /proc/sys/net/ipv4/ip_forward`
echo "0" > /proc/sys/net/ipv4/ip_forward
echo 1 > /proc/sys/net/ipv4/conf/all/rp_filter
echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
echo 1 >
/proc/sys/net/ipv4/icmp_ignore_bogus_error_responses
echo 60 > /proc/sys/net/ipv4/tcp_fin_timeout
echo 7200 > /proc/sys/net/ipv4/tcp_keepalive_intvl
echo 1 > /proc/sys/net/ipv4/tcp_window_scaling
echo 0 > /proc/sys/net/ipv4/tcp_sack
echo 1 > /proc/sys/net/ipv4/tcp_syncookies
echo 0 > /proc/sys/net/ipv4/tcp_timestamps


$IPTABLES -P OUTPUT  DROP
$IPTABLES -P INPUT   DROP
$IPTABLES -P FORWARD DROP

cat /proc/net/ip_tables_names | while read table; do
  $IPTABLES -t $table -L -n | while read c chain rest; do
      if test "X$c" = "XChain" ; then
        $IPTABLES -t $table -F $chain
      fi
  done
  $IPTABLES -t $table -X
done

#  Rule 0(NAT)
#
$IPTABLES -t nat -A PREROUTING  -p tcp  -s
200.167.129.16/28 --source-port 1024:65535  --destination-
port 80 -j REDIRECT --to-ports 3128
#

$IPTABLES -t drop -A DROPPING -j LOG  --log-level 6 --log-
prefix "RULE %N -- %A "  --log-tcp-sequence  --log-tcp-
options  --log-ip-options
$IPTABLES -A INPUT   -m state --state ESTABLISHED,RELATED -
j ACCEPT
$IPTABLES -A OUTPUT  -m state --state ESTABLISHED,RELATED -
j ACCEPT
$IPTABLES -A FORWARD -m state --state ESTABLISHED,RELATED -
j ACCEPT

# Rule 0(global)
#
$IPTABLES -N RULE_0
```

```
$IPTABLES -A INPUT -p tcp  -s 200.167.129.34 --source-port
1024:65535 --destination-port 80 -m state --state NEW -j
RULE_0
$IPTABLES -A OUTPUT -p tcp  -s 200.167.129.34 --source-port
1024:65535 --destination-port 80 -m state --state NEW -j
RULE_0
$IPTABLES -A RULE_0 -j LOG  --log-level 6 --log-prefix
"RED-"  --log-tcp-sequence  --log-tcp-options  --log-ip-
options
$IPTABLES -A RULE_0 -j ACCEPT

# Rule 1(global)
#
$IPTABLES -N RULE_1
$IPTABLES -A INPUT -p tcp  -s 200.167.129.16/28 --source-
port 1024:65535 --destination-port 3128 -m state --state
NEW -j RULE_1
$IPTABLES -A FORWARD -p tcp  -s 200.167.129.16/28 --source-
port 1024:65535 --destination-port 3128 -m state --state
NEW -j RULE_1
$IPTABLES -A RULE_1 -j LOG  --log-level 6 --log-prefix
"RED-"  --log-tcp-sequence  --log-tcp-options  --log-ip-
options
$IPTABLES -A RULE_1 -j ACCEPT

# Rule 2(global)
#
$IPTABLES -N RULE_2
$IPTABLES -A INPUT -p tcp  -s 200.167.129.16/28 --source-
port 1024:65535 --destination-port 80 -m state --state NEW
-j RULE_2
$IPTABLES -A INPUT -p tcp  -s 200.167.129.16/28 --source-
port 1024:65535 --destination-port 20:21 -m state --state
NEW -j RULE_2
$IPTABLES -A INPUT -p tcp  -s 200.167.129.16/28 --source-
port 1024:65535 --destination-port 443 -m state --state NEW
-j RULE_2
$IPTABLES -A INPUT -p tcp  -s 200.167.129.16/28 --source-
port 1024:65535 --destination-port 110 -m state --state NEW
-j RULE_2
$IPTABLES -A INPUT -p tcp  -s 200.167.129.16/28 --source-
port 1024:65535 --destination-port 143 -m state --state NEW
-j RULE_2
$IPTABLES -A INPUT -p tcp  -s 200.167.129.16/28 --source-
port 1024:65535 --destination-port 220 -m state --state NEW
-j RULE_2
$IPTABLES -A INPUT -p udp  -s 200.167.129.16/28 --source-
```

```
port 1024:65535 --destination-port 53 -m state --state NEW
-j RULE_2
$IPTABLES -A INPUT -p udp  -s 200.167.129.16/28 --source-
port 53 --destination-port 53 -m state --state NEW -j
RULE_2
$IPTABLES -A FORWARD -p tcp  -s 200.167.129.16/28 --source-
port 1024:65535 --destination-port 80 -m state --state NEW
-j RULE_2
$IPTABLES -A FORWARD -p tcp  -s 200.167.129.16/28 --source-
port 1024:65535 --destination-port 20:21 -m state --state
NEW -j RULE_2
$IPTABLES -A FORWARD -p tcp  -s 200.167.129.16/28 --source-
port 1024:65535 --destination-port 443 -m state --state NEW
-j RULE_2
$IPTABLES -A FORWARD -p tcp  -s 200.167.129.16/28 --source-
port 1024:65535 --destination-port 110 -m state --state NEW
-j RULE_2
$IPTABLES -A FORWARD -p tcp  -s 200.167.129.16/28 --source-
port 1024:65535 --destination-port 143 -m state --state NEW
-j RULE_2
$IPTABLES -A FORWARD -p tcp  -s 200.167.129.16/28 --source-
port 1024:65535 --destination-port 220 -m state --state NEW
-j RULE_2
$IPTABLES -A FORWARD -p udp  -s 200.167.129.16/28 --source-
port 1024:65535 --destination-port 53 -m state --state NEW
-j RULE_2
$IPTABLES -A FORWARD -p udp  -s 200.167.129.16/28 --source-
port 53 --destination-port 53 -m state --state NEW -j
RULE_2
$IPTABLES -A RULE_2 -j LOG  --log-level 6 --log-prefix
"BASICOS-"  --log-tcp-sequence  --log-tcp-options  --log-
ip-options
$IPTABLES -A RULE_2 -j ACCEPT

# Rule 3(global)
#
$IPTABLES -N RULE_3
$IPTABLES -A OUTPUT -p tcp --source-port 1024:65535  -d
200.167.129.16/28 --destination-port 80 -m state --state
NEW -j RULE_3
$IPTABLES -A OUTPUT -p tcp --source-port 1024:65535  -d
200.167.129.16/28 --destination-port 20:21 -m state --state
NEW -j RULE_3
$IPTABLES -A OUTPUT -p tcp --source-port 1024:65535  -d
200.167.129.16/28 --destination-port 443 -m state --state
NEW -j RULE_3
$IPTABLES -A OUTPUT -p tcp --source-port 1024:65535  -d
```

```
200.167.129.16/28 --destination-port 110 -m state --state
NEW -j RULE_3
$IPTABLES -A OUTPUT -p tcp --source-port 1024:65535  -d
200.167.129.16/28 --destination-port 143 -m state --state
NEW -j RULE_3
$IPTABLES -A OUTPUT -p tcp --source-port 1024:65535  -d
200.167.129.16/28 --destination-port 220 -m state --state
NEW -j RULE_3
$IPTABLES -A OUTPUT -p udp --source-port 1024:65535  -d
200.167.129.16/28 --destination-port 53 -m state --state
NEW -j RULE_3
$IPTABLES -A OUTPUT -p udp --source-port 53  -d
200.167.129.16/28 --destination-port 53 -m state --state
NEW -j RULE_3
$IPTABLES -A FORWARD -p tcp --source-port 1024:65535  -d
200.167.129.16/28 --destination-port 80 -m state --state
NEW -j RULE_3
$IPTABLES -A FORWARD -p tcp --source-port 1024:65535  -d
200.167.129.16/28 --destination-port 20:21 -m state --state
NEW -j RULE_3
$IPTABLES -A FORWARD -p tcp --source-port 1024:65535  -d
200.167.129.16/28 --destination-port 443 -m state --state
NEW -j RULE_3
$IPTABLES -A FORWARD -p tcp --source-port 1024:65535  -d
200.167.129.16/28 --destination-port 110 -m state --state
NEW -j RULE_3
$IPTABLES -A FORWARD -p tcp --source-port 1024:65535  -d
200.167.129.16/28 --destination-port 143 -m state --state
NEW -j RULE_3
$IPTABLES -A FORWARD -p tcp --source-port 1024:65535  -d
200.167.129.16/28 --destination-port 220 -m state --state
NEW -j RULE_3
$IPTABLES -A FORWARD -p udp --source-port 1024:65535  -d
200.167.129.16/28 --destination-port 53 -m state --state
NEW -j RULE_3
$IPTABLES -A FORWARD -p udp --source-port 53  -d
200.167.129.16/28 --destination-port 53 -m state --state
NEW -j RULE_3
$IPTABLES -A RULE_3 -j LOG  --log-level 6 --log-prefix
"BASICOS-"  --log-tcp-sequence  --log-tcp-options  --log-
ip-options
$IPTABLES -A RULE_3 -j ACCEPT

# Rule 4(global)
#
$IPTABLES -N RULE_4
$IPTABLES -A INPUT -p tcp  -s 200.167.129.20 --source-port
```

```
1024:65535 --destination-port 25 -m state --state NEW -j
RULE_4
$IPTABLES -A INPUT -p tcp  -s 200.167.129.23 --source-port
1024:65535 --destination-port 25 -m state --state NEW -j
RULE_4
$IPTABLES -A INPUT -p tcp  -s 200.167.129.25 --source-port
1024:65535 --destination-port 25 -m state --state NEW -j
RULE_4
$IPTABLES -A FORWARD -p tcp  -s 200.167.129.20 --source-
port 1024:65535 --destination-port 25 -m state --state NEW
-j RULE_4
$IPTABLES -A FORWARD -p tcp  -s 200.167.129.23 --source-
port 1024:65535 --destination-port 25 -m state --state NEW
-j RULE_4
$IPTABLES -A FORWARD -p tcp  -s 200.167.129.25 --source-
port 1024:65535 --destination-port 25 -m state --state NEW
-j RULE_4
$IPTABLES -A RULE_4 -j LOG  --log-level 6 --log-prefix
"SMTP_OUT-"  --log-tcp-sequence  --log-tcp-options  --log-
ip-options
$IPTABLES -A RULE_4 -j ACCEPT

# Rule 5(global)
#
$IPTABLES -N RULE_5
$IPTABLES -A OUTPUT -p tcp --source-port 1024:65535  -d
200.167.129.20 --destination-port 25 -m state --state NEW -
j RULE_5
$IPTABLES -A OUTPUT -p tcp --source-port 1024:65535  -d
200.167.129.23 --destination-port 25 -m state --state NEW -
j RULE_5
$IPTABLES -A OUTPUT -p tcp --source-port 1024:65535  -d
200.167.129.25 --destination-port 25 -m state --state NEW -
j RULE_5
$IPTABLES -A FORWARD -p tcp --source-port 1024:65535  -d
200.167.129.20 --destination-port 25 -m state --state NEW -
j RULE_5
$IPTABLES -A FORWARD -p tcp --source-port 1024:65535  -d
200.167.129.23 --destination-port 25 -m state --state NEW -
j RULE_5
$IPTABLES -A FORWARD -p tcp --source-port 1024:65535  -d
200.167.129.25 --destination-port 25 -m state --state NEW -
j RULE_5
$IPTABLES -A RULE_5 -j LOG  --log-level 6 --log-prefix
"SMTP_IN-"  --log-tcp-sequence  --log-tcp-options  --log-
ip-options
$IPTABLES -A RULE_5 -j ACCEPT
```

```
# Rule 6(global)
#
$IPTABLES -N RULE_6
$IPTABLES -A INPUT  -s 127.0.0.1 -m state --state NEW -j
RULE_6
$IPTABLES -A FORWARD  -s 127.0.0.1 -m state --state NEW -j
RULE_6
$IPTABLES -A RULE_6 -j LOG  --log-level 6 --log-prefix
"RULE 6 -- ACCEPT "  --log-tcp-sequence  --log-tcp-options
--log-ip-options
$IPTABLES -A RULE_6 -j ACCEPT

# Rule 7(global)
#
$IPTABLES -N RULE_7
$IPTABLES -A INPUT -p icmp  -s 200.167.129.16/28 --icmp-
type 8/0  -m state --state NEW -j RULE_7
$IPTABLES -A FORWARD -p icmp  -s 200.167.129.16/28 --icmp-
type 8/0  -m state --state NEW -j RULE_7
$IPTABLES -A RULE_7 -j LOG  --log-level 6 --log-prefix
"ICMP_REQ-"  --log-tcp-sequence  --log-tcp-options  --log-
ip-options
$IPTABLES -A RULE_7 -j ACCEPT

# Rule 8(global)
#
$IPTABLES -N RULE_8
$IPTABLES -A OUTPUT -p icmp  -d 200.167.129.16/28 --icmp-
type 0/0  -m state --state NEW -j RULE_8
$IPTABLES -A FORWARD -p icmp  -d 200.167.129.16/28 --icmp-
type 0/0  -m state --state NEW -j RULE_8
$IPTABLES -A RULE_8 -j LOG  --log-level 6 --log-prefix
"ICMP_RESP"  --log-tcp-sequence  --log-tcp-options  --log-
ip-options
$IPTABLES -A RULE_8 -j ACCEPT

# Rule 9(global)
#
$IPTABLES -N RULE_9
$IPTABLES -A FORWARD -p tcp  -s 200.222.17.0/24 --source-
port 1024:65535  -d 200.167.129.16/28 --destination-port
60000 -m state --state NEW -j RULE_9
$IPTABLES -A FORWARD -p tcp  -s 200.20.228.0/24 --source-
port 1024:65535  -d 200.167.129.16/28 --destination-port
60000 -m state --state NEW -j RULE_9
$IPTABLES -A FORWARD -p tcp  -s 200.167.209.0/24 --source-
```

```
port 1024:65535  -d 200.167.129.16/28 --destination-port
60000 -m state --state NEW -j RULE_9
$IPTABLES -A FORWARD -p tcp  -s 200.167.129.32/28 --source-
port 1024:65535  -d 200.167.129.16/28 --destination-port
60000 -m state --state NEW -j RULE_9
$IPTABLES -A FORWARD -p tcp  -s 200.167.129.16/28 --source-
port 1024:65535  -d 200.167.129.16/28 --destination-port
60000 -m state --state NEW -j RULE_9
$IPTABLES -A FORWARD -p tcp  -s 200.222.17.0/24 --source-
port 1024:65535  -d 200.167.129.16/28 --destination-port 22
-m state --state NEW -j RULE_9
$IPTABLES -A FORWARD -p tcp  -s 200.20.228.0/24 --source-
port 1024:65535  -d 200.167.129.16/28 --destination-port 22
-m state --state NEW -j RULE_9
$IPTABLES -A FORWARD -p tcp  -s 200.167.209.0/24 --source-
port 1024:65535  -d 200.167.129.16/28 --destination-port 22
-m state --state NEW -j RULE_9
$IPTABLES -A FORWARD -p tcp  -s 200.167.129.32/28 --source-
port 1024:65535  -d 200.167.129.16/28 --destination-port 22
-m state --state NEW -j RULE_9
$IPTABLES -A FORWARD -p tcp  -s 200.167.129.16/28 --source-
port 1024:65535  -d 200.167.129.16/28 --destination-port 22
-m state --state NEW -j RULE_9
$IPTABLES -A RULE_9 -j LOG  --log-level 4 --log-prefix
"SSH-"  --log-tcp-sequence  --log-tcp-options  --log-ip-
options
$IPTABLES -A RULE_9 -j ACCEPT

# Rule 10(global)
#
$IPTABLES -N RULE_10
$IPTABLES -A OUTPUT  -s 224.0.0.0/4  -d 200.167.129.16/28 -
j RULE_10
$IPTABLES -A FORWARD  -s 192.0.2.0/24  -d 200.167.129.16/28
-j RULE_10
$IPTABLES -A FORWARD  -s 10.0.0.0/8  -d 200.167.129.16/28 -
j RULE_10
$IPTABLES -A FORWARD  -s 172.16.0.0/12  -d
200.167.129.16/28 -j RULE_10
$IPTABLES -A FORWARD  -s 192.168.0.0/24  -d
200.167.129.16/28 -j RULE_10
$IPTABLES -A FORWARD  -s 169.254.0.0/16  -d
200.167.129.16/28 -j RULE_10
$IPTABLES -A FORWARD  -s 192.88.99.0/24  -d
200.167.129.16/28 -j RULE_10
$IPTABLES -A FORWARD  -s 198.18.0.0/23  -d
200.167.129.16/28 -j RULE_10
```

```
$IPTABLES -A FORWARD  -s 240.0.0.0/5  -d 200.167.129.16/28
-j RULE_10
$IPTABLES -A RULE_10 -j LOG  --log-level 1 --log-prefix
"IP_SPOOFADO-"  --log-tcp-sequence  --log-tcp-options  --
log-ip-options
$IPTABLES -A RULE_10 -j DROP

# Rule 11(global)
#
$IPTABLES -N RULE_11
$IPTABLES -A FORWARD -p tcp  -s 200.167.129.16/28 --source-
port 1024:65535  -d 200.201.174.0/24 --destination-port
2631 -m state --state NEW -j RULE_11
$IPTABLES -A RULE_11 -j LOG  --log-level 6 --log-prefix
"RULE 11 -- ACCEPT "  --log-tcp-sequence  --log-tcp-options
--log-ip-options
$IPTABLES -A RULE_11 -j ACCEPT

# Rule 12(global)
#
$IPTABLES -N RULE_12
$IPTABLES -A FORWARD  -s 200.167.129.16/28  -d
200.252.232.0/24 -m state --state NEW -j RULE_12
$IPTABLES -A FORWARD  -s 200.167.129.16/28  -d
200.252.141.0/24 -m state --state NEW -j RULE_12
$IPTABLES -A RULE_12 -j LOG  --log-level 6 --log-prefix
"RULE 12 -- ACCEPT "  --log-tcp-sequence  --log-tcp-options
--log-ip-options
$IPTABLES -A RULE_12 -j ACCEPT

# Rule 13(global)
#
$IPTABLES -N RULE_13
$IPTABLES -A FORWARD -p tcp  -s 200.167.129.16/28 --source-
port 1024:65535  -d 200.18.223.0/24 --destination-port
81:82 -m state --state NEW -j RULE_13
$IPTABLES -A FORWARD -p tcp  -s 200.167.129.16/28 --source-
port 1024:65535  -d 200.18.223.0/24 --destination-port 8000
-m state --state NEW -j RULE_13
$IPTABLES -A RULE_13 -j LOG  --log-level 6 --log-prefix
"RULE 13 -- ACCEPT "  --log-tcp-sequence  --log-tcp-options
--log-ip-options
$IPTABLES -A RULE_13 -j ACCEPT

# Rule 14(global)
#
$IPTABLES -N RULE_14
```

```
$IPTABLES -A FORWARD -p tcp  -s 200.167.129.16/28 --source-
port 1024:65535  -d 161.148.0.0/16 --destination-port 8999
-m state --state NEW -j RULE_14
$IPTABLES -A FORWARD -p tcp  -s 200.167.129.16/28 --source-
port 1024:65535  -d 161.148.0.0/16 --destination-port 23000
-m state --state NEW -j RULE_14
$IPTABLES -A FORWARD -p tcp  -s 200.167.129.16/28 --source-
port 1024:65535  -d 161.148.0.0/16 --destination-port 3456
-m state --state NEW -j RULE_14
$IPTABLES -A RULE_14 -j LOG  --log-level 6 --log-prefix
"RULE 14 -- ACCEPT "  --log-tcp-sequence  --log-tcp-options
--log-ip-options
$IPTABLES -A RULE_14 -j ACCEPT

# Rule 15(global)
#
$IPTABLES -N RULE_15
$IPTABLES -A FORWARD -p udp  -s 200.167.129.16/28 --source-
port 1024:65535  -d 200.167.129.32/28 --destination-port
161 -m state --state NEW -j RULE_15
$IPTABLES -A FORWARD -p udp  -s 200.167.129.16/28 --source-
port 1024:65535  -d 200.167.129.32/28 --destination-port
162 -m state --state NEW -j RULE_15
$IPTABLES -A RULE_15 -j LOG  --log-level 6 --log-prefix
"RULE 15 -- ACCEPT "  --log-tcp-sequence  --log-tcp-options
--log-ip-options
$IPTABLES -A RULE_15 -j ACCEPT

# Rule 16(global)
#
$IPTABLES -N RULE_16
$IPTABLES -A FORWARD -p tcp  -s 200.167.129.16/28 --source-
port 1024:65535  -d 64.191.82.5 --destination-port 2082 -m
state --state NEW -j RULE_16
$IPTABLES -A RULE_16 -j LOG  --log-level 4 --log-prefix
"GREMIO"  --log-tcp-sequence  --log-tcp-options  --log-ip-
options
$IPTABLES -A RULE_16 -j ACCEPT

# Rule 17(global)
#
$IPTABLES -N RULE_17
$IPTABLES -A OUTPUT -m psd --psd-weight-threshold 5 --psd-
delay-threshold 1000 -j RULE_17
$IPTABLES -A INPUT -m psd --psd-weight-threshold 5 --psd-
delay-threshold 1000 -j RULE_17
$IPTABLES -A FORWARD -m psd --psd-weight-threshold 5 --psd-
```

```
delay-threshold 1000 -j RULE_17
$IPTABLES -A RULE_17 -j LOG  --log-level 2 --log-prefix
"PORT_SCAN_DETECTADO"  --log-tcp-sequence  --log-tcp-
options  --log-ip-options
$IPTABLES -A RULE_17 -j DROP

# Rule 18(global)
#
$IPTABLES -N RULE_18
$IPTABLES -A OUTPUT -p ip -f  -j RULE_18
$IPTABLES -A INPUT -p ip -f  -j RULE_18
$IPTABLES -A FORWARD -p ip -f  -j RULE_18
$IPTABLES -A RULE_18 -j LOG  --log-level 4 --log-prefix
"FRAGMENTO_IP"  --log-tcp-sequence  --log-tcp-options  --
log-ip-options
$IPTABLES -A RULE_18 -j DROP

# Rule 19(global)
#
$IPTABLES -N RULE_19
$IPTABLES -A INPUT -p tcp  -s 200.222.17.0/24 --source-port
1024:65535  -d 200.167.129.34 --destination-port 22 -m
state --state NEW -j RULE_19
$IPTABLES -A INPUT -p tcp  -s 200.20.228.0/24 --source-port
1024:65535  -d 200.167.129.34 --destination-port 22 -m
state --state NEW -j RULE_19
$IPTABLES -A INPUT -p tcp  -s 200.167.209.0/24 --source-
port 1024:65535  -d 200.167.129.34 --destination-port 22 -m
state --state NEW -j RULE_19
$IPTABLES -A INPUT -p tcp  -s 200.167.129.32/28 --source-
port 1024:65535  -d 200.167.129.34 --destination-port 22 -m
state --state NEW -j RULE_19
$IPTABLES -A INPUT -p tcp  -s 200.167.129.16/28 --source-
port 1024:65535  -d 200.167.129.34 --destination-port 22 -m
state --state NEW -j RULE_19
$IPTABLES -A RULE_19 -j LOG  --log-level 6 --log-prefix
"SSH-"  --log-tcp-sequence  --log-tcp-options  --log-ip-
options
$IPTABLES -A RULE_19 -j ACCEPT

# Rule 20(global)
#
$IPTABLES -N RULE_20
$IPTABLES -A OUTPUT -p icmp  -d 200.167.129.34 --icmp-type
0/0  -m state --state NEW -j RULE_20
$IPTABLES -A INPUT -p icmp  -d 200.167.129.34 --icmp-type
0/0  -m state --state NEW -j RULE_20
```

```
$IPTABLES -A RULE_20 -j LOG  --log-level 6 --log-prefix
"ICMP-"  --log-tcp-sequence  --log-tcp-options  --log-ip-
options
$IPTABLES -A RULE_20 -j ACCEPT

# Rule 21(global)
#
$IPTABLES -N RULE_21
$IPTABLES -A INPUT -p icmp  -s 200.167.129.34 --icmp-type
8/0  -m state --state NEW -j RULE_21
$IPTABLES -A OUTPUT -p icmp  -s 200.167.129.34 --icmp-type
8/0  -m state --state NEW -j RULE_21
$IPTABLES -A RULE_21 -j LOG  --log-level 6 --log-prefix
"ICMP-"  --log-tcp-sequence  --log-tcp-options  --log-ip-
options
$IPTABLES -A RULE_21 -j ACCEPT

# Rule 22(global)
#
$IPTABLES -N RULE_22
$IPTABLES -A INPUT -p tcp  -s 200.167.129.20 --source-port
1024:65535  -d 200.167.129.17 --destination-port 2049 -m
state --state NEW -j RULE_22
$IPTABLES -A INPUT -p tcp  -s 200.167.129.20 --source-port
1024:65535  -d 200.167.129.17 --destination-port 111 -m
state --state NEW -j RULE_22
$IPTABLES -A INPUT -p udp  -s 200.167.129.20 --source-port
1024:65535  -d 200.167.129.17 --destination-port 2049 -m
state --state NEW -j RULE_22
$IPTABLES -A INPUT -p udp  -s 200.167.129.20 --source-port
1024:65535  -d 200.167.129.17 --destination-port 111 -m
state --state NEW -j RULE_22
$IPTABLES -A RULE_22 -j LOG  --log-level 6 --log-prefix
"NFS-"  --log-tcp-sequence  --log-tcp-options  --log-ip-
options
$IPTABLES -A RULE_22 -j ACCEPT

# Rule 23(global)
#
$IPTABLES -N RULE_23
$IPTABLES -A OUTPUT -j RULE_23
$IPTABLES -A INPUT -j RULE_23
$IPTABLES -A FORWARD -j RULE_23
$IPTABLES -A RULE_23 -j LOG  --log-level 6 --log-prefix
"DROP-"  --log-tcp-sequence  --log-tcp-options  --log-ip-
options
$IPTABLES -A RULE_23 -j DROP
```

```
#
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Anexo E

ESTATÍSTICA DE VÍRUS NO EMAIL - CEFET

Virus Scan for Linux v4.16.0
Copyright (c) 1992-2001 Networks Associates Technology Inc. All rights reserved.
(408) 988-3832 LICENSED COPY - Nov 13 2001

Scan engine v4.1.60 for Linux.
Virus data file v4311 created Dec 24 2003
Scanning for 83913 viruses, trojans and variants.

Total de e-mail com vírus 5687

Quantidade e nome dos vírus

2 JS/Fortnight.gen@M
19 JS/Fortnight@M
1 Keylog-Spider.dr
1 VBS/Haptime.gen@MM
326 W32/Bugbear.b.dam
81 W32/Bugbear.b@MM
1 W32/Dumaru.a@MM
1 W32/Gibe.gen@MM
24 W32/Hybris.gen@MM
6 W32/Magistr.b@MM
132 W32/Mimail@MM
14 W32/Sobig.a@MM
11 W32/Sobig.dam
3 W32/Sobig.e@MM
4329 W32/Sobig.f@MM
258 W32/Swen@MM
1 W32/Valla.b
1 W32/Yaha.g@MM
7 W97M/Generic
1 W97M/Marker.gen
1 test
1 called

Quantidade de vírus por dia em e-mail encontrados no ano de 2003

54 Jan2	4 Jan5	4 Jan6	18 Jan9	6 Jan10
2 Jan11	3 Jan12	16 Jan13	5 Jan14	2 Jan15
1 Jan16	5 Jan17	3 Jan18	3 Jan19	1 Jan20
3 Jan22	9 Jan23	11 Jan24	7 Jan27	5 Jan28
15 Jan29	2 Jan30	1 Jan31	12 Fev2	2 Fev3
3 Fev4	6 Fev5	2 Fev6	5 Fev7	2 Fev8
4 Fev10	2 Fev11	3 Fev12	2 Fev13	5 Fev14
3 Fev16	2 Fev17	3 Fev19	2 Fev20	1 Fev22
1 Fev25	5 Fev26	1 Fev27	1 Fev28	1 Mar10

4 Mar12 7 Mar13 1 Mar14 1 Mar17 1 Mar27
2 Mar31 10 Abr1 14 Abr2 1 Abr4 3 Abr11
1 Abr12 1 Abr15 1 Abr16 1 Abr17 6 Abr23
1 Abr25 3 Abr26 1 Abr28 5 Mai3 1 Mai6
1 Mai7 1 Mai10 1 Mai11 3 Mai14 29 Mai15
2 Mai22 1 Mai23 2 Mai25 2 Mai26 2 Mai28
1 Mai29 3 Mai30 1 Mai31 213 Jun1 94 Jun2
2 Jun4 3 Jun5 56 Jun6 4 Jun8 12 Jun9
9 Jun10 7 Jun11 71 Jun12 50 Jun13 1 Jun14
60 Jun15 3 Jun16 6 Jun17 3 Jun18 2 Jun19
4 Jun20 4 Jun21 1 Jun22 50 Jun24 2 Jun25
1 Jun26 2 Jun27 26 Jun28 1 Jun29 37 Jul1
28 Jul2 5 Jul3 5 Jul4 63 Jul5 1 Jul6
1 Jul7 5 Jul8 2 Jul9 1 Jul11 1 Jul13
7 Jul14 9 Jul15 5 Jul16 8 Jul17 3 Jul18
1 Jul19 4 Jul21 8 Jul22 2 Jul23 2 Jul24
3 Jul25 1 Jul26 1 Jul28 1 Jul29 1 Jul31
2326 Ago2 3 Ago4 111 Ago5 10 Ago6 12 Ago7
2 Ago9 1 Ago11 2 Ago13 15 Ago14 3 Ago15
3 Ago19 143 Ago20 175 Ago21 230 Ago22 70 Ago23
24 Ago24 78 Ago25 195 Ago26 813 Ago27 377 Ago28
220 Ago29 58 Ago30 10 Ago31 467 Set1 378 Set2
264 Set3 350 Set4 300 Set5 4 Set6 6 Set7
79 Set8 215 Set9 2 Set10 4 Set17 13 Set19
10 Set20 10 Set21 9 Set22 10 Set23 26 Set24
16 Set25 7 Set26 9 Set27 4 Set28 3 Set29
10 Set30 104 Out1 32 Out2 3 Out3 1 Out4
1 Out5 8 Out6 3 Out7 18 Out9 28 Out10
7 Out11 9 Out12 35 Out13 7 Out14 1 Out15
1 Out16 1 Out17 1 Out18 10 Out19 5 Out20
3 Out21 2 Out22 3 Out23 1 Out24 12 Out25
1 Out26 4 Out28 1 Out30 1 Out31 66 Nov1
8 Nov2 3 Nov6 1 Nov8 3 Nov11 2 Nov19
3 Nov25 1 Nov26 2 Nov27 9 Dez1 2 Dez7
1 Dez13 2 Dez14 2 Dez15 1 Dez19