KARISTON PEREIRA

UMA PROPOSTA DE METODOLOGIA PARA O CONTROLE E DEFESA CONTRA VÍRUS E OUTROS *MALWARES* EM AMBIENTES CORPORATIVOS

FLORIANÓPOLIS – SC

2001

UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

KARISTON PEREIRA

UMA PROPOSTA DE METODOLOGIA PARA O CONTROLE E DEFESA CONTRA VÍRUS E OUTROS *MALWARES* EM AMBIENTES CORPORATIVOS

Dissertação submetida à Universidade Federal de Santa Catarina como parte dos requisitos para a obtenção do grau de Mestre em Ciência da Computação.

VITÓRIO BRUNO MAZZOLA

Florianópolis, Dezembro de 2001

UMA PROPOSTA DE METODOLOGIA PARA O CONTROLE E DEFESA CONTRA VÍRUS E OUTROS *MALWARES* EM AMBIENTES CORPORATIVOS

Kariston Pereira

Esta Dissertação foi julgada adequada para a obtenção do título de Mestre em Ciência da Computação na Área de Concentração Sistemas de Computação e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.

Banca Examinadora

Professor Fernando Alvaro O. Gauthier, Dr.

Professor Vitório Bruno Mazzola, Dr. (Orientador)

Professor Roberto Willrich, Dr.

Professor Murilo Silva de Camargo, Dr.

Professor João Bosco da Mota Alves, Dr.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

RESUMO

Este trabalho de Dissertação de Mestrado, desenvolve e analisa conteúdos relacionados aos Vírus de Computador e outros *softwares*, classificados como maliciosos ou mal intencionados, geralmente chamados de *Malware*, a partir de uma perspectiva histórica, e os meios utilizados para a prevenção e contenção dessas ameaças. O foco da dissertação é o estudo de metodologias alternativas na implantação e controle de um sistema eficaz de segurança, introduzindo diferentes abordagens no contexto da Segurança em Redes de Computadores, gerando sugestões de modelos para a implantação de soluções completas de defesa, envolvendo diversos aspectos relevantes no sistema e processo como um todo.

ABSTRACT

This Master's degree Dissertation, develops and analyses contents related to Computer Viruses and other softwares, classified as malicious or bad-intentioned, often called Malware, from a historical perspective, and the means used for the prevention and containment of these threats. The focus of the Dissertation is the study of alternative metodologies in the implementation and control of a efective security system, introducing different approaches into the context of Computer Network Security, generating suggestions of models for the implementation of defense complete solutions, involving several important aspects for the system and the process as a whole.

INTRODUÇÃO

Esta Dissertação de Mestrado, visa desenvolver conteúdos pertinentes ao estudo e análise da problemática dos Vírus de Computador e outros *softwares* mal intencionados aqui denominados genericamente de *Malware* (*Malicious Software*), introduzindo-os no contexto da Segurança em Redes Computacionais, estudando sua evolução, seus princípios de funcionamento e as tecnologias empregadas na sua defesa e controle. Ao final, são apresentados abordagens alternativas e como contribuição a sugestão de um novo modelo a ser adotado em ambientes corporativos.

Entre as principais motivações para o desenvolvimento deste trabalho, destaca-se a necessidade de geração de uma metodologia mais abrangente e eficaz, em complemento a metodologias já existentes e praticadas em ambientes corporativos, na defesa e controle contra os Vírus Digitais e demais *Malwares*, contemplando diversos aspectos considerados importantes para o sucesso de tal metodologia. Contribui nesse sentido, a experiência profissional adquirida ao longo de mais de dois anos de implantação de soluções tecnológicas existentes no mercado, bem como modelos de aplicação para essas tecnologias, e a constatação de diversas vantagens e desvantagens de tais modelos.

Assim sendo, o principal objetivo desta Dissertação de Mestrado, é, além de expor os problemas relacionados ao tema e apresentar as pesquisas desenvolvidas ao longo de sua confecção, o de apresentar uma proposta de solução em forma de uma Metodologia, e buscar um melhor emprego de soluções tecnológicas e não tecnológicas já existentes. Norteando-se por este objetivo, o documento foi dividido em sete capítulos, a saber:

1 - Introdução à Segurança em Redes de Computadores, onde é dada uma rápida apresentação da evolução das redes de computadores no que diz respeito à Segurança, seguida da apresentação de conceitos relacionados, contemplando princípios, ameaças, mecanismos e serviços de segurança em geral. A idéia é gerar uma fundamentação conceitual básica e desenvolver um contexto para propiciar o desenvolvimento dos

demais capítulos, envolvendo especificamente Vírus de Computador e ameaças relacionadas.

2 - História do Vírus de Computador, apresentando a história e evolução do Vírus de Computador e demais *Malwares*, desde a concepção do primeiro Vírus conhecido, até a situação em que nos encontramos atualmente.

3 - Conceitos e Fundamentos Teóricos, apresentando os conceitos relacionados aos Vírus de Computador, *Worms*, Cavalos de Tróia, etc., bem como a fundamentação teórica básica das tecnologias envolvidas.

4 - Como Funciona o Processo de Reprodução e Disseminação de um Vírus de Computador, expondo os princípios de funcionamento dos Vírus de Computador, abrangendo mecanismos de disseminação e reprodução.

5 - Tecnologias e Metodologias de Defesa, capítulo que apresenta as tecnologias e estratégias utilizadas em ferramentas e metodologias que objetivam defender sistemas informatizados contra as ameaças de *software*, geralmente classificadas como Vírus de Computador. Este capítulo é uma espécie de transição entre os estudos desenvolvidos na fase de revisão bibliográfica, apresentados nos capítulos anteriores e a contribuição da Dissertação propriamente dita, visto que desenvolve o assunto de uma forma analítica nas diversas considerações de cunho dissertativo.

6 – Vírus em Ambientes Unix/Linux, parte do documento que discute a crescente problemática da ocorrência de Vírus e *Malware* em ambientes Unix e de uma forma especial, em ambientes Linux, cada vez mais populares seja no uso comercial ou mesmo doméstico. O conteúdo é desenvolvido expondo-se a realidade da situação encontrada nesses ambientes, desmistificando supostos tabus e apresentando sucintamente as diversas possibilidades de danos que podem ser gerados.

7 – Modelo das Tríades Conjugadas, capítulo final da Dissertação de Mestrado, apresentando uma sugestão de Metodologia a ser empregada em sistemas de defesas, fazendo o uso de um modelo amplo e de aplicação genérica, passível de ser implantado nos mais diversos ambientes, sob as mais diversas tecnologias de defesa, discutindo-se sua aplicabilidade e gerando algumas abordagens no tocante à sua implantação.

Para finalizar o documento, são apresentados um capítulo com as conclusões finais e outro com as referências bibliográficas das obras citadas neste trabalho.

1. INTRODUÇÃO À SEGURANÇA EM REDES DE COMPUTADORES

1.1. Introdução

No princípio, o computador era um equipamento isolado, sem qualquer tipo de comunicação com outros computadores. A crescente necessidade de compartilhamento de recursos e troca de informações levou ao desenvolvimento de tecnologias de comunicação e interconexão de sistemas computacionais, gerando a fusão dos computadores e das comunicações, o que passou a influenciar, significativamente, no modo como os computadores eram organizados. O modelo de um computador atendendo a todas as necessidades de uma organização foi substituído pelas chamadas redes de computadores, nas quais os trabalhos são realizados por uma série de computadores interconectados.

As redes de computadores facilitaram a troca de informações e possibilitaram o compartilhamento de recursos, colocando todos os programas, equipamentos e dados ao alcance de todas as pessoas da rede, em detrimento da localização física dos recursos e dos usuários. A rede também aumentou a confiabilidade do sistema, gerando fontes alternativas de fornecimento de recursos e dados. A rede propiciou ainda, a baixa nos custos e conseqüente economia, estabelecendo uma melhor relação preço/desempenho, ao passo que oferece a vantagem da escalabilidade, permitindo o aumento gradual no desempenho do sistema à medida que cresce o volume de carga.

Desde a primeira implantação dos sistemas computacionais em rede, a grande preocupação sempre foi a funcionalidade e o desempenho do sistema como um todo, procurando-se desenvolver sistemas de comunicação e interconexão leves e com um mínimo de controle possível, além de conexões eficientes, de forma que as redes de computadores se tornassem de uso prático e passassem a ser a base para a computação moderna.

A segurança das comunicações via rede de computadores, não foi alvo de estudos aprofundados em sua concepção. Na verdade a preocupação com a segurança surgiu apenas anos mais tarde. No início, as redes eram utilizadas por cientistas em suas pesquisas e havia fortes laços de confiança dentro da comunidade científica na sua

utilização, com o nobre objetivo da rápida difusão da informação e troca de experiências entre os Centros de Pesquisa e Universidades envolvidas. No meio comercial, a rede surgiu principalmente como um meio de compartilhamento de recursos. [TANEMBAUM, 1997, p.657-658], escreveu o seguinte:

> "Durante as primeiras décadas de sua existência, as redes de computadores foram principalmente usadas por pesquisadores universitários, para enviar mensagens de correio eletrônico, e por funcionários de empresas, para compartilhar impressoras. Sob essas condições, a segurança nunca precisou de maiores cuidados. Mas atualmente, como milhões de cidadãos comuns estão usando as redes para executar operações bancárias, fazer compras e arquivar suas devoluções de impostos, a segurança das redes está despontando no horizonte como um problema em potencial."

O surgimento e desenvolvimento da Internet não fugiu à regra. A Internet teve como precursora a ARPANET, que era uma rede de pesquisa criada pelo departamento de defesa dos Estados Unidos. Com o tempo, inúmeras universidades e órgãos do governo norte americano foram sendo conectados à ARPANET através de linhas privadas. Com o surgimento das redes de rádio e satélite, os protocolos existentes começaram a provocar problemas de forma que se fez necessário a criação de uma arquitetura de referência com o objetivo de conectar várias redes ao mesmo tempo. Assim surgiu o Modelo de Referência TCP/IP, base da Internet. Segundo [TANEMBAUM, 1997, p.39], esse modelo foi definido pela primeira vez por Cerf e Kahn, em 1974. A idéia da Internet, foi de criar um mecanismo de interligação de redes que fosse flexível o suficiente para que no caso de uma determinada conexão estar com problema, uma conexão alternativa seja acionada e a comunicação não seja interrompida. No princípio, o projeto era de desenvolver uma rede de comunicação de longa distância imune a ataques nucleares. Nesse sentido, [TANEMBAUM, 1997, p.40] faz a seguinte colocação:

> "Diante da preocupação do Pentágono de que seus preciosos hosts, roteadores e *gateways* de inter-rede fossem destruídos de uma hora para outra, definiu-se também que a rede fosse capaz de sobreviver à perda de hardware da sub-rede, impedindo que as conversas que estivessem sendo travadas fossem interrompidas. Em outras palavras, o Pentágono queria que as conexões permanecessem intactas enquanto as máquinas de origem e de destino estivessem funcionando, mesmo que algumas máquinas ou linhas de transmissão intermediárias deixassem de operar repentinamente. Por essa razão, era preciso criar uma arquitetura flexível, capaz de se adaptar a aplicações com necessidades

> divergentes, como por exemplo a transferência de arquivos e a transmissão de dados de voz em tempo real."

Percebe-se que a segurança, também no caso da Internet, deixou de ser alvo em sua concepção. No caso especial da Internet, que permite a interligação de redes dos mais diferentes tipos de tecnologia, distribuídas por todo o mundo, a situação se torna extremamente delicada, sendo que a necessidade de segurança foi sendo impulsionada gradativamente na medida que a Internet foi se tornando algo imprescindível para, primeiramente, a iniciativa militar e acadêmica, e em seguida na sua utilização como mecanismo de realização de negócios nos mais diversos segmentos, sendo que hoje a Internet já figura como uma das maiores revoluções tecnológicas que o homem presenciou.

Na segunda metade da década de oitenta, com o surgimento e rápida difusão dos chamados Vírus de computador, a segurança dos sistemas passou a despontar como uma preocupação de qualquer entidade que utilizasse a informática em seus negócios. A partir de então, intensificou-se a geração de esforços no desenvolvimento de técnicas, mecanismos, políticas e serviços voltados à segurança computacional, seja como medida de contenção, contingência, defesa ou mesmo no desenvolvimento de sistemas e redes seguros.

Hoje, com a explosão das redes, consagrada pelo sucesso da Internet, a Segurança é considerada indispensável, figurando como uma das maiores preocupações de quem se utiliza de sistemas computacionais, principalmente em ambientes distribuídos.

1.2. Princípios da Segurança

[KATZAN, 1977, p.1] afirma que um dos problemas mais árduos que a indústria de computadores enfrenta é o da segurança de dados. O problema tem diversas facetas e envolve instalações físicas, procedimentos operacionais, características de *hardware* e convenções de *software* bem como de programação.

A segurança de redes tem por objetivo a preservação do patrimônio da empresa (os dados e as informações fazem parte do patrimônio), manutenção dos serviços prestados e segurança do corpo funcional.

Em caso de problemas, deve-se fazer a detecção das causas e origens dos problemas, minimização das conseqüências dos mesmos, retorno às condições normais no menor prazo, com o menor custo e com o menor trauma possíveis. Os meios para alcançar estes objetivos são bastante claros. Eles são baseados na detecção e análise dos pontos vulneráveis, estabelecimento de políticas de segurança, execução das políticas de segurança, acompanhamento, avaliação dos resultados contra os objetivos traçados, correção de objetivos e políticas. Segundo [SOARES, 1995, p.448], "A necessidade de proteção deve ser definida em termos das possíveis ameaças e riscos e dos objetivos de uma organização, formalizados nos termos de uma política de segurança". Basicamente, deve ser criado um Plano de Segurança (como evitar problemas) e um Plano de Contingência (o que fazer em caso de problemas).

É oportuno frisar que segurança absoluta não existe - ninguém é imune a ataques nucleares, colisões com cometas ou asteróides, epidemias mortais, seqüestros, guerras, etc. Trata-se de descobrir os pontos vulneráveis, avaliar os riscos, tomar as providências adequadas e investir o necessário para ter uma segurança homogênea e suficiente. O axioma da segurança é bastante conhecido de todos, mas é verdadeiro: "Uma corrente não é mais forte do que o seu elo mais fraco".

1.2.1. Ameaças, Vulnerabilidades, Serviços e Mecanismos

Uma ameaça pode ser qualquer pessoa, objeto, ou evento que, se realizado poderia potencialmente causar danos à rede. Ameaças podem ser maliciosas tais como modificações intencionais a informações críticas, ou podem ser acidentais, tais como um erro de cálculo, ou a exclusão acidental de um arquivo. Ameaças também podem ser ações da natureza. O dano imediato causado por uma ameaça pode ser referenciado por seu impacto.

[SOARES, 1995, p.448] classifica as ameaças como passivas ou ativas. As ameaças passivas não resultam em qualquer modificação, operação ou estado das informações contidas em um sistema. Já as ativas envolvem alteração da informação, ou modificação de seu estado e operação.

Vulnerabilidades são fraquezas em uma rede que podem ser exploradas por uma ameaça. Por exemplo, acesso não autorizado à rede poderia ocorrer por um acesso

externo utilizando uma senha óbvia (de descoberta fácil), sendo que a vulnerabilidade explorada, nesse caso, é a escolha de uma senha pobre feita pelo usuário. Reduzindo ou eliminando as vulnerabilidades da rede pode-se reduzir ou eliminar as chances de riscos de ameaças para a rede. Como exemplo pode-se considerar uma ferramenta que possibilite o auxílio, ou evite o usuário de escolher senhas pobres, para criar senhas mais robustas que inibam e dificultem os acessos não autorizados. As vulnerabilidades abrem brechas para ameaças que podem acarretar em destruição de informação ou outros recursos, modificação, deturpação, roubo, remoção ou revelação de informação e interrupção de serviços. A materialização de uma ameaça intencional configura um ataque [SOARES, 1995, p.448, 449].

Um serviço de segurança pode ser visto como um conjunto de mecanismos de segurança, suportando arquivos de dados, e procedimentos que ajudem a proteger a rede de ameaças específicas. Por exemplo, a identificação e autenticidade de serviço ajudam a proteger a rede de acessos não autorizados requerendo que o usuário identifique-se, assim como verificar sua identidade.

Conforme [FIPS 191, 1994], mecanismos de segurança são controles implementados para prover a segurança necessária a serviços na rede. Por exemplo, um sistema de autenticação baseado em *token* (que exige que o usuário posua o *token* requerido) pode ser implementado em um mecanismo que realize a identificação e autenticação do serviço. Outros mecanismos que podem ajudar a manter a confidencialidade e a autenticidade da informação podem ser considerados como parte do serviço de identificação e autenticação.

1.2.2. Políticas de Segurança

Um plano de segurança computacional abrangente envolve as seguintes informações [KATZAN, 1977, p. 26]:

1. Necessidades: uma análise completa das necessidades de segurança computacional da organização;
2. Descrição: uma inspeção e descrição dos procedimentos e medidas de segurança que estão normalmente disponíveis;

3. Deficiência (Vulnerabilidade): áreas onde as facilidades existentes não satisfazem as necessidades;
4. Avaliação: os custos, exigências técnicas e alterações organizacionais necessárias para a implementação das medidas de segurança necessárias;
5. Implementação: especificação de um plano e um horário para a implementação de um plano de segurança;

"Uma política de segurança é um conjunto de leis, regras e práticas que regulam como uma organização gerencia, protege e distribui suas informações e recursos" [SOARES,1995, p. 450].

Segundo [TANENBAUM,1997, p. 659], os problemas de segurança em redes de computadores podem ser divididos em algumas áreas interligadas, tais como sigilo, autenticação, não repudiação e controle de integridade. O sigilo diz respeito ao fato de se isolar as informações de usuários não autorizados. A autenticação resume-se no processo de identificação do receptor antes de se enviar qualquer mensagem de caráter sigiloso, ou entrar em uma transação comercial. A não repudiação trata da questão das assinaturas digitais. O controle de integridade é para garantir que a mensagem enviada seja recebida sem adulterações, cuidando do aspecto legitimidade. Paralelo a esse contexto, poderíamos adicionar a segurança de sistemas, que trata do controle de segurança em Sistemas Operacionais e Aplicações, destacando-se os Vírus de computador e outros *Malwares* (*Malicious Software*) como umas das principais ameaças.

[SOARES, 1995, p. 450] afirma que a política de segurança "define o que é, e o que não é permitido em termos de segurança, durante a operação de um dado sistema. A base da política de segurança é a definição do comportamento autorizado para os indivíduos que interagem com um sistema".

O conjunto de regras que definem uma política pode ser de dois tipos (com base na natureza da autorização envolvida)[SOARES, 1995, p. 450,451]:

1. Regras baseadas em atributos de sensibilidade genéricos, classificando por exemplo, entidades em graus de segurança como secreto, ultra-secreto, etc. A política definida por esse tipo de regras, é conhecida como Política de Segurança Baseada em Regras, onde os dados ou recursos devem ser marcados com rótulos de segurança que indiquem seu nível de sensibilidade;
2. Regras baseadas em atributos individuais específicos que fundamentam a Política de Segurança Baseada em Identidade, que representa o tipo de controle mais comum nos computadores atuais, onde se especifica tipos de acesso às informações e recursos solicitados;

1.3. Segurança na Internet

Segundo[BERNSTEIN, 1997, p. 26], de acordo com diversos estudos, de 80 a 95% do número total de incidentes de segurança acontecem nas redes internas, restando à Internet, uma pequena faixa no contexto das ameaças. O nível de segurança da rede interna é primordial. Contudo existe sempre a possibilidade de acontecimento de incidentes provocados por acessos externos, na grande maioria das vezes através da Internet. O que precisa existir é uma consciência da necessidade de um programa de segurança global efetivo para garantir uma proteção adequada.

1.3.1. Ameaças e Vulnerabilidades

As propriedades intrínsecas da Internet representam a principal fonte de suas vulnerabilidades/falhas e ataques. Diversos problemas podem ocorrer na enorme teia de redes interconectadas através da Internet, tais como roteamentos incorretos, falhas de transmissão, adulteração de dados, falhas nos equipamentos, etc.

> "A Internet é uma rede pública, com comutação de pacotes; portanto, os pacotes trafegam por rotas que não são tão bem controladas quanto nas redes privadas: pode haver perda de informações; talvez os sistemas não estejam sempre

> funcionando do modo adequado; pessoas ocultas podem conduzir atos de espionagem ou ataques de falsificação ideológica a partir de locais remotos. Protocolos, que definem as regras para o controle de como as mensagens são trocadas em uma rede de computadores, podem apresentar defeitos ou parar de funcionar. Todos esses fatores contribuem para que a Internet seja um meio de comunicação não-confiável e inseguro". [BERNSTEIN, 1997, p. 27].

Entre os principais tipos/classes de ameaças que surgem da interligação de redes com o estabelecimento de conexão com a Internet são [BERNSTEIN, 1997, p. 28-30]:

- Ameaças à Rede Corporativa: furos de segurança propiciam o acesso a intrusos a componentes da rede;
- Ameaças ao Servidor da Internet: intrusos podem entrar no servidor da Internet (WWW, FTP, etc.) e consultar ou até mesmo alterar arquivos armazenados;
- Espionagem: a identidade de um ou mais usuários envolvidos em algum tipo de comunicação é observada para ser mal utilizada posteriormente. Informações confidenciais são observadas durante sua transmissão através da rede;
- Disfarce: um usuário finge ser outro para se aproveitar de privilégios alheios;
- *Replay*: uma seqüência de eventos ou comandos é observada e reproduzida posteriormente para que possa efetivar alguma ação não autorizada;
- Manipulação de dados: a integridade dos dados é danificada durante o armazenamento ou durante a transmissão sem que isso seja detectado;
- Roteamento Incorreto: uma comunicação para o usuário A é roteada para o usuário B, que pode levar a uma interceptação de mensagem. Os roteamentos incorretos podem ser usados em conjunto com disfarces, manipulações e *replays*;
- Armadilha(Cavalo de Tróia): um processo não autorizado pode executar um programa como se fosse um processo autorizado; um determinado aplicativo ou programa de sistema é substituído por outro que contém uma seção adicional alterada, permitindo algum tipo de atividade mal intencionada não detectada;

- Vírus: os Vírus de computador são códigos de programa que se auto-reproduzem. Eles se associam a um componente de um arquivo executável ou a um programa aplicativo de um sistema e posteriormente o modificam. Os Vírus podem alterar ou eliminar diversos arquivos de sistema, alterar dados ou negar disponibilidade;
- Repúdio: um ou mais usuários negam a participação em uma comunicação, uma ameaça crítica para transações financeiras eletrônicas e acordos contratuais eletrônicos;
- Negação de Serviço (*Denial of Service*): o acesso a um sistema/aplicação é interrompido ou impedido, o sistema ou aplicação deixa de estar disponível, ou uma aplicação cujo tempo de execução seja crítico é atrasada ou abortada;

De modo geral, as vulnerabilidades tecnológicas associadas à Internet, podem ser classificadas em duas categorias: as causadas por deficiências inerentes a mecanismos e produtos, e as resultantes de configurações incorretas de sistemas operacionais e de programas aplicativos.

[BERNSTEIN, 1997, p. 31-33] afirma que entre os pontos fracos inerentes, muitos problemas são gerados por problemas nos protocolos de comunicação, que definem as regras nas quais se baseia a interoperação das redes. Por exemplo, no caso do TCP/IP, o principal problema é a inabilidade para confirmar a identidade dos participantes em um processo de comunicação. Existem ainda, produtos com fraquezas de segurança inerentes, como no caso do UNIX, originalmente projetado para compartilhar informações sem qualquer restrição. Entre os pontos fracos na configuração podemos exemplificar: contas de usuários inseguras (logins livres ou contas expiradas); contas de sistema com senhas muito óbvias ou já conhecidas; serviços da Internet incorretamente configurados; parâmetros básicos inseguros nos produtos.

[BERNSTEIN, 1997, p. 33] defende que uma das maneiras de auxiliar o tratamento desses pontos fracos é utilizar recursos de segurança e auditoria nos sistemas para detectar os problemas assim que eles surgirem.

As políticas de segurança corporativa fornecem as bases para um bom programa de segurança. Em geral, os controles básicos abrangem aspectos como: controle de acesso físico, controles de acesso lógico, administração de segurança, monitoração e auditoria de segurança, gerenciamento de modificações em *software* e *hardware*, *backup* e recuperação de desastre, continuidade dos negócios. Faz-se também necessária, a criação de políticas e procedimentos específicos para as tecnologias, que no mínimo, deveriam levar as seguintes áreas em consideração, segundo [BERNSTEIN, 1997, p. 34,35]:

- Monitoração de segurança realizada regularmente para eventos registrados em *logs* de sistema e de auditoria;
- Coleta, disseminação e implementação de informações sobre aspectos de segurança vulneráveis na Internet e do que pode ser feito para corrigi-los;
- Verificação e implementação de *patchs* de fornecedor, recursos de segurança e dispositivos semelhantes;
- Integridades de arquivos de sistema, cuja importância é crítica;

Ainda de acordo com[BERNSTEIN, 1997, p. 35], as deficiências tecnológicas podem se originar de problemas com produtos, configurações incorretas desses produtos ou de políticas inadequadas de segurança. Assume importância equivalente a existência e imposição de políticas e procedimentos de segurança sólidos que aceitem esses padrões tecnológicos.

1.3.2. Mecanismos e Serviços de Segurança

São apresentados a seguir, conforme [BERNSTEIN, 1997, p. 56-58], os principais serviços de controle de segurança que devem ser incluídos no programa de segurança de uma empresa para conexões com a Internet. O sucesso desse programa, na verdade, pode ser medido por meio da eficiência e da efetividade com que esses controles forem incorporados à rede:

- Identificação e Autenticação: determinar a identidade de uma entidade (usuário, aplicação ou sistema) e confirmar se a entidade é quem ou o que afirma ser;
- Controle de Acesso: uma vez que uma entidade tenha sido identificada e autenticada, o ato de decidir quais serão as concessões, atribuições, autorizações, direitos ou permissões associadas à execução da tarefa solicitada; impedir o acesso físico não-autorizado a sistemas e redes;
- Integridade dos Dados: garantir que os dados não tenham sido modificados, acrescentados ou eliminados ao serem transportados ou armazenados e que estão completos;
- Confidencialidade dos Dados: garantir que a proteção e a não-revelação de dados devido a natureza (legal, reguladora, patenteada ou confidencial) do recurso de informação;
- Não Repudiação: fornecer a integridade e a origem dos dados em uma relação que não deve ser falsificada e que pode ser confirmada por terceiros a qualquer momento; já existem, inclusive, recursos de comprovação de entrega;
- Disponibilidade dos Dados: garantir que os dados estão presentes, acessíveis e podem ser obtidos rapidamente de acordo com pré-requisitos reguladores. Isso também implica a utilidade dos dados, pois eles devem estar prontos para serem usados;
- Recursos de Auditoria: garantir a existência de trilhas de auditoria adequadas que forneçam registros de atividade cuja função é atestar os serviços de segurança; as trilhas de auditoria também devem confirmar a integridade do mecanismo de auditoria propriamente dito.

Entre os principais mecanismos e tecnologias de segurança podemos citar [BERNSTEIN,1997, p. 140-173]: Segurança de Host (fundamenta-se na configuração de cada computador da rede para que ele seja resistente à intrusão), Criptografia e Assinatura Digital, e ainda Autenticação.

1.3.2.1. Sistemas de Identificação de Intrusões

Um dos tipos mais noticiados de ameaça à segurança é a intrusão (o outro são os Vírus), geralmente efetuada pelos chamados *hackers* e *crackers*. [ANDE, 1980], citado por [STALLINGS, 1999, p. 478], identifica três classes de intrusos:

- *Masquerader*: um indivíduo que não é autorizado a usar o computador e que penetra no sistema para explorá-lo com uma conta de usuário legítimo;
- *Misfeasor*: um usuário legítimo que tem a acesso a dados, programas, ou recursos para os quais o acesso não é autorizado;
- Usuário Clandestino: um indivíduo que toma conta do controle de supervisão do sistema para fugir de auditorias e controle de acesso;

De acordo com [STALLINGS, 1999, p. 479-480], o objetivo do intruso é conseguir acesso ao sistema ou aumentar o número de privilégios de acesso ao sistema. Geralmente, isso requer que o intruso adquira ou acesse informações que deveriam estar protegidas. Em grande parte dos casos essa informação está em forma de senhas de usuários. Com algumas senhas de usuários, um intruso poderia entrar no sistema e usufruir de todos os privilégios concedidos a um usuário legítimo.

Segundo [STALLINGS, 1999, p. 490-491], o melhor sistema de prevenção de intrusões, inevitavelmente, fracassará. Uma segunda linha de defesa é a detecção de intrusões, e isso tem sido o foco de muitas pesquisas recentemente. O interesse é motivado por um bom número de considerações, entre elas:

1. Se um intruso é detectado rapidamente, ele pode ser identificado e retirado do sistema antes que qualquer dano seja feito. Quanto mais cedo se detectar um intruso, menos riscos de ações prejudiciais o sistema estará correndo;
2. Um sistema de detecção de intrusões eficiente pode servir como um meio de desencorajar a ação de intrusos;
3. A detecção de intrusões habilita a aquisição de informações sobre técnicas de intrusão que podem ser usadas para reforçar a prevenção na ação de intrusos;

A detecção de intrusões é baseada na suposição de que o comportamento de um intruso difere do comportamento de um usuário legítimo de forma que possa ser

quantificado. Naturalmente, não se pode esperar que exista uma exata distinção entre um ataque e uma ação normal de um usuário. Mas usando-se de funções de densidade de probabilidade pode-se chegar a resultados satisfatórios.

1.3.2.2. Criptografia, Assinatura Digital, e Autenticação

"A criptografia surgiu da necessidade de se enviar informações sensíveis através de meios de comunicação não confiáveis, ou seja, em meios onde não é possível garantir que um intruso não irá interceptar o fluxo de dados para leitura ou modificação." [SOARES, 1995, p. 452]

De acordo com [BERNSTEIN, 1997, p. 148] a criptografia oferece proteção às ameaças de perda de confiabilidade, integridade ou não repudiação. A criptografia é quase tão antiga quanto à própria escrita, mas foi apenas após a Segunda Guerra Mundial que essa área obteve avanços consideráveis.

Segundo [TANENBAUM, 1997, 663-665], historicamente os métodos de criptografia têm sido separados em duas classes: as cifras de substituição e as cifras de transposição. Em uma cifra de substituição, cada letra ou grupo de letras é substituído por outra letra ou grupo de letras, obedecendo a uma determinada regra de formação do alfabeto usado para a substituição. As cifras de substituição disfarçam a ordem dos símbolos no texto simples, apesar de preservarem sua ordem. Em outra mão, as cifras de transposição reordenam as letras, mas não as disfarçam ou substituem, sendo que a transposição é feita baseando-se em uma chave formada por uma palavra pequena sem letras repetidas.

Embora a criptografia moderna utilize as mesmas idéias básicas de substituição e transposição, a ênfase mudou. A tendência atual é a criação de algoritmos complexos em vez da utilização de algoritmos simples e com chaves muito longas para sua segurança. [TANENBAUM, 1997, 669].

Os métodos de criptografia podem ainda ser classificados, conforme as chaves empregadas, em simétricos e assimétricos.

Conforme [BERNSTEIN, 1997, p. 149], a criptografia com chave simétrica utiliza a mesma chave para codificação e decodifição de mensagens. Esse método, conhecido também como criptografia tradicional, funciona bem em aplicações limitadas, onde o emissor e o receptor podem se preparar antecipadamente para trocar a

chave. Infelizmente, de uma forma geral esse método não funciona muito bem, uma vez que trocar chaves secretas com todas as pessoas a quem se queira enviar uma mensagem é praticamente impossível.

Ainda de acordo com [BERNSTEIN, 1997, p. 150,151], os algoritmos de chave simétrica mais comuns são o DES (*Data Encryption Standard*), DES Triplo, o RC2, o RC4, e o IDEA (*International Data Encryption Algorithm*).

De acordo com [TANENBAUM, 1997, p. 681], em 1976, dois pesquisadores da Universidade de Stanford, Diffie e Hellman, propuseram um sistema de criptografia inovador, no qual as chaves de codifição e decodifição eram diferentes e a chave de decodifição não podia ser derivada da chave de codificação.

[BERNSTEIN, 1997, p. 151-153] afirma que em um sistema de chave pública, cada pessoa tem duas chaves: uma chave pública e outra privada. As mensagens criptografadas com uma chave do par só podem ser descriptografadas com a outra chave correspondente; assim, qualquer mensagem codificada com a chave privada só pode ser decodificada pela chave pública e vice-versa. A chave pública normalmente é disponibilizada a todos; em outra mão, a chave privada deve ser mantida em segredo, para garantir a segurança do modelo.

[SOARES, 1995, p. 456] defende que o mais importante método de criptografia assimétrico é o RSA, cujo nome é inspirado em seus três desenvolvedores, Ron Rivest, Adi Shamir e Leonard Adleman.

A existência de duas chaves separadas proporciona um benefício adicional: a assinatura digital. "O mecanismo de assinatura digital envolve dois procedimentos: assinatura de uma unidade de dados e verificação da assinatura em uma unidade de dados. O primeiro procedimento baseia-se em informação privada (única e secreta) do signatário. O segundo utiliza informação pública para reconhecer a assinatura". [SOARES, 1995, p. 457].

Para exemplificar imaginemos um emissor enviando uma mensagem usando sua chave privada. Os destinatários recebem a mensagem e a descriptografam utilizando a chave pública do emissor, e podem estar certos que a mensagem foi enviada pelo emissor anunciado, desde que sua chave privada fora realmente mantida em segredo.

Consoante [BERNSTEIN,1997, p. 153], a assinatura digital implementa os objetivos de segurança da integridade e não repudiação. Ela assegura aos participantes

que a mensagem não foi alterada (integridade) e que veio realmente de quem diz ter enviado (autenticidade). Alem disso, o emissor não pode negar que tenha enviado a mensagem (não repudiação), pois é o único com acesso a sua chave privada.

Segundo [TANENBAUM, 1997, p. 685], "a autenticação é a técnica através da qual um processo confirma que seu parceiro na comunicação é quem deve ser, e não um impostor".

De acordo com [BERNSTEIN, 1997, p. 171-173], os serviços de autenticação são um elemento importante em qualquer sistema de Segurança na Internet. A aplicação e a natureza dos dados que estão sendo protegidos impõem o tipo de autenticação utilizado. Em geral, quanto mais tipos de autenticação forem utilizados, mais segura será a transação. As principais desvantagens dos sistemas que utilizam mais de uma técnica de autenticação são o custo mais alto e a dificuldade de administração. Entre os principais mecanismos de autenticação, podemos citar: ID Biométrico, *Callback*, ID da Chamada, Identificação de Nó, Senha Ocasional, PC *Card*, *Smart Card*, *SmartDisk*, Dispositivo de *Token*, Senha Tradicional.

1.3.2.3. *Firewall*

Um mecanismo muito usado na prática para aumentar a segurança de redes ligadas à Internet é o *firewall*, que é uma espécie de barreira de proteção. "Os *firewalls* são apenas uma adaptação moderna de uma antiga forma de segurança medieval: cavar uma fossa profunda em torno do castelo." [TANENBAUM, 1997, p.468]. [BERNSTEIN, 1997, p. 176] apresenta que um dos objetivos mais importantes de um *firewall* é reduzir os danos em caso de um desastre, de maneira semelhante ao que acontece quando um carro ou edifício pega fogo. Adaptando para o contexto da Internet, os *firewalls* têm por finalidade controlar os prejuízos e proteger uma rede, no caso de uma invasão através da Internet. E, num sentido mais amplo, o *firewall* é usado para regular o fluxo do tráfego entre duas redes.

> "Uma ferramenta extremamente avançada que oferece segurança na rede, o *firewall* representa uma eficiente estratégia para implementar a política de acesso à Internet em uma organização. (...) Os *firewalls* podem oferecer proteção contra ataques e protocolos ou aplicações individuais, protegem contra ataques

> de *spoofing* e tem relativa flexibilidade de configuração, ou seja, oferecem várias restrições para diferentes tipos de tráfego. Os *firewalls* implementam controles de acesso baseados nos conteúdos dos pacotes de uma conexão. A melhor maneira de entender como age um *firewall* é imaginá-lo como sendo um guarda, ou sentinela, da rede, que inspeciona a documentação para os pacotes que chegam e depois decide se dará passagem ou não". [BERNSTEIN, 1997, p. 176].

Os três principais tipos de *firewall*, segundo [STALLINGS, 1999, p.520-525], são o Filtro de Pacotes, *Gateway* de Aplicação(*Proxy Server)* e *Gateway* de Circuito. O primeiro (Filtro de Pacotes), aplica um conjunto de regras para cada pacote IP que chega, descartando-o ou encaminhando-o . O *Gateway* de Aplicação age como um intermediador/controlador de tráfego a nível de aplicação. O último (*Gateway* de Circuito) não permite conexões TCP fim a fim, funcionando como um intermediador a nível de conexão.

Contudo, ainda segundo [STALLINGS, 1999, p.519-520], os *Firewalls* apresentam algumas limitações, a saber:

- O *Firewall* não pode proteger contra ataques que ultrapassem ou contorne o *Firewall*;
- O *Firewall* não protege contra ameaças internas, tais como um empregado frustrado ou que inconscientemente coopere com um atacante externo;
- O *Firewall* não pode proteger contra transferência de programas ou arquivos infectados por Vírus. Devido à variedade de sistemas operacionais e aplicações suportadas dentro do perímetro, seria impraticável e talvez impossível de o *Firewall* varrer todos os arquivos e mensagens à procura de Vírus;

2. HISTÓRIA DO VÍRUS DE COMPUTADOR

2.1. Como Tudo Começou

Segundo [LUNDELL, 1989, p.30-31], no final dos anos 60, surgiu um programa em linguagem *Assembly* denominado Creeper, o qual possuía apenas a característica de se reproduzir dentro de uma mesma rede. A sua disseminação foi muito rápida e contundente, ocupando espaço e degradando a *performance* do sistema. A primeira solução imaginada foi a de criar um segundo programa auto-replicável chamado Reaper, cuja única finalidade era procurar e destruir as cópias do Creeper. Reaper foi bem sucedido no seu intento, e quando conseguiu eliminar a última cópia do Creeper, ele se auto-destruiu.

Creeper, o primeiro vírus conhecido, foi detectado pela primeira vez em 1970. Ele foi desenvolvido por Bob Thomas do *Bulletin Board Network* (BBN), sendo criado inicialmente somente para demonstração mas ficou fora de controle. Ele se infiltrou na ARPANET (rede nacional do Pentágono), deixando, por onde quer que passasse, uma mensagem dizendo: "Sou o Creeper, se puderem , apanhem-me!".

Ao tomar conhecimento do caso Creeper/Reaper, o professor A. K. Dewdney, da Universidade do *Western* Ontário, teve a idéia de criar um novo jogo para computadores, o "Core War". Segundo divulgação feita em um artigo do professor Dewdney na revista Scientific American, em maio de 1984, o "Core War" é um jogo de dois programas que se espreitam um ao outro no seu próprio habitat, os circuitos integrados de memória de um computador digital [LUNDELL, 1989, p.33].

O primeiro torneio de "Core War" foi realizado em 1984 no histórico *Boston Computer Museum*. A idéia de programas procurarem destruírem-se uns aos outros tornou-se um passa-tempo de alta tecnologia. Muitas das técnicas desenvolvidas por divertimento ao jogar o "Core Ware" tornar-se-iam mais tarde os alicerces da nova tecnologia dos Vírus de Computador.

Enquanto isso, ainda segundo [LUNDELL, 1989, p.40], em 1983, o Dr. Frederick B. Cohen, na época doutorando na Universidade do Sul da Califórnia, realizava pesquisas no sentido de gerar um Vírus realmente reprodutivo. A idéia surgiu enquanto participava de uma aula lecionada pelo especialista em Criptografia Len

Adleman. O Dr. Cohen programou, para um VAX 11/750 que rodava o sistema operacional UNIX, o primeiro Vírus realmente reprodutivo, usando sobretudo instruções de linguagem *Assembly*. Foram tomadas medidas preventivas no sentido de evitar a perda do controle do Vírus no processo de "infecção". Cada "infecção" era executada manualmente, e não foi registrada nenhuma conseqüência desastrosa.

A princípio, a nova criação não possuía nome, mas à medida que se reproduzia e penetrava no sistema nível após nível sem ser detectado, o seu amigo Dr. Len Adleman lembrou-se de um sósia na natureza: o Vírus [LUNDELL, 1989, p.42].

No artigo de número 20, apresentado em [DENNING, 1990, p. 316-355], e escrito por Eugene H. Spafford, Kathleen A. Heaphy e David J. Ferbrache, existe uma menção de que o primeiro uso do termo "Vírus" se refere a um código de computador indesejado ocorrido em 1972 em uma novela de ficção científica chamada *When Harley Was One*, de David Gerrold. Contudo, a descrição do "Vírus" apresentada naquele livro não condiz com a definição de vírus de computador aceitada atualmente, fundamentada nos trabalhos desenvolvidos pelo Dr. Frederick B. Cohen.

Os Vírus de Computador podem ser classificados, conforme o ambiente em que se encontram, em: de pesquisa, ou *"zôo-viruses"* [SARC a, 1998], e "in the wild". Um Vírus em pesquisa é aquele que é desenvolvido com propósitos de estudo e não são (ou ao menos não deveriam) distribuídos para fora do ambiente controlado em que estão confinados. Em outra mão, os Vírus que estão fora de controle e soltos em um ambiente não protegido, são chamados "in the wild". O primeiro Vírus encontrado "in the wild" foi o AppleII, reportado em 1981, conforme [DENNING, 1990], citado por [DRAKOS a, 1994], o qual poderia ser classificado originalmente como um experimento inofensivo, mas causava, por vezes, o término espontâneo de programas.

O primeiro Vírus conhecido para microcomputadores foi identificado em 1986, conhecido como o "Vírus Paquistanês" [WEBER, 1989, p. 93] ou "Pakistani Brain" [LUNDELL, 1989, p.49]. Este Vírus foi escrito por dois irmãos, Basit Fariiq Alvi, na época com dezenove anos, e Amjad Farooq Alvi, com vinte e seis. Eles possuíam um negócio cujo o nome era *Brain Computer Services*, vendendo a baixos preços, *software* para IBM-PC para comerciantes locais e turistas. O proprietário era Amjad, formado em física pela Universidade do Punjad. Em 1985, tomaram conhecimento de que o *software*

que produziam estava sendo pirateado por uma firma de Lahore sem consentimento. Em janeiro de 1986, Amjad decidiu fazer algo para impedir a pirataria e a idéia que lhe surgiu foi a de criar um Vírus que "infectasse" os computadores com cópias não autorizadas do *software* da Brain, prejudicando operações do sistema, obrigando o usuário a contactar Amjad para reparo. O Vírus foi instalado no *software* encomendado. Contudo, inseriram o Vírus também em outros *softwares* que revendiam, como o Lotus 1-2-3 e o WordStar [LUNDELL, 1989, p.51]. Algumas fontes, contudo, afirmavam que o Vírus se tratava de um caso de chantagem para vender o antídoto por um preço que chegava a dois mil dólares [WEBER, 1989, p. 93]. O vírus paquistanês, segundo [WEBER, 1989, p. 93]:

> "(...)é do tipo *boot* vírus, guardando o setor de *boot* original e mais cinco outros setores auxiliares em áreas do disco marcadas como defeituosas. Este vírus roda em IBM-PC, e sua ação consiste em *deletar* arquivos contidos em um disco. Pode ser detectado pelo fato de alterar o rótulo do disco para ©BRAIN, marcar 6 setores do disco como defeituosos e causar excessiva atividade de leitura e escrita em disquetes".

O setor de *boot* de um disco contaminado continha a seguinte mensagem [LUNDELL, 1989, p.52]:

"BEM VINDO À MASMORRA
CUIDADO COM ESTE VÍRUS
CONTACTE-NOS PARA A VACINA"

A mensagem incluía o endereço e telefone da *Brain Computer Services*, em Lahore, Paquistão, e o nome dos dois irmãos.

Ainda conforme [LUNDELL, 1989, p.53], o Vírus ficou totalmente fora de controle, espalhando-se pelos Estados Unidos, infectando cerca de 100.000 discos de PCs da IBM, incluindo cerca de 10.000 só na Universidade George Washington. Isso foi possível devido à maioria dos clientes estrangeiros dos irmãos Brain serem estudantes americanos.

Em paralelo a tudo isso, um tipo especial de Vírus, os *Worms*, eram utilizados como mecanismos legítimos para a realização de tarefas em um ambiente distribuído. Os *Worms* de rede foram considerados promissores para o desempenho no gerenciamento de tarefas em redes de computadores, em uma série de experimentos no

Xerox Palo Alto Research Center, em 1982. Contudo, existia um grande problema, o próprio gerenciamento do *Worm*, no controle do número de cópias em execução [DRAKOS b, 1994].

Os *Worms* só foram anunciados como uma potencial ameaça à Segurança de Computadores, conforme [DENNING, 1990], citado por [DRAKOS b, 1994], quando o *Christmas Tree Exec* atacou um *mainframe* IBM em dezembro de 1987. Ele "derrubou" a rede IBM e BITNET. Contudo, ele na verdade não era um verdadeiro *Worm*, e sim um Cavalo de Tróia (*Trojan Horse)* com um mecanismo de replicação. O usuário recebia um *e-mail* com um cartão de natal que incluía um código executável (REXX). O executável prometia executar uma árvore de natal na tela, e realmente o fazia, mas também enviava uma cópia da mensagem para todos os endereços existentes na lista de usuários.

O *Internet Worm*, foi o primeiro *Worm* verdadeiro a ser disseminado, afirma [SPAFFORD, 1989] citado por [DRAKOS b, 1994], com seu primeiro ataque registrado em 02 de Novembro de 1988. Ele atacou os sistemas SUN e DEC UNIX conectados à Internet (incluindo dois conjuntos de binários, um para cada sistema). Este *Worm* utilizou os protocolos TCP/IP, protocolos comuns da camada de aplicação, *bugs* dos sistemas operacionais, e várias falhas na administração dos sistemas para executar sua propagação. O *Internet Worm* causou vários problemas no gerenciamento da rede, degradando substancialmente o desempenho do sistema e provocando a indisponibilização de serviços na rede (*denial of network services*).

O autor do Vírus/*Worm* da Internet foi Robert T. Morris Jr., na época um estudante de vinte e três anos formado em informática pela Universidade de Cornell. Havia trabalhado nos laboratórios AT&T da Bell em Murray Hill, New Jersey, e estava trabalhando no laboratório Aiken de Harvard, sendo filho de Robert Morris, especialista em segurança computacional e um dos desenvolvedores do Sistema Operacional UNIX [LUNDELL, 1989, p.16,21,27].

De acordo com [LUNDELL, 1989, p.19-20], o *Worm* da Internet era um programa completo, correndo por si próprio e se reproduzindo de máquina para máquina. O objetivo básico era entrar em outras máquinas para garantir sua reprodução. Ele possuía três maneiras de penetrar nos sistemas das máquinas:

1. Ataque ao "Sendmail": entrava nos sistemas dos computadores através de uma brecha no utilitário "Sendmail" do correio eletrônico deixada propositalmente pelo criador do programa;

2. Ataque ao "Fingerd": o *Worm* utilizava-se do "Fingerd" que funciona mais ou menos como uma lista telefônica. O "Fingerd" era atacado derrubando a sua entrada com bombardeio excessivo e rápido de informação. Isso provocava um mal funcionamento da entrada permitindo o *Worm* conduzir-se pelo "Fingerd" até o sistema local pelo qual o operador era responsável. Ele também fazia, antes de se aproveitar de tal mecanismo, uma verificação se o sistema em questão já havia sido infectado anteriormente. Se não, executava o processo. Tudo indica que era essa parte do *Worm* que causava sua reprodução incontrolável;

3. Decodificação de Palavras Chave: após executar as duas operações anteriores, o vírus ficava inativo por um breve período de tempo e então entrava nas contas das pessoas através da tentativa de decifração de senhas, utilizando combinações do nome do usuário, o último, o primeiro, o último mais o primeiro, etc. e mais uma lista de mil palavras de uso comum.;

A partir de então, passaram a surgir, seguindo um crescimento exponencial, diversos outros Vírus, incluindo o aparecimento de novos tipos e mecanismos inovadores de replicação, contaminação e camuflagem. Entre os Vírus que podemos considerar como pertencentes ao conjunto dos primeiros Vírus de Computador (principalmente aqueles que afetam IPM-PCs), estão os Vírus apresentados por [WEBER, 1989, p. 93-104] como casos conhecidos de Vírus até 1989, entre eles: o Vírus Israelense, Israeli, Sexta-Feira ou Jerusalém; o Vírus Lehigh; o Vírus Alameda; o Vírus Ping-Pong; o Vírus Typo; o Vírus Cascata; o Vírus Swap; o Vírus Icelandic; o Vírus DataCrime; o Vírus Vacsina; o Vírus Mix1; o Vírus Dark Avenger; o Vírus Ogre; o Vírus Gfa-Basic; o Vírus Scores; o Vírus nVir; o Vírus MacMag; o Vírus Frank; os Vírus no Atari ST; o Vírus no Amiga.

2.2. Evolução/Crescimento dos Vírus

Hoje, a existência dos Vírus e a necessidade de proteção contra eles é uma realidade inevitável. No início foram criados, como já mencionado, em laboratório de Universidades para demonstrar o potencial de ameaça que tais códigos de *software* poderiam provocar. A partir de 1987, os Vírus se difundiram através de Universidades no mundo todo. Três dos mais conhecidos vírus, que inclusive ainda podem ser encontrados (*Stoned*, Cascata e Sexta-Feira 13), surgiram em 1987 [SARC a, 1998].

Para entendermos melhor o processo de disseminação e crescimento exponencial de ameaças virais aos computadores, poderemos considerar a similaridade com os Vírus biológicos [SYMANTEC a, 1997].

Os Vírus biológicos, há mais de 500 anos atrás, eram transmitidos de comunidade para comunidade através dos viajantes de a pé ou a cavalo. Um viajante alcançava uma cidade e disparava um surto. Na época das grandes navegações se introduziu um segundo nível de ameaça devido ao contato com civilizações diferentes, propiciando troca de doenças entre os povos. Com a Revolução Industrial a transmissão de Vírus biológicos entre povos não passava de meses, com a utilização de navios a vapor e trens. Hoje, a principal forma de transmissão rápida (questão de horas) de Vírus biológicos entre populações é o avião.

De uma maneira similar, o Vírus eletrônico se utilizou primeiramente de disquetes para "viajar" de um computador pessoal a outro. As redes possibilitaram a conexão entre máquinas clientes e máquinas servidoras, gerando um segundo nível de ameaça de transmissão (e infecção). Com a explosão da Internet, e sua utilização como uma ferramenta para comércio, comunicação, pesquisa e entretenimento, foi gerado um terceiro nível de ameaça, através da eliminação das fronteiras entre as redes corporativas e o resto do mundo. Essa situação é altamente propícia para a contaminação em massa por Vírus de Computador por todo o mundo, o que acontece, de maneira similar, com os Vírus biológicos na utilização de aviões como meio de transporte, muito embora a contaminação de Vírus Eletrônicos via Internet parece ser mais eficiente.

A criação de Vírus continua a proliferar-se, e de forma acelerada, devido ao crescimento da população de jovens inteligentes e literados em computador, que

apreciam o desafio mas não a ética, na criação e disseminação de Vírus. Outra razão para a extraordinária rapidez com que se desenvolvem novos Vírus é a facilidade de criá-los, cada vez mais presente devido à utilização de sofisticadas ferramentas de desenvolvimento de *software* [SARC a, 1998]. O surgimento e disponibilização de pacotes de desenvolvimento de Vírus e *Malwares* (incluindo *Worms* e *Trojans* – Cavalos de Tróia) , com rotinas pré-elaboradas e interface gráfica aprimorada, permitindo o uso do conceito de programação visual, levou ao alcance de usuários menos experientes ou preparados, mas não menos nocivos, a possibilidade de geração rápida e relativamente fácil de programas mal intencionados (*malicious codes/ Malwares*) com características viróticas potencializadas. Segundo [MAX, 2000, p.160], os *kits* de criação de Vírus circulando na Internet, estão resultando em um aumento dos vírus "in the wild" (no ambiente selvagem), sendo que alguns desses *kits* são, entre outros: o "Virus Creation Laboratories", o "Virus Factory", o "Virus Creation 2000", o "Virus Construction Set" e o "The Window Virus Engine".

O crescimento exponencial do uso da Internet, vem gerando uma verdadeira pandemia de ataques de Vírus. É possível hoje, com apenas um clique no botão do mouse, transmitir um Vírus de computador e gerar a contaminação em todo o planeta em questão de horas, ou até menos. Conforme [SARC a, 1998], em 1986 existiam apenas quatro Vírus para IBM-PC, sendo que novos Vírus eram uma raridade, sendo que, em média, um Vírus era criado a cada três meses. Em 1989, já existiam casos de aparecimentos de novos Vírus a cada semana. Segundo [WEBER, 1989, p. 104,105], em 1989 existiam cerca de 40 Vírus conhecidos para IBM-PC. Hoje chegam perto da casa dos 50.000 Vírus (de acordo com a lista de Vírus do Norton AntiVirus, da Symantec).

3. CONCEITOS E FUNDAMENTOS TEÓRICOS

3.1. Vírus de Computador

Afinal o que é um Vírus de Computador? Vírus de Computadores são, simplesmente, programas executáveis de computador, com algumas características especiais. Assim como um Vírus biológico, um Vírus de Computador pode encontrar e se anexar ("infectar") a um hospedeiro, que ao invés de células pode ser um arquivo executável ou a área de inicialização do sistema operacional de um computador(*boot Record*).

Segundo [WEBER, 1989, p. 80], o Vírus de computador é na realidade um programa que guarda muitas semelhanças com o homônimo orgânico, o Vírus biológico. Entre as semelhanças, destacam-se a necessidade de um sistema ou programa hospedeiro, a capacidade de se propagar de forma furtiva, tempo de latência, ações que podem variar de brandas até a destruição total dos recursos do sistema infectado.

[WEBER, 1989, p. 81] apresenta a definição que Frederick B. Cohen expôs no seu trabalho de doutorado, do qual se originou o termo Vírus, da seguinte forma:

> "define-se vírus de computador como sendo um programa que pode infectar outros programas, modificando-os de maneira a permitir a inclusão de uma cópia sua. Com esta capacidade de infecção, um vírus pode se propagar em um sistema ou rede, utilizando as autorizações e privilégios dos seus usuários em proveito próprio".

De acordo com [STALLINGS, 1999, p. 504], Vírus são programas que podem "infectar" outros programas através de sua modificação, onde a modificação inclui a cópia do Vírus, podendo infectar, a partir de então, outros programas.

Ainda consoante [STALLINGS, 1999, p. 504], os Vírus biológicos são minúsculos fragmentos de código genético (DNA ou RNA), que podem assumir o controle do mecanismo de uma célula viva e gerar milhões de réplicas do Vírus original. Como na contrapartida biológica, os Vírus de Computador também carregam instruções de código para a geração perfeita de cópias. Ao passo que estiver *logado* em um computador hospedeiro, um Vírus típico toma temporariamente o controle do sistema operacional do computador. Então, toda vez que o sistema do computador infectado

entrar em contato com um pedaço de *software* não infectado, uma nova cópia do Vírus é passada para o novo programa acessado. Portanto, a "infecção" pode se espalhar de computador para computador sem o consentimento do usuário que, não sabendo da existência do Vírus, pode trocar disquetes com outros usuários ou enviar programas através da rede. Em um ambiente de rede, a facilidade de se acessar aplicações e serviços do sistema em computadores remotos, provê um nicho perfeito para a disseminação de Vírus de Computadores.

[VASCONCELLOS, 1999, p. 105] complementa, afirmando existir especialistas que se referem ao Vírus de Computador como "a primeira forma de vida construída pelo homem". O Vírus é capaz de se reproduzir sem a interferência do homem e também de garantir sozinho sua sobrevivência. A reprodução e a manutenção da vida são, segundo alguns cientistas, os requisitos básicos para um organismo ser descrito como vivo. O Vírus *Stoned*, por exemplo, é encontrado até hoje, anos após sua concepção. Contudo, apesar da sofisticação das instruções do código de um Vírus, elas podem determinar o comportamento do mesmo, permitindo a prevenção contra seu funcionamento.

Muito embora existam autores que afirmam ser o Vírus de Computador um programa desenvolvido para alterar, nociva e clandestinamente, sistemas instalados em um computador [GARGAGLIONE, 1999, p. 20], na verdade, o que caracteriza um Vírus de Computador não é o objetivo para o qual foi projetado, seja nocivo ou não, e sim a capacidade de auto-reprodução que possui, similar ao Vírus biológico, e por isso batizado de "Vírus". Conforme [STALLINGS, 1999, p. 505], um Vírus de Computador pode fazer qualquer coisa que outros programas fazem. A única diferença é que ele pode se anexar a outros programas e executar secretamente quando o programa hospedeiro for executado. Uma vez que o vírus esteja executando, ele pode realizar qualquer função, que em alguns casos pode ser apagar arquivos e programas. De acordo com [SARC b, 1998], apenas 35% dos Vírus são deliberadamente malignos.

De acordo com [WEBER, 1989, p. 81], um Vírus é ativado quando um programa "contaminado" é disparado, ou quando o sistema é inicializado a partir de uma mídia contaminada. Uma vez ativo, o Vírus passaria a executar duas rotinas distintas:

- Uma rotina de propagação/reprodução, que procura por programas ainda não contaminados, a fim de infectá-los;

- Uma rotina de atuação/ataque, que realiza uma função qualquer, ativada quando certa condição é satisfeita;

Em função da rotina de atuação, que pode tomar as mais variadas formas, [WEBER, 1989, p. 81] qualifica os Vírus como:

- Benignos: quando não causa maiores danos ao computador ou ao programa ao qual se hospedou. Nenhuma informação é destruída e o único dano caracterizado é a queda no desempenho, devido ao consumo de espaço de memória ou tempo de CPU;
- Malignos: um Vírus maligno quase sempre destrói informação ou provoca qualquer outro dano considerado sério em termos de perdas.

A rotina de atuação pode ser ativada, ainda segundo [WEBER, 1989, p. 82], por diversas situações ou condições, destacando-se: temporal (geralmente relacionado a uma determinada data), contagem de eventos (quando uma quantidade de eventos pré-estabelecidos ocorrerem), espera de uma situação (executada quando uma determinada situação física ou lógica for atingida).

Já de acordo com [STALLINGS, 1999, p. 505], durante seu tempo de vida, um Vírus típico passaria por quatro fases, a saber:

- Fase Dormente: o Vírus não está executando. O Vírus será eventualmente ativado por algum evento, tal como uma data, a presença de outro programa ou arquivo, ou ainda o alcance de algum limite da capacidade do disco. Nem todos os Vírus passam por esse estágio;
- Fase de propagação: o Vírus aplica cópias suas a programas hospedeiros ou em certas áreas do sistema no disco. Cada programa infectado passará a conter um clone do Vírus, que também entrará em uma fase de propagação;
- Fase de Ativação/Disparo: o Vírus é ativado para realizar a função para a qual ele foi planejado. Como na fase dormente, a fase de ativação pode ser causada por uma variedade de eventos do sistema, incluindo o número de cópias para replicação realizadas pelo Vírus;

- Fase de Execução: a função do Vírus é executada. A função pode ser inofensiva (benigna), como uma mensagem na tela, ou nociva (maligna), como a destruição de programas e arquivos de dados;

A maioria dos Vírus executa sua função de uma maneira específica para um sistema operacional particular e, em alguns casos, específica para uma plataforma de *hardware* particular. Portanto, em geral, os vírus são projetados para tomar vantagem de detalhes e fraquezas de sistemas particulares.

Duas classes principais de Vírus podem ser identificadas, quanto ao modo de operação:

3.1.1. Vírus de Disco (*Boot*)

Os Vírus de Disco (*Boot*) "infectam" o registro mestre do *sistema (Master Boot Record* – MBR) e/ou o "*boot sector*" de um disquete (parte do disco responsável pela manutenção dos arquivos). Da mesma maneira que um índice funciona para um livro, ou um fichário em uma biblioteca, um disco precisa de uma tabela com os endereços dos dados armazenados. Qualquer operação que acesse um arquivo no disco, precisa do uso dessa tabela. Salvar ou carregar um arquivo num disquete infectado possibilitaria a ativação do Vírus, que poderia infectar outros disquetes e o disco rígido.

Segundo [GARGAGLIONE, 1999, p. 27], a única maneira de se obter uma contaminação de um Vírus de disco é através da tentativa de se inicializar um sistema a partir de um disquete contaminado. Uma vez executado, o Vírus pode tornar-se residente em memória e também contaminar o MBR do disco rígido. Se o Vírus estiver em memória, uma simples leitura em um disquete não contaminado, proporcionará a "infecção" do mesmo. A maioria dos Vírus de disco é nociva, no sentido de causar danos, seja como resultado do seu ataque ou mesmo ao gravar a área original do MBR ou *Boot Sector* em outro setor, que poderá estar ocupado, sobrescrevendo-o.

De acordo com [SARC a, 1998], um disco não precisa ser de inicialização para ser infectado por um Vírus de *boot*; os dados do disco também podem conter Vírus de *boot*. Uma maneira típica de um computador conseguir uma infecção desse tipo de vírus, é reiniciar com um disquete deixado inadvertidamente na unidade (*drive*). Mesmo se o disquete não é de inicialização (disco de *boot*), o Vírus poderá ser ativado e se

espalhar pelo disco rígido. Quando o sistema é inicializado a partir do disco rígido, o Vírus se espalhará para todos os disquetes que forem inseridos na unidade de disquete.

3.1.2. Vírus de Arquivo

Este tipo de Vírus "infecta" arquivos de programas, geralmente arquivos executáveis ou de sistema (por exemplo com extensões .SYS, .OVL, .COM, .MNU, etc). A idéia básica por trás deste tipo de Vírus é a de se copiar para o início ou fim de um arquivo, de modo que, ao se chamar o programa infectado, o Vírus se ativa, executando ou não outras tarefas e depois ativa o verdadeiro programa. Existem espécies que sobrescrevem o código do programa hospedeiro, gerando danos praticamente irreparáveis ao mesmo.

Consoante [GARGAGLIONE, 1999, p. 28], a única maneira de se gerar uma "infecção" por este tipo de Vírus, é executando um arquivo já infectado. De uma forma geral, o Vírus é ativado e torna-se residente em memória, passando a contaminar outros arquivos, de acordo com sua disciplina/processo de "infecção".

Pode-se dividir, o estudo dos Vírus de Arquivos, de acordo com seu comportamento, tecnologias e princípios utilizados, em:

Vírus Companheiros

Os Vírus companheiros (*companion viruses*) são vírus que não infectam arquivos executáveis de programa. Ao invés disso, criam um arquivo de mesmo nome do arquivo alvo mas com extensão .COM, o qual terá seu atributo alterado para *Hidden* (escondido, no caso do DOS). Como o arquivo .COM é executado sempre antes do .EXE (DOS), o Vírus antecipa-se, entra na memória e depois chama o programa [VASCONCELLOS, 1999, p. 109].

Segundo [SARC a, 1998], os Vírus companheiros são uma exceção à regra de que um Vírus precisa se anexar a um segundo arquivo. Um Vírus companheiro cria um novo arquivo e conta com o comportamento do DOS para executá-lo antes do arquivo de programa que é normalmente executado. Os Vírus companheiros podem usar várias estratégias. Alguns Vírus companheiros criam um arquivo com a extensão .COM com o mesmo nome de um arquivo com extensão .EXE já existente. Por exemplo, um Vírus companheiro poderia criar um arquivo chamado CHKDSK.COM e colocá-lo no mesmo

diretório do arquivo CHKDSK.EXE. Toda vez que o DOS precisar escolher entre executar dois arquivos com o mesmo nome, onde um possui a extensão .EXE e o outro a extensão .COM, o arquivo com a extensão .COM será executado primeiro.

Vírus Polimórficos

O Vírus polimórfico é caracterizado pelo fato de se alterar a cada vez que "infecta" um novo arquivo [VASCONCELLOS, 1999, p. 109]. A permanente mutação objetiva a alteração do próprio código do Vírus, dificultando a ação dos antivírus, cujo mecanismo de rastreamento do Vírus é feito através da análise das assinaturas (seqüências de bits) que os identificam (segundo uma base de dados com tais definições).

Conforme [STALLINGS, 1999, p. 508,509], um Vírus polimórfico cria cópias durante a replicação que são funcionalmente equivalentes mas possuem diferenças nos padrões de bits. Assim como os Vírus *stealth*, o propósito é frustrar programas de rastreamento/varredura de Vírus. Neste caso a assinatura do Vírus variará de acordo com cada cópia. Para efetuar tal variação, o Vírus pode randomicamente inserir instruções ou trocar a ordem de instruções independentes. Uma abordagem mais efetiva é o uso da criptografia. Uma parte do Vírus, geralmente chamada "máquina de mutação", cria uma chave de criptografia randômica para criptografar a parte restante do Vírus. A chave é armazenada com o Vírus, e a própria "máquina de mutação" é alterada. Quando um programa infectado é invocado, o Vírus usa a chave randômica armazenada para descriptografar o Vírus. Quando o Vírus se replica, uma diferente chave randômica é selecionada.

Vírus Invisíveis (*Stealth*)

O nome *Stealth* advém da característica de invisibilidade dos aviões *Stealth*, invisíveis ao radar. Uma das técnicas utilizadas para sua camuflagem, é o uso da compressão de dados, que faz com que o programa infectado figure exatamente com o mesmo tamanho de uma versão não infectada. [STALLINGS, 1999, p. 508] afirma que além da técnica de compressão, diversas outras técnicas são possíveis. Um exemplo

seria um Vírus que poderia interceptar as rotinas de entrada e saída do disco, e assim que existisse uma tentativa de ler porções suspeitas do disco usando essas rotinas, o Vírus apresentará de volta o programa original não infectado. Portanto a tecnologia *Stealth* é usada para evitar a detecção.

Vírus de Macro

De acordo com [VASCONCELLOS, 1999, p. 122], macros são comandos codificados em alguma linguagem, como o *Visual Basic*, organizados de forma que qualquer programa que suporte *Visual Basic* possa executar determinadas tarefas automaticamente, toda vez que for executado, com o objetivo de facilitar o uso de rotinas especiais para melhorar o desempenho do aplicativo.

Vírus de Macro são simplesmente macros criadas em qualquer um desses aplicativos, como o *MS-Word* da Microsoft (alvo principal), desde que suporte a linguagem em que a macro foi escrita (no caso do *MS-Word*, em *Visual Basic*). Por exemplo, toda vez que um documento do *MS-Word* é aberto, macros são automaticamente carregadas (*AutoOpen*, por exemplo), executando determinadas tarefas. No caso dos Vírus de macro, tais tarefas, em geral, causariam algum efeito maléfico no computador. De acordo com [STALLINGS, 1999, p. 508], os Vírus de macro são perigosos por diversas razões, entre elas:

- Um Vírus de macro é uma plataforma independente. A grande maioria dos Vírus de macro "infecta" documentos do Microsoft Word. Qualquer plataforma de *hardware* e sistema operacional que suporte o Word pode ser "infectado";
- Vírus de macro "infectam" documentos e não porções executáveis de código. A maioria da informação introduzida em um sistema de computador está na forma de um documento e não de um programa;
- Vírus de macro são facilmente disseminados. Um método muito comum é através do uso do correio eletrônico;

RetroVírus

São Vírus que tem como alvo os próprios antivírus.

Um terceiro tipo de Vírus pode ser considerado, na união das características das duas classificações anteriores:

3.1.3. Vírus Multi-Parte (*Multipartite*)

Estes Vírus utilizam-se de características tanto do Vírus de Arquivo como do Vírus de Disco (*boot sector/partition table viruses*), podendo "infectar" arquivos e a área de inicialização de discos rígidos ou disquetes. Enquanto um Vírus de disco tradicional se dissemina somente a partir de disquetes "infectados", Vírus multi-parte podem se disseminar através de um Vírus de arquivo, inserindo ainda, uma "infecção" no setor de inicialização ou tabela de partição. Estas características, fazem deste tipo de Vírus algo difícil de ser erradicado [SARC a, 1998].

3.2. Taxonomia

Segundo [WEBER, 1989, p. 81 - 84], pode-se levantar distinção entre os diversos tipos de programas chamados *malicious programs* (*Malwares*), considerando-se a possível existência de três módulos lógicos em um programa , a saber:

- Módulo de Reprodução: capaz de gerar cópias do programa ou de partes do mesmo;
- Módulo de Ataque: módulo que possui o código que executará as tarefas para o qual o programa foi realmente desenhado, no caso de ser um *malicious program*, podendo potencialmente causar problemas ao sistema alvo;
- Módulo Normal: efetuaria tarefas úteis ao usuário, para as quais o programa deveria ter sido concebido;
- Módulo de Defesa: visa defender a integridade dos demais módulos. Pode ser utilizado por "malicious programs" com o objetivo de proteger os módulos de reprodução e de ataque contra sua descoberta e/ou eliminação;

Assim sendo, poderíamos classificar os diversos tipos de programas como:

- Programa Normal: o programa que conteria única e exclusivamente o módulo normal, não causando prejuízos (a não ser por erro na codificação);
- Programa Traidor: aquele que possui apenas o módulo de ataque, sendo que dispararia a sua função, gerando danos ou não (conforme a intenção de quem o projetou), toda vez que o programa fosse ativado;
- Cavalo de Tróia: caracteriza-se por possuir um módulo normal e outro de ataque, iludindo o usuário que, enquanto executa operações supostamente normais (módulo normal), não imagina que tal programa possa estar executando tarefas indesejadas (módulo de ataque). Existem basicamente dois tipos de Cavalos de Tróia: um que usa o módulo normal apenas para mascarar o ataque e outro que possui um funcionamento perfeitamente normal mas cujo módulo de ataque coleta informações sobre o sistema alvo. Portanto, Cavalos de Tróia não se classificam na categoria de Vírus, pois não possuem o módulo de reprodução, embora freqüentemente sejam incluídos como uma espécie de Vírus;
- *Worm* (verme): é um programa auto-contido, capaz de se auto-propagar. Possui apenas o módulo de reprodução e não necessita de um sistema hospedeiro. Eles são muito comuns em redes de computadores, propagando-se de um nó para outro na rede;
- Vírus: programa ou parte de um programa inserido em outro programa. O programa resultante possuirá então três módulos (reprodução e ataque, representados pelo Vírus, e normal, representado pelo programa hospedeiro). É interessante observar que um Vírus obtém o controle do computador quando o seu hospedeiro é executado, pelo simples fato do sistema normalmente não distinguir entre o Vírus e o programa hospedeiro, fazendo com que o Vírus ganhe todos os benefícios e privilégios do hospedeiro;

[STALLINGS, 1999, p. 502-505] sugere uma taxonomia alternativa baseando-se na divisão dos *malicious programs* (*Malwares*) em duas categorias: aqueles que precisam de um programa hospedeiro e aqueles que são independentes. Outro fator levado em consideração, é a característica do programa quanto à replicação, ou seja, a divisão dos programas entre aqueles que podem se reproduzir e aqueles que não possuem essa capacidade (figura 1).

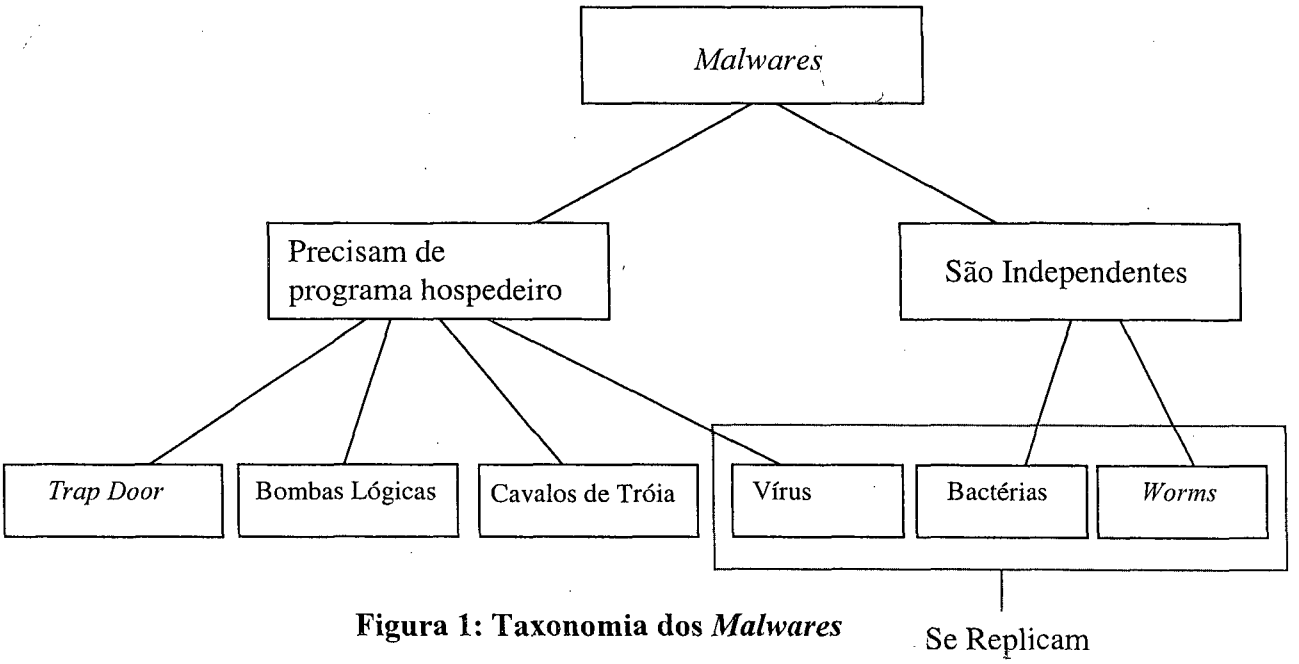

Figura 1: Taxonomia dos *Malwares*

Embora a taxonomia apresentada seja bastante útil, é importante salientar que alguns dos tipos de ameaça representados podem atuar em conjunto com outros, aumentando o potencial de dano.

Segue-se uma breve descrição dos elementos cidados na figura 1:

- *Trap Doors* : Uma *Trap Door* é uma entrada secreta em um programa que permite que alguém que possua o conhecimento dela, possa obter acesso sem passar pelos procedimentos usuais de segurança. Por muitos anos, programadores se utilizavam de T*rap Doors* para encontrar erros em programas em desenvolvimento bem como testá-los. Isso era usualmente implementado quando do desenvolvimento de uma aplicação que possuísse um procedimento de autenticação, ou um longo processo de configuração, requerendo que o usuário entrasse com vários valores para rodar a aplicação. Para procurar erros de codificação, o desenvolvedor desejava acesso com privilégios especiais para se abster da necessidade da configuração e autenticação. O programador poderia também desejar garantir a existência de um método alternativo de ativação do programa no caso de algo estar errado com o procedimento de autenticação. *Trap Door* é um código que reconhece alguma seqüência especial de entrada ou é engatilhado quando executado por um determinado usuário ou por uma improvável seqüência de eventos. É difícil implementar controles de sistemas

operacionais para *Trap Doors*. Medidas de segurança precisam colocar em foco o desenvolvimento do programa e atividades de atualização do *software*;

- Bombas Lógicas: um dos tipos mais antigos de *Malware*, anterior inclusive aos Vírus e *Worms*, é a Bomba Lógica. Trata-se de um código embutido em algum programa legítimo sendo configurado para "explodir" quando certas condições são satisfeitas. Exemplos de condições que podem ser usadas como gatilho para uma Bomba Lógica são a presença ou ausência de certos arquivos, uma data particular ou um determinado usuário rodando uma aplicação. Quando disparada, uma Bomba Lógica pode alterar ou apagar dados ou arquivos inteiros, causar a interrupção de serviços na máquina, etc.
- Cavalos de Tróia (*Trojan Horses*): trata-se de um programa útil, ou aparentemente útil, contendo código oculto que, quando invocado, realiza alguma atividade indesejada podendo ser nociva ou não. Normalmente são utilizados para permitir que usuários não autorizados executem funções indiretamente, as quais não podem executar diretamente. Outra motivação é a destruição dos dados. Enquanto aparentemente o Cavalo de Tróia executa funções normais, ele pode estar apagando os arquivos dos usuários silenciosamente;
- Vírus: um Vírus é um programa que pode "infectar" outros programas através da modificação dos mesmos; a modificação inclui uma cópia do Vírus, permitindo a "infecção" de outros programas;
- *Worms* (Vermes): um programa *Worm* usa as conexões de rede para se espalhar de sistema a sistema. Uma vez ativado dentro de um sistema, um *Worm* pode comportar-se como um Vírus ou Bactéria ou mesmo implantar um Cavalo de Tróia a fim de realizar alguma ação. Para se replicar, um *Worm* de rede pode usar algum dos seguinte veículos: facilidades do correio eletrônico; capacidade de execução remota; capacidade de *login* remoto. A nova cópia do *Worm* quando atinge seu alvo, é então executada no sistema remoto, onde, além de qualquer atividade que ele venha a executar naquele sistema, ele continua a se disseminar. Um *Worm* de rede exibe as mesmas características de um Vírus de computador, contendo as fases de dormência, propagação, ativação e execução. A fase de propagação é normalmente caracterizada pelas seguintes funções: a) Procura de um sistema remoto para "infectar" através da análise da tabela de *hosts* ou repositórios similares de

endereços de sistemas remotos; b) Estabelecer a conexão com um sistema remoto; c) Copiar a si próprio para o sistema remoto e executar a cópia.

- Bactéria: são programas que não danificam explicitamente qualquer arquivo. Seu único propósito é a replicação. Um caso típico poderia não fazer nada mais que executar duas cópias de si mesmo simultaneamente em um sistema multiprogramado, gerando um alto grau de replicação. As Bactérias se reproduzem exponencialmente, eventualmente consumindo toda a capacidade do processador, memória ou espaço em disco, desabilitando o acesso dos usuários a outros recursos (*Denial of Services*).

4. COMO FUNCIONA O PROCESSO DE REPRODUÇÃO E DISSEMINAÇÃO DE UM VÍRUS DE COMPUTADOR

4.1. Introdução

Um Vírus é um programa projetado e escrito para se mover através dos computadores sem o conhecimento ou permissão dos usuários, copiando a si mesmo de hospedeiro para hospedeiro. Os Vírus de computador não são espontaneamente encontrados em arquivos nos computadores. Eles precisam ser escritos por alguém com o propósito específico de estar criando um Vírus. Eles não aparecem acidentalmente.

Um Vírus, segundo [STALLINGS, 1999, p. 505], pode se anexar no início ou final de um programa executável, ou ainda pode ser embutido de alguma outra forma. A chave de sua operação reside no fato de que um programa "infectado", quando invocado, primeiro executará o código do Vírus e então executará o código original do programa.

Uma vez o Vírus tenha entrado no sistema através da "infecção" de um simples programa, ele estará em posição de "infectar" alguns ou todos os outros arquivos executáveis ou de dados naquele sistema, quando o programa inicialmente "infectado" executar. Portanto, uma "infecção" viral pode ser completamente evitada através da prevenção da primeira "infecção" no sistema. Contudo, a prevenção muitas vezes não é muito simples de ser executada.

De acordo com [STALLINGS, 1999, p. 509], a maioria das "infecções" virais iniciam através de disquetes com arquivos "infectados", que são copiados para a máquina, gerando possivelmente uma "infecção" generalizada no sistema. Contudo, hoje, apenas há aproximadamente dois anos após a publicação da obra de William Stallings consultada, as estatísticas de empresas desenvolvedoras de ferramentas antivírus apontam ser a Internet, principalmente através do Correio Eletrônico, o meio através do qual ocorrem o maior número de "infecções" (55% dos casos, segundo a Trend Micro – dados de 1998. A *International Computer Security Association* – ICSA, aponta 87% dos casos – dados de 2000).

4.2. Como os Vírus se espalham através dos sistemas

[COHEN, 1994, p. 2-5] defende que o conceito geralmente visto pela maioria das pessoas, o de que "Um Vírus é um programa que pode 'infectar' outros programas através de sua modificação para incluir uma, possivelmente envolvida, versão de si próprio", não é definitivo. Sua conclusão é de que um Vírus pode se espalhar de programa a programa e de usuário a usuário, ao se analisar, por exemplo, um sistema multiusuário. Para uma melhor explicação, pode-se lançar mão de um exemplo mais elucidativo (Figura 2). Imagine um sistema *time-sharing* com três usuários: U1, U2 e U3; eles possuem três programas P1, P2 e P3, respectivamente, em tempos t1, t2 e t3. Se no tempo t1, o programa P1 é infectado com um Vírus V, e no tempo t2 o usuário U2 roda o programa P1, então devido ao fato do usuário U2 estar autorizando o programa P1 a agir como dele e o usuário U2 estar autorizado a modificar o programa P2, o programa P2 passa a ser "infectado". De um modo similar, se no tempo t3 o usuário U3 rodar o programa P2, o programa P3 torna-se "infectado". Afinal, Cohen complementa afirmando que tal exemplo expõe um caso típico de um ambiente de computação *time-sharing*, com os mecanismos de proteção padrão implantados. Sendo assim, finaliza mostrando que os mecanismos de proteção usados pela comunidade computacional são inadequados para se defender contra os Vírus.

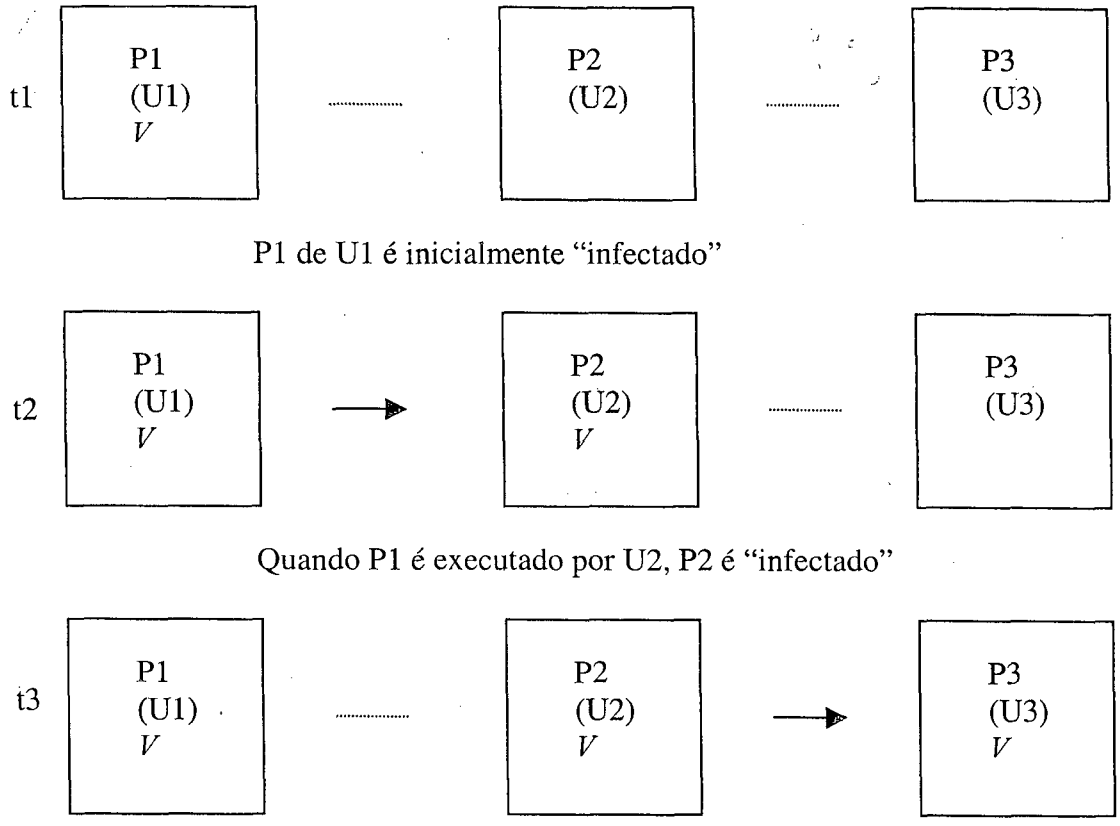

Figura 2: Um sistema *time-sharing* (multiusuário)

Analisemos, agora, um pseudocódigo de um possível Vírus apresentado por Cohen.

Programa V:=
{ 1234567;
Sub-rotina infecta-executável :=
{ loop: arquivo =executável-aleatório;
Se (primeira-linha do arquivo = 1234567)
então vai-para loop;
Senão anexa V no início do arquivo;}
Sub-rotina executa-dano :=
{Qualquer dano que você possa programar}

Sub-rotina gatilho-puxado :=

{Qualquer gatilho que você deseje}

Programa-principal-do-Virus :=
{ infecta-executavel;
Se (gatilho-puxado) então executa-dano;
Vai-para próximo;}
próximo:
}

O pseudocódigo do Vírus *V* funciona da seguinte forma: ele começa com um marcador "1234567". Isto não é necessário, mas neste caso em particular, ele é usado para identificar que este Vírus em particular já infectou um dado programa; assim não é necessário infectar o mesmo programa repetidamente. Ele possui então três sub-rotinas seguidas pelo programa principal. O programa principal do Vírus inicia infectando outro programa através da sub-rotina "infecta-executável". Esta sub-rotina fica em *looping*, examinando arquivos executáveis de um modo aleatório até encontrar um arquivo sem a primeira linha "1234567". Quando ele encontra um executável não infectado, *V* copia seu próprio código para o início do arquivo analisado (ainda não "infectado"), portanto "infectando-o". Quando todos os executáveis estiverem "infectados", o Vírus ficará indefinidamente nesse *loop* .

Depois da infecção, o Vírus verifica a condição "gatilho-puxado", que pode ser qualquer coisa, como uma seqüência de eventos. Se a condição estiver ativa, ele realiza qualquer dano que esteja programado na sub-rotina "executa-dano". Finalmente, o programa principal do Vírus salta para o final do código do Vírus e executa o programa "infectado" normalmente. Assim, se o Vírus estiver no início de um programa executável, e o mesmo for executado, o Vírus irá se anexar no início do próximo programa e se gatilho não estiver puxado, ele simplesmente permitirá a execução do programa ao qual está anexado. Este processo é relativamente rápido e os usuários raramente notariam alguma alteração na execução durante a operação normal do sistema.

Poderíamos aprofundar mais os exemplos para dar uma idéia de que tipo de dano poderiam ser executados. Imaginemos o seguinte:

gatilho-puxado :=
{ Se a data é maior que 01/01/2001;}

executa-dano :=
{ loop: vai-para loop;}

Esse é um exemplo simples de *denial of services.* A condição de gatilho é dirigida nesse caso pelo tempo, e o dano seria um *loop* infinito. Se este Vírus se disseminasse por um computador ou rede, então assim que a data especificada no gatilho fosse alcançada, todo programa "infectado" entraria em um *loop* infinito. Assim, em uma típica rede de computadores, todo programa na rede pararia de operar a partir daquele momento.

Este é apenas um dos mais simples e óbvios tipos de ataque, sendo relativamente fácil detectá-los e controlá-los, não requerendo muita sofisticação para contê-los. Contudo poderíamos ter situações mais complexas. A figura 3 mostra um Vírus mais interessante, que poderíamos chamá-lo de "Vírus de Compactação" (CV).

```
Programa CV :=
{ 01234567;
sub-rotina infecta-executável :=
        { loop:
        arquivo := arquivo-executável-aleatório;
        Se primeira-linha do arquivo = 01234567
        então  vai-para loop;
(1)     compacta arquivo;
(2)     anexar CV no início do arquivo;
        }

programa-principal :=
        { Se permissão-ok
```

então infecta-executável;

(3) descompacta o resto-do-arquivo;

(4) executar o arquivo descompactado;

}

}

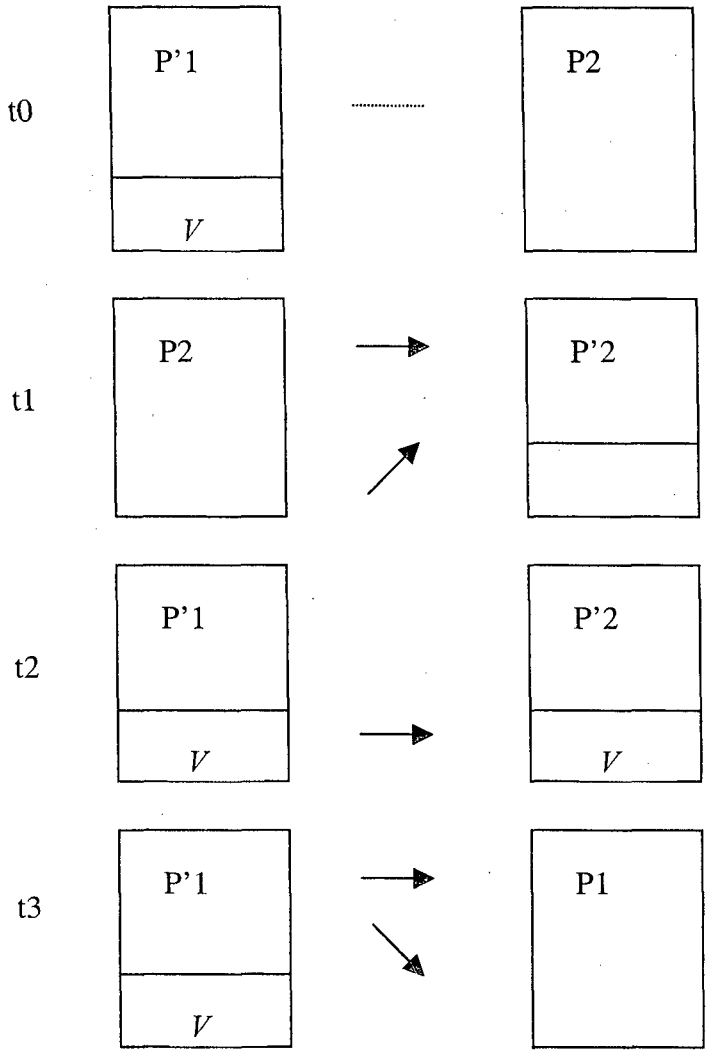

Figura 3: Um Vírus de Compactação

Este Vírus funciona em um processo de quatro passos, demarcados 1, 2, 3 e 4 no pseudocódigo. Analisando a figura o processo pode ser melhor demonstrado. Ao iniciar no tempo t0 com o programa P'1, uma versão infectada do programa P1, e um programa

limpo P2, que nunca foi infectado com esse Vírus. Se no tempo t1, o programa P1 é executado, temos os seguintes passos:

t1 – O programa P2 é compactado para P'2.

t2 – *V* se anexa a P'2.

t3 – P'1 é descompactado para o programa original P1.

t4 – O programa original P1 é executado normalmente.

Existem alguns fatores que merecem menção no caso particular desse Vírus de compactação. Um deles é que o tamanho de P'2 é tipicamente reduzido à metade de seu original P2. Ou seja, a compressão do arquivo salva aproximadamente 50% do espaço ocupado. O efeito total é que se esse vírus for espalhado através de um sistema, você pode salvar em torno da metade do espaço ocupado pelos programas. A penalidade que se paga por essa economia de espaço seria o tempo extra necessário na descompactação dos arquivos toda vez que eles forem executados.

Esse tipo de Vírus poderia facilmente burlar a detecção baseada na comparação com o tamanho original dos arquivos não infectados. Exemplificando, uma vez que o programa resultante P'2 possui alguns *bytes* a menos que seu original P2, torna-se fácil o mascaramento do tamanho do arquivo através do simples preenchimento de P'2 com *bytes* nulos, de forma que P'2 fique exatamente do mesmo tamanho que P2 e assim, não haveria a detecção de tal infecção.

Um sistema de detecção baseado em uma simples verificação de *checksum* também poderia ser burlado por um Vírus como esse. A técnica baseada em *checksum* simplesmente captura um *chechsum* do arquivo original e utiliza esse valor para detectar a infecção por Vírus, uma vez que os arquivos infectados apresentariam um *checksum* diferente. Contudo como o arquivo contaminado possuirá um tamanho menor que o original, o preenchimento do mesmo para que fique do exato tamanho de seu original, pode ser feito não com *bytes* nulos mas sim com bytes que façam com que o *checksum* também seja idêntico ao calculado no arquivo original. Da mesma forma, análises baseadas em códigos CRC (*cyclical redundancy check)* podem ser facilmente forjados.

Assim, percebe-se que Vírus bem projetados podem burlar análises de tamanho, data de modificação, *checksum*, e CRC, e portanto, tais Vírus apresentam um certo grau de dificuldade para serem detectados em suas ações.

De acordo com [PITKOWSKI, 1992, p. 18-26], uma das principais características de um Vírus de computador é que ele se executa porque outro programa é executado e existem dois meios para que isso aconteça: a) o Vírus se insere em outro programa; b) o Vírus liga-se a outro programa.

Um programa não "infectado" apresenta-se, de uma forma geral, como na figura 4, abaixo, onde a execução começa num ponto de entrada que é definido pelo DOS (ou outro sistema operacional):

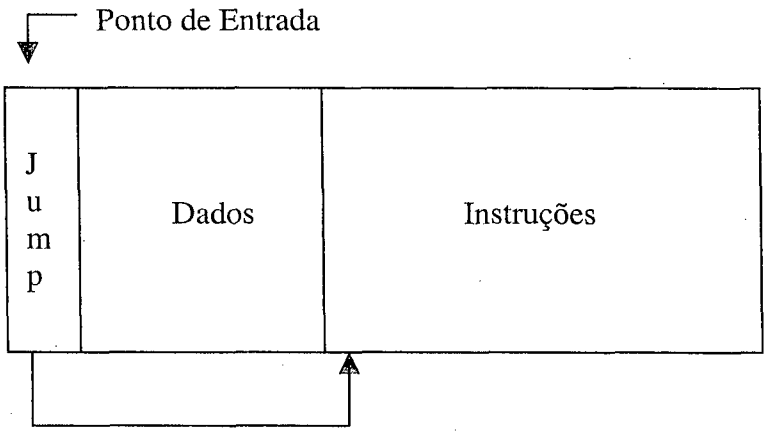

Figura 4: Arquivo não "infectado"

Muitas vezes a primeira instrução de um programa é um *Jump* para as intruções do programa, em seu interior.

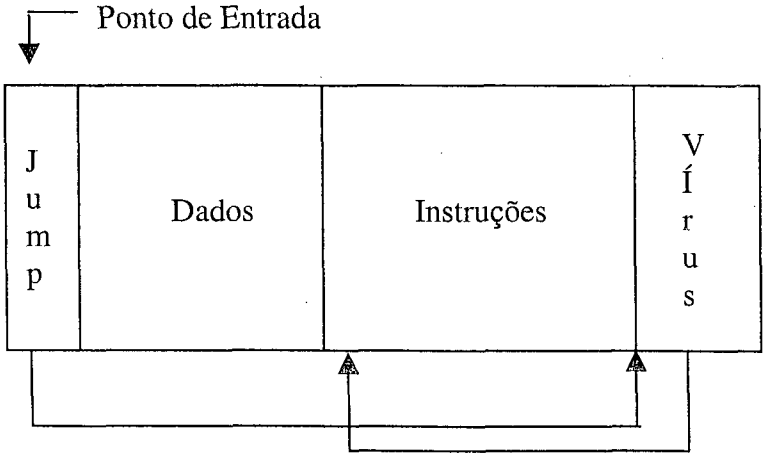

Figura 5: Arquivo "infectado"

Conforme a figura 5, um Vírus modificou a instrução inicial de *Jump* para o fim do programa, onde existe uma cópia agregada do Vírus. A última instrução do Vírus geralmente é um *Jump* de volta pra o começo das instruções do programa. O resultado dessa modificação realizada pelo Vírus, é que toda vez que o programa infectado for executado, o código do Vírus é executado primeiro.

A figura 6, abaixo, apresenta o diagrama do processo normal de inicialização de um computador com o DOS como sistema operacional, inicializado por um disquete ou disco rígido.

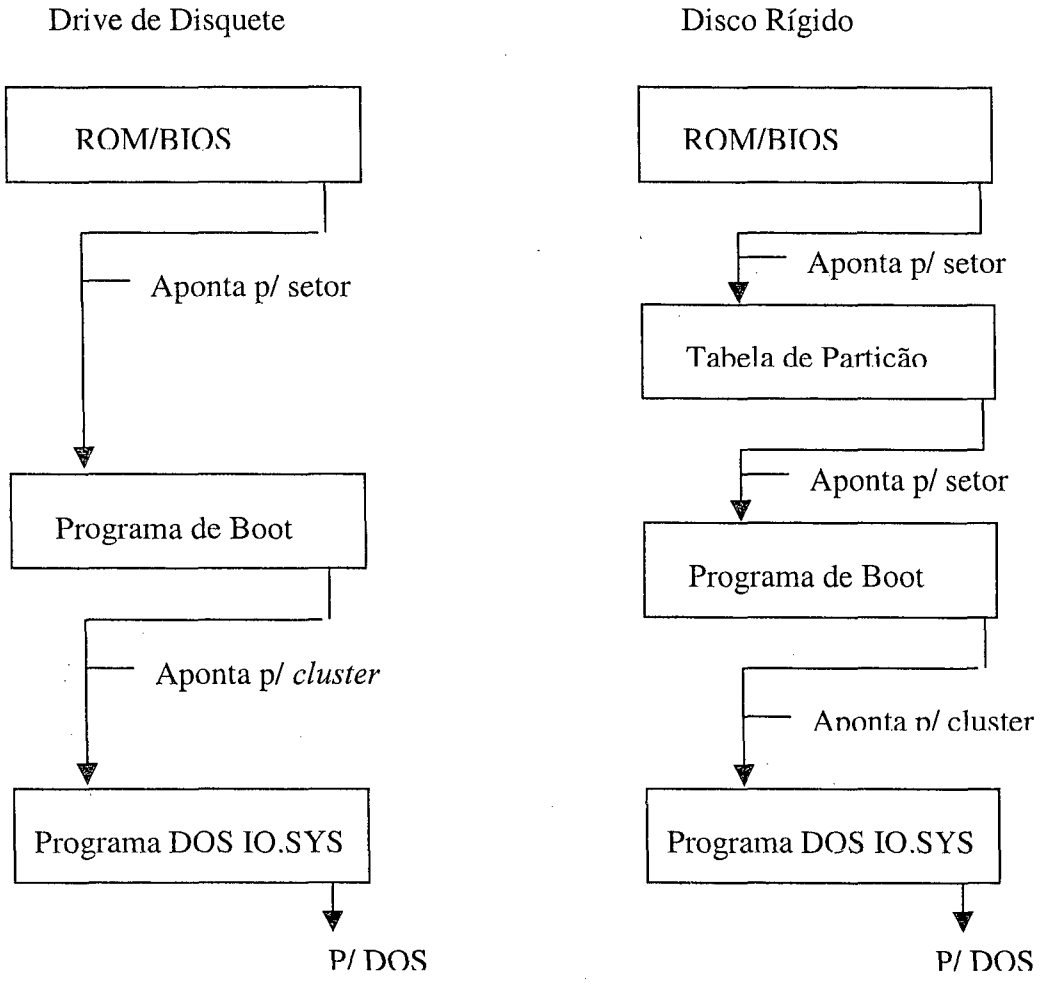

Figura 6: O processo de inicialização

A figura apresenta setores especiais que contêm os arquivos de sistema, que são programas executados sempre que ocorre um *boot* (carga de inicialização) a partir da ROM/BIOS. O programa de *boot* não carrega estes programas em memória, mas aponta para o primeiro *cluster* do diretório de entradas do DOS. O programa contido na

ROM/BIOS sempre aponta para o primeiro setor físico de um dispositivo de disco que está tentando dar *boot* no PC. No caso de um disquete, o primeiro setor físico sempre contém o programa de *boot*, como apresentado na figura 6 à esquerda. No caso de discos rígidos, o primeiro setor físico sempre contém a tabela de partição.

A figura 7 apresenta como um vírus pode tirar vantagem do procedimento de *boot* de um sistema operacional, no caso o DOS.

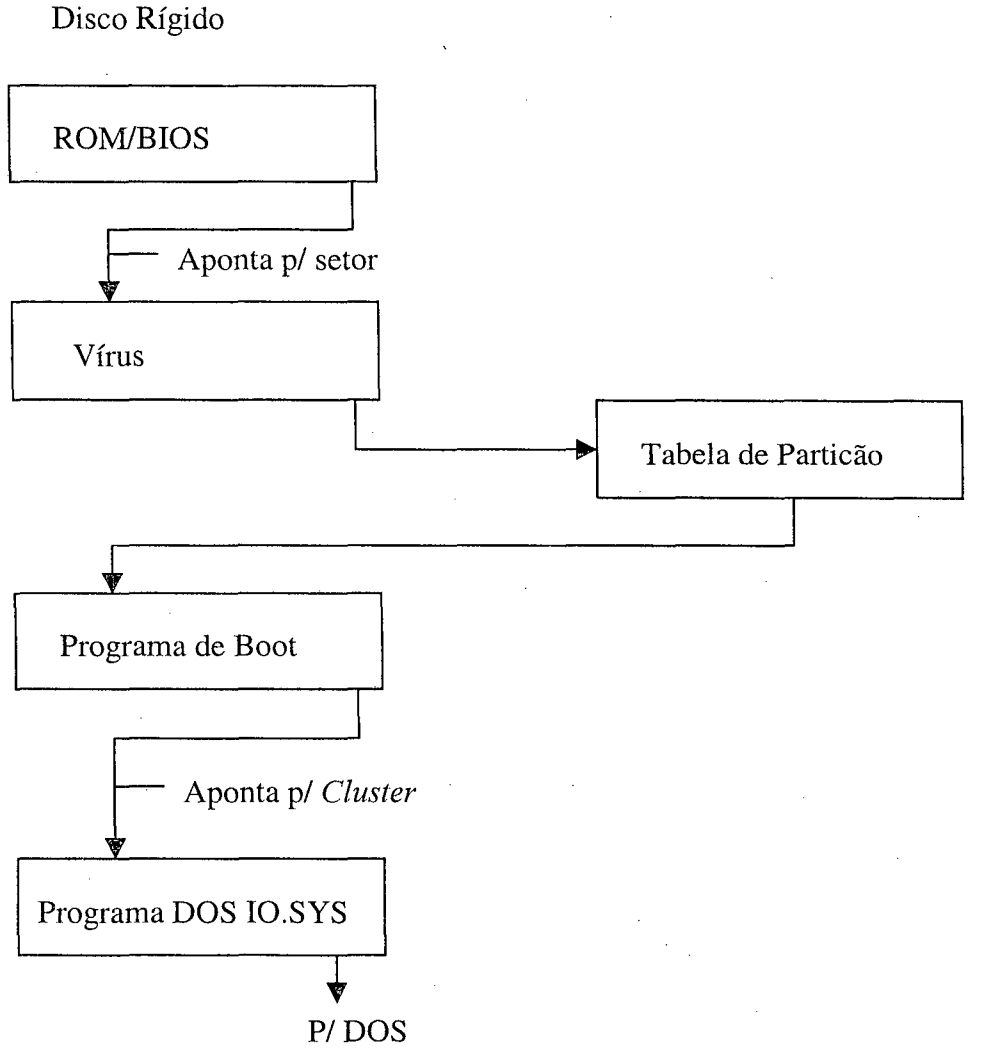

Figura 7: A "infecção" no processo de *boot*

Nesse caso, o Vírus moveu a tabela de partição para outro setor qualquer, tendo copiado a si próprio para o setor normalmente ocupado pela tabela de partição. Sendo assim, o programa da ROM/BIOS executa automaticamente o programa de Vírus a cada vez que o PC inicializa, dando oportunidade ao Vírus para se instalar em memória RAM

primeiramente. Finalmente o Vírus liga-se ao setor em que gravou a tabela de partição original e o PC inicializa normalmente.

A maioria dos Vírus de *boot* seleciona (de acordo com seu projeto) os *clusters* que serão usados para gravar a tabela de partição original, marcando-os como blocos ruins (*BAD blocks*) na FAT (tabela de alocação de arquivos). Tal procedimento é para a defesa da área para qual transferiu a tabela de partições, evitando que o sistema operacional possa sobregravar esses *clusters* mais tarde. Se por acaso já existirem dados gravados nesses *clusters*, os mesmos serão sobrescritos.

As figuras 8 e 9, apresentam como, ainda de acordo com [PITKOWSKI, 1992, p. 18-26], um Vírus pode se agregar a outro programa alterando a estrutura de um diretório.

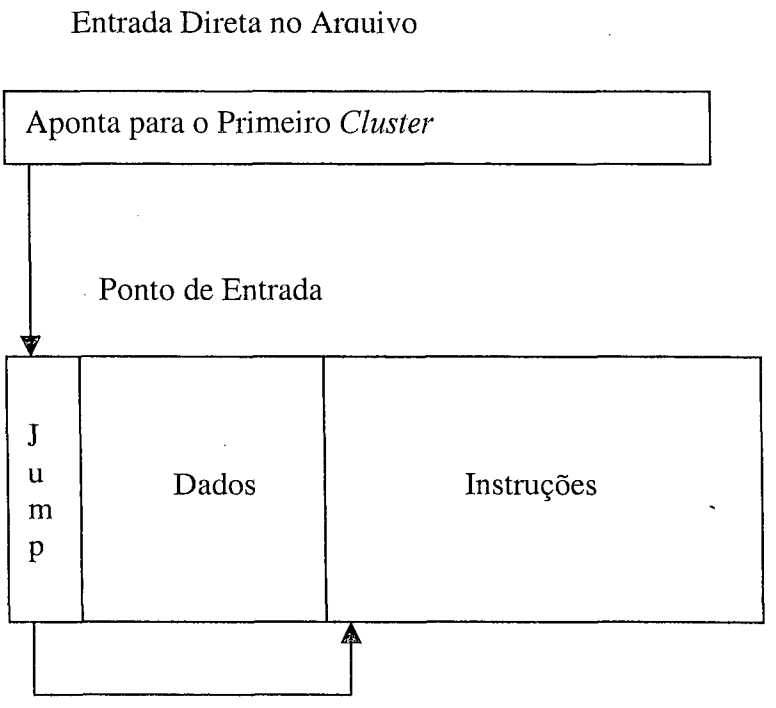

Figura 8: Sem "infecção"

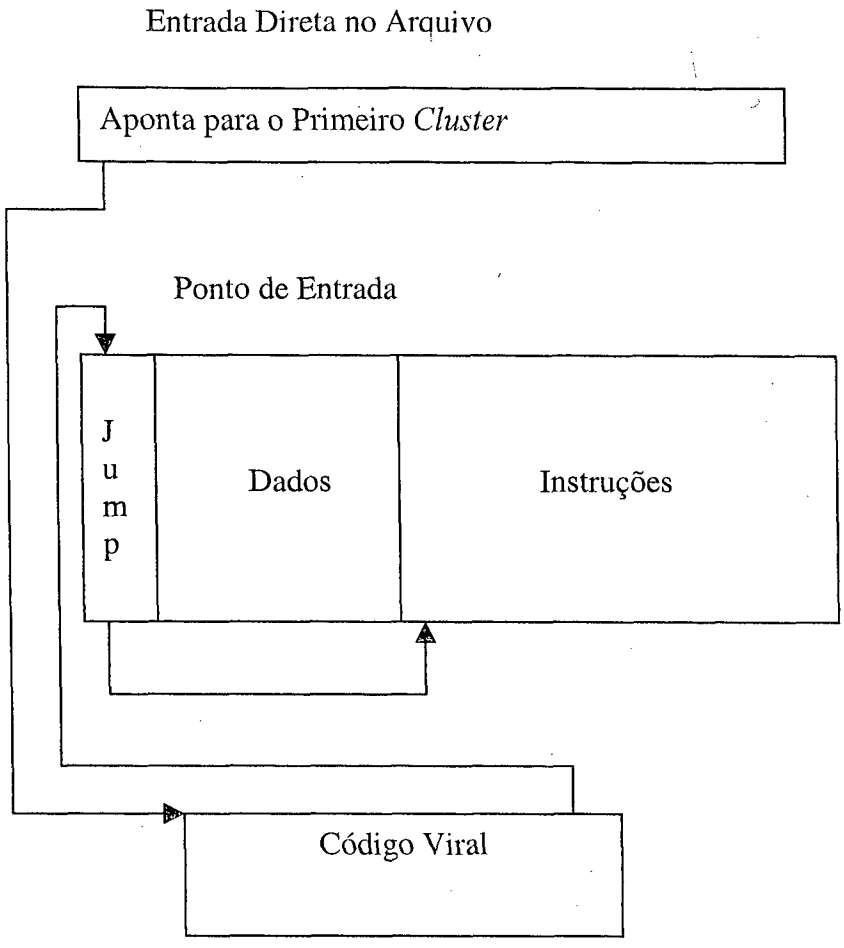

Figura 9: Com "infecção"

O programa não "infectado" começa a ser executado a partir do primeiro *cluster* (que é a própria entrada do subdiretório). No caso de o programa estar infectado, a entrada do diretório foi alterada para apontar para o código do Vírus, isto tudo no primeiro *cluster*, e a última instrução do código do Vírus aponta para o primeiro *cluster* do programa.

5. TECNOLOGIAS E METODOLOGIAS DE DEFESA

5.1. Abordagens Utilizadas por Metodologias e Ferramentas AntiVírus

Existem diversos métodos de combater a ameaça do Vírus digital, seja no uso de metodologias ou na aplicação de ferramentas antivírus. [CIDALE, 1990, p. 37] afirma que os antivírus podem ser classificados como: *software* de prevenção (interrompem o processo de reprodução e evitam que os Vírus contaminem o sistema), *software* de detecção (detectam a contaminação imediatamente após a sua ocorrência) e os *softwares* de identificação (programas específicos que identificam com precisão o Vírus que está atuando no sistema).

5.1.1. Prevenção

[WEBER, 1989, p. 113], classifica a prevenção como um tipo de programa que tenta evitar que Vírus "infectem" os sistemas, apresentando algumas características, dentre elas: a proibição de alteração em programas executáveis, a proibição da instalação de programas que fiquem residentes em memória (com exceção das instalações explícitamente aprovadas pelos usuários), a recusa de execução de programas que não estejam em uma lista de programas totalmente testados e aprovados. Esses tipos de ferramentas, não são específicas contra Vírus individuais, mas em outra mão fornecem uma proteção genérica contra a maioria das tecnologias de Vírus empregadas, com exceção, por exemplo, dos Vírus de *boot*, que executam antes do sistema estar completamente iniciado e portanto dificultam sua verificação. Os sistemas automatizados de prevenção antivírus formam a primeira linha de defesa, podendo ainda implementar adicionalmente esquemas de proteção através de controle de acesso.

De acordo com [DENNING, 1990, p. 330-334], prevenir uma "infecção" viral é a melhor forma de se proteger contra possíveis danos. Se um Vírus não pode se estabelecer dentro de um sistema então ele também não conseguirá causar danos ou

sobrecarga de recursos. Contudo, a prevenção não pode ser executada através apenas do uso de tecnologias, pois ela envolve muito mais do que isso, envolvendo desde a implantação de políticas de segurança, controle de distribuição de *software*, educação adequada dos usuários, até uso de ferramentas antivírus de comprovada eficácia e eficiência. No caso da implantação de uma política de segurança com enfoque na prevenção de Vírus e ameaças automatizadas de *software*, a mesma deve englobar orientações para o uso responsável dos recursos de informática disponibilizados aos envolvidos, a criação de uma equipe técnica que esteja pronta a responder por incidentes e manutenção segura do ambiente, a educação e treinamento aos usuários, etc. Além disso, a política deverá ser formalizada e circulada pelo ambiente e todos os envolvidos devem ser conscientizados. O propósito de tal política é estabelecer regras robustas e eficazes a todos os membros da organização com o objetivo de controlar as ameaças automatizadas de *software* e garantir a execução normal do trabalho de todos, de forma a manter a produtividade.

O controle da aquisição de *software* também é um aspecto importante a se relevar, visto que um vírus, para se disseminar, precisa de um meio de transporte, seja através de uma rede eletrônica ou através do compartilhamento manual de mídias contaminadas. A restrição no uso de *software* cujo fabricante é desconhecido ou não é confiável é algo a se ponderar. Entretanto, como é complicado definir que fontes podem ser confiáveis, é interessante o uso de ferramentas automatizadas que auxiliem nesse processo, fazendo uma varredura na entrada e não permitindo que programas mal intencionados ou arquivos contaminados adentrem na organização. Contudo, complementarmente, a melhor política é restringir a entrada de qualquer *software* que não seja necessário na condução dos projetos que a organização estiver realizando.

Uma proteção mínima adicional contra Vírus seria a encriptação de determinados arquivos. Se os arquivos alvos de "infecções" virais não estiverem no formato esperado, possivelmente o Vírus não os compreenderá, então usualmente ele será incapaz de "infectá-los". Contudo isso poderia funcionar apenas para um conjunto limitado de arquivos e dependeria da tecnologia empregada em apenas algumas espécies de Vírus.

O uso de ferramentas antivírus se torna uma pré-condição em uma abordagem preventiva, já partindo para uma estratégia alternativa. Contudo, deve-se lembrar que as

ferramentas antivírus nunca estarão em condição de defender o ambiente contra todas as espécies de Vírus e *software* mal intencionados existentes, pelo menos não com as tecnologias empregadas por tais ferramentas na atualidade. Por isso a prevenção, se possível, é sempre a mais eficaz das abordagens.

Para evitar a disseminação de Vírus no ambiente da organização, [COHEN, 1994, p. 57-64] afirma existir apenas três coisas que se pode fazer para, absolutamente e perfeitamente, prevenir a replicação e disseminação de Vírus digitais através dos sistemas e redes de computadores: limitar a distribuição e compartilhamento, limitar a transitividade e limitar a funcionalidade e acesso a recursos dos sistemas. Para se evitar a distribuição e compartilhamento, Cohen sugere um modelo que ele chamou de POSet (*Partially Ordered Set*). Basicamente o POSet pode ser desenhado como ramos de caixas com linhas entre elas, e uma regra que regule o fluxo da informação, como mostrado na figura 10.

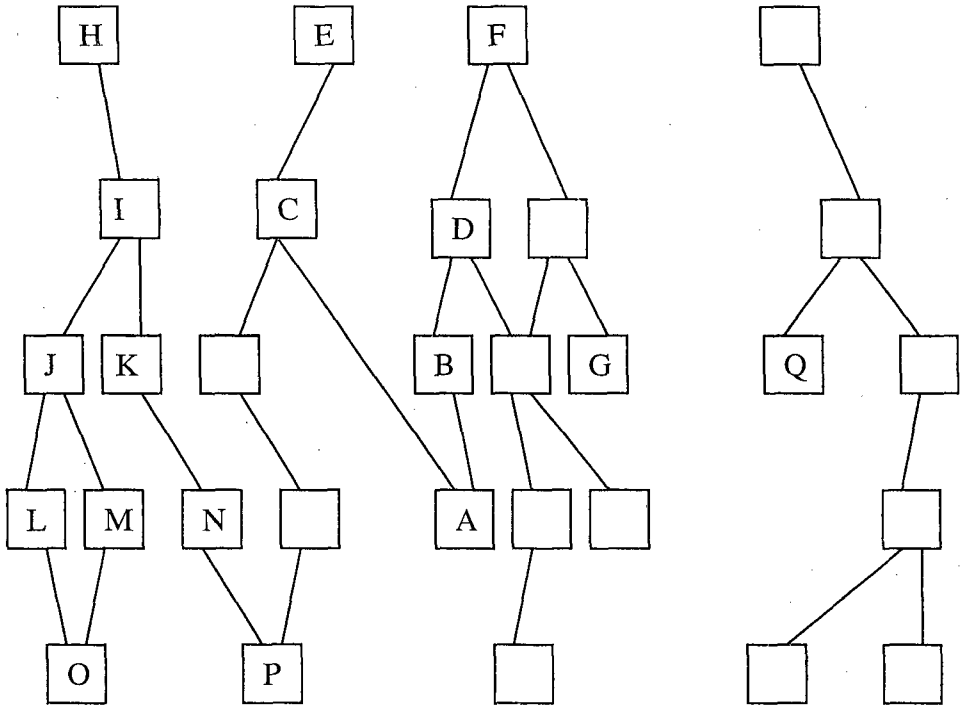

Figura 10: O POSet sugerido por Frederick B. Cohen

O POSet funcionaria da seguinte forma: suponha que alguém em A escreva um Vírus de computador. Ele poderia disseminar tal Vírus para B, C, D, E, e F, mas não

conseguiria levá-lo a G, pois não existe nenhum caminho de A a G. Da mesma forma, se alguém em H estiver tentando acessar informações, ele poderia potencialmente acessar informações de I, J, K, L, M, N, O, e P, mas não conseguiria nunca obter informações de Q, visto que não existe caminho para a informação em Q trafegar até H. Em outras palavras, o que o POSet faz é limitar o fluxo de informação, sendo a estrutura mais generalizada de se fazer isso. Entre as vantagens, está a limitação do grau de disseminação de um Vírus, ficando restrito a um subconjunto da organização, bem como limitar o acesso e vazamento de informações, e gerando a possibilidade de que se um desses casos acontecer, poder se rastrear as origens dos eventos.

A limitação na transitividade, segue o princípio de relações de confiança, onde por exemplo, A pode se comunicar com B e B com C, mas A não pode se comunicar com C. Isto faz com que uma ameaça de Vírus tenha um espaço limitado de disseminação. Contudo, tal abordagem se torna inviável quando posta em prática, visto que um vírus pode ser considerado um conjunto de informações como qualquer outra, permitindo burlar os limites de transitividade, uma vez que no exemplo dado, B poderia ser "infectado" por A e poderia transmitir a "infecção" para C, pois C confia em B.

A limitação funcional a certos recursos e informações disponibilizadas no sistema, é mais uma medida que permite obter um maior controle do ambiente organizacional. Ou seja, só teriam direitos de acessar determinas informações ou certos recursos dos sistema quem estivesse explicitamente autorizado para isso. O problema de tal abordagem, é que os mecanismos adotados para se efetivar tais controles devem ser absolutamente eficazes e também eficientes de maneira a não diminuir a produtividade dos membros envolvidos e também ser à prova de fraudes, pois o acontecimento de uma fraude seria caracterizado como uma falha absolutamente grave no esquema de segurança implantado. De outra forma, o acesso disponibilizado a apenas pessoas autorizadas, faz com que a disseminação de Vírus seja dificultada e controlada.

Consoante a pesquisa elaborada por [STALLINGS, 1999, p. 510], a solução ideal contra as ameaças automatizadas de *software* seria a prevenção, ou seja, em primeiro plano, não permitir que um Vírus de computador invada seus sistemas. Contudo, Stallings afirmara, de outro modo, que tal prevenção é praticamente impossível de ser executada a contento, e alternativamente, a melhor abordagem a ser

utilizada, então, seria a utilização de ferramentas antivírus automatizadas baseando-se nos seguintes princípios: detecção, identificação e remoção.

5.1.2. Detecção

A detecção caracteriza-se no fato de que, uma vez a "infecção" tenha ocorrido, determinar que ela ocorreu e localizar o Vírus. O que se espera de uma ferramenta de detecção é que a mesma seja eficaz na identificação de todos os arquivos "infectados" no sistema. Esta tarefa é extremamente complicada, devido à contínua liberação de novos Vírus e invenção de novas técnicas de "infecção". Esse tipo de abordagem faz com que a ferramenta antivírus só seja efetiva após a entrada de Vírus no sistema. Os programas de detecção se baseiam no princípio de que uma contaminação pode ser localizada e contida imediatamente após ter ocorrido. Os programas detectam o Vírus por meio das pistas deixadas por ele durante a invasão do sistema.

Consoante [WEBER, 1989, p. 113], as ferramentas de detecção procuram por efeitos característicos de ação dos Vírus, ou seja, elas verificam os programas executáveis, as áreas de *boot* do sistema, ou procuram por assinaturas de Vírus armazenadas em uma base de dados do antivírus ou verificam se a assinatura do programa ou arquivo (via *checksum*, CRC ou através de funções *hashing*) previamente calculada não foi alterada. Algumas soluções permitem que o exame seja efetuado sobre os discos, enquanto que outros examinam apenas os programas no momento em que são executados. Muitos programas de detecção analisam, ainda, as configurações de memória e das tabelas de interrupções do sistema de entrada e saída e sistema operacional, podendo, desse modo, detectar qualquer atividade potencialmente ilegal na memória ou em disco.

Como resultado, o processo de detecção pode incorrer em erros de dois tipos [POLK, 1995, p. 66]: falsos positivos e falsos negativos.

Quando a ferramenta de detecção identifica um arquivo não "infectado" como hospedeiro de um Vírus, temos a situação de um falso positivo. Em tais casos, um usuário iria gastar tempo e esforço desnecessários em procedimentos de identificação e limpeza. O usuário poderia substituir o arquivo supostamente "infectado" por uma cópia

limpa provinda do *backup* e a ferramenta de detecção continuaria a apontar o arquivo como "infectado". Isso poderia confundir o usuário e resultar em perda de confiança tanto nos procedimentos de detecção como na própria ferramenta antivírus.

Quando a ferramenta de detecção examina um arquivo "infectado" e incorretamente o considera como livre de Vírus, temos a situação conhecida como falso negativo. Este tipo de erro gera uma falsa sensação de segurança para o usuário e possibilita a ocorrência de possíveis desastres.

[LUDWIG, 2000, p. 273] discute três diferente técnicas que são utilizadas para localizar os Vírus. São elas: rastreamento, verificação de comportamento e verificação de integridade.

5.1.2.1. Rastreamento de Vírus

Rastreamento de Vírus é o método mais antigo e mais popular empregado para a localização de Vírus digitais. Os rastreadores têm uma importante vantagem frente a outros tipos de proteção, uma vez que permitem caçar o Vírus antes de sua execução. A idéia básica por trás do rastreamento é a verificação de conjuntos de *bytes* que são conhecidos como parte de um Vírus. Tipicamente, um rastreador conterá campos associados a cada registro com a seqüência de *bytes* a ser pesquisada, que diga onde procurar por tal seqüência. Essa capacidade de gerar uma seletividade e maior precisão, dispensa buscas por força bruta diminuindo a carga de trabalho no sistema alvo, fazendo com que o rastreador finalize o seu trabalho mais rapidamente.

[DENNING, 1991, p. 336] classifica os *softwares* de detecção por assinatura em:

- Detectores de Vírus Genéricos: procuram por seqüências no código dos arquivos que podem indicar um Vírus, incluindo leitura ou escrita para setores, instruções para formatação de discos, instruções de pesquisa em diretórios, nomes de arquivos ou seqüências suspeitas, etc;
- Detectores de Vírus Específicos: procuram por seqüências de código que são conhecidas e pertencentes a um determinado tipo de Vírus. Freqüentemente, tais assinaturas podem ser aquelas utilizadas pelos próprios Vírus para evitar uma "reinfecção";

- Assinaturas de Chamadas ao Sistema: monitoram chamadas adicionadas ao sistema por Vírus conhecidos, e geram alertas se houver uma detecção. Eles também podem retornar a seqüência correta de reconhecimento para prevenir uma "infecção" da memória do sistema pelo Vírus, exercendo, portanto, uma função de inoculação.

5.1.2.2. Verificação de Comportamento

Após os rastreadores, o tipo de tecnologia mais empregada por produtos antivírus na atualidade são os verificadores de comportamento. Esses verificadores de comportamento observam o sistema de um computador em busca de atividades classificadas como atividades típicas de Vírus digitais, e alerta o usuário quando isso acontece. Tipicamente, um verificador de comportamento é um programa residente em memória. Exemplos de comportamentos não usuais, que poderiam ser flagrados incluem: tentativas de abertura de arquivos de extensão COM ou EXE em modo de leitura e escrita, tentativas de escrever em setores de *boot* ou no MBR, ou tentativas de programa se tornar residente em memória. Geralmente, programas que observam estes tipos de comportamentos, o fazem através do controle das interrupções do sistema de entrada e saída (figura 11).

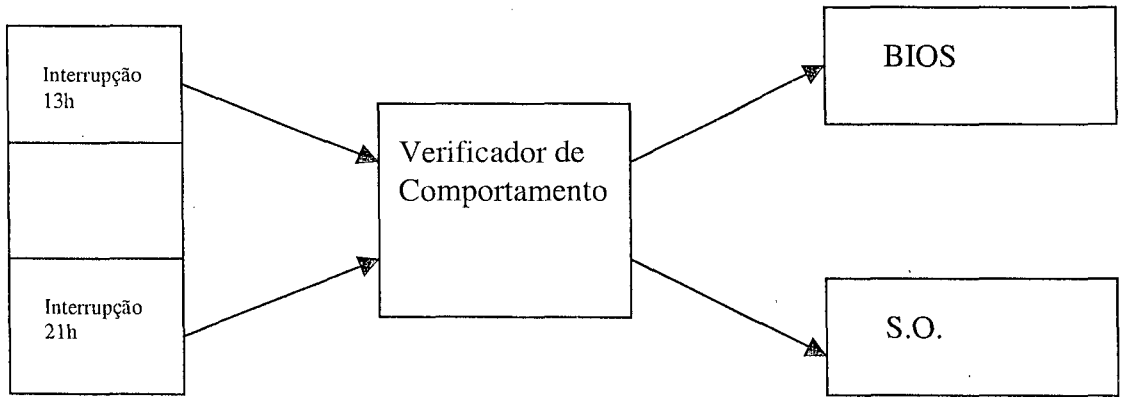

Figura 11: Acesso às interrupções com Verificação de Comportamento

A grande dificuldade com os verificadores de comportamento, segundo [DENNING, 1991, p. 336] é sua inabilidade de interceptar os acessos à BIOS

proferidos pelos Vírus de *boot*. Um Vírus de *boot* torna-se ativo antes do verificador de comportamento e, portanto, pode ele próprio fazer uma cópia do manipulador de endereços de interrupção da BIOS, ganhando a capacidade de enganar os verificadores de comportamento que analisam a tabela de interrupções (figura 12).

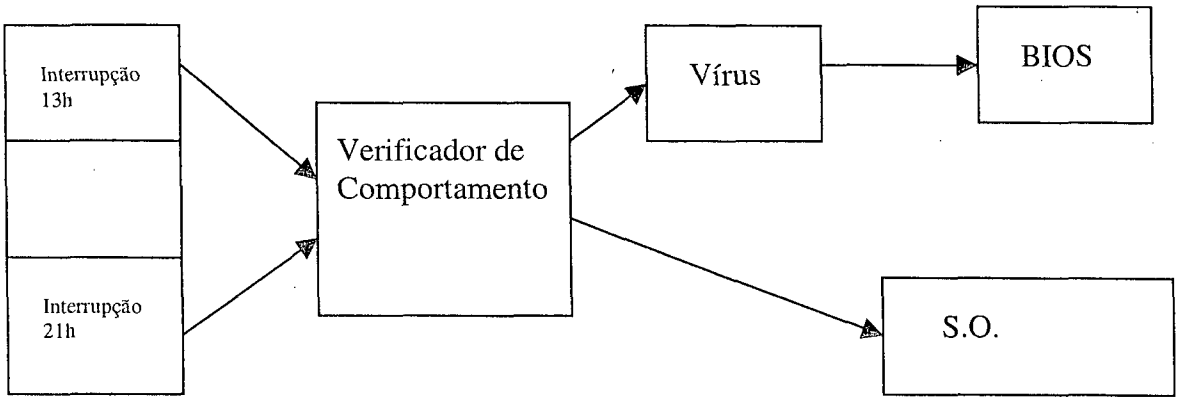

Figura 12: Interrupções e Verificação de Comportamento com Vírus de Boot

5.1.2.3. Verificação de Integridade

Geralmente, um verificador de integridade construirá uma base de dados contendo todos os nomes dos arquivos existentes no computador e alguma caracterização desses arquivos. Essa caracterização, ou assinatura conforme mencionado por [WEBER, 1989, p. 113], pode consistir de dados básicos a respeito do arquivo como o seu tamanho, data e horário de criação e última alteração, como bem pode ser um *checksum*, CRC, ou um *checksum* criptográfico de algum tipo. De acordo com [POLK, 1995, p. 20,21], os *checksums* via CRC (*Cyclic Redundance Checks*) que comumente são usados para verificação de integridade de pacotes em redes e outros tipos de comunicações entre computadores, são bastante eficientes e bem compreedidos, contudo não são extremamente seguros, pois eles são baseados em um conjunto conhecido de algoritmos e podem ser quebrados. Os *checksums* criptográficos são obtidos a partir da aplicação de um algoritmo criptográfico nos dados. Ambos, algoritmos simétricos e assimétricos podem ser usados. Em geral, os algoritmos de chaves simétricas são utilizados por questões de eficiência. Essas técnicas são algumas vezes usadas em conjunto com dois outros procedimentos, *message digesting* e *hashing*.

No uso de *message digesting*, *hashing* é usado em conjunto com *checksums* criptográficos. A função *hash*, geralmente muito rápida, é aplicada diretamente ao executável. O resultado é muito menor que os dados originais. O *checksum* é computado aplicando a função criptográfica ao resultado da função *hash*. Essa abordagem é muito mais segura.

Cada vez que o usuário executa o verificador de integridade, ele examina cada arquivo no sistema e o compara com a assinatura construída anteriormente. Um verificador de integridade capturará a maioria das mudanças feitas nos arquivos do computador, incluindo as modificações elaboradas pelos Vírus. Isto funciona, devido ao fato de que, se um Vírus se adiciona a um arquivo de programa, ele provavelmente o tornará maior e alterará sua assinatura. Então, presumidamente, um verificador de integridade notará que algo foi modificado, e gerará um alerta ao usuário do fato ocorrido e assim o usuário preparado poderá tomar alguma ação preventiva ou de contigência

Segundo [SARC c, 1998], um problema com o método de verificação de integridade, é que o mesmo gravará qualquer mudança nos arquivos, mesmo aquelas realizadas legitimamente pelos usuários. O usuário deverá lembrar de recomputar o arquivo de integridade assim que uma mudança legítima for efetuada, para que a mesma não seja relatada como ação de Vírus. Por essa razão, essa abordagem não funciona apropriadamente para uma das maiores ameaças da atualidade: os Vírus de macro. Em adição, essa abordagem apenas reporta mudanças feitas aos arquivos e não acusa qual Vírus causou a mudança, ou ao menos que tipo de Vírus.

5.1.3. Identificação

Uma vez que a detecção tenha sido executada, identifica-se o Vírus que propiciou a "infecção". As ferramentas de identificação, identificam qual Vírus "infectou" um programa ou arquivo em particular.

As ferramentas de identificação de vírus, fazendo uso do estudo desenvolvido por [SANKARY, 1989], tiveram suas origens nos programas utilitários desenvolvidos

em 1982 e 1983 para verificação de *software* de domínio público em busca de Bombas Lógicas e Cavalos de Tróia antes que eles fossem executados. Esses programas utilitários inicialmente verificavam instruções questionáveis no código objeto de programas suspeitos. Instruções diretas de entrada/saída, chamadas de interrupções, seqüências de formatação e instruções similares, se encontradas, eram sinalizadas e o usuário era notificado. Versões posteriores incluíam testes para verificação de seqüências de dados tipicamente utilizadas por projetistas de Cavalos de Tróia. Programas suspeitos eram analisados em busca de palavras chave, geralmente profanas, como *gotcha* ou *sucker*, e seqüências de dados que poderiam ser encontradas em programas Cavalos de Tróia específicos. Alguns programas eram observados, também, por nomes dos arquivos freqüentemente utilizados por Cavalos de Tróia e Bombas Lógicas. Essas ferramentas, contudo, eram raramente capazes de identificar uma Bomba Lógica ou um Cavalo de Tróia específico. Em outra mão, elas indicavam que o programa suspeito continha instruções ou mensagens de natureza questionável.

Isto, todavia, não era suficiente para lidar com os Vírus digitais. Os Vírus criaram problemas totalmente diferentes. Os Vírus se replicam e podem "infectar" centenas ou mesmo milhares de programas em um sistema. Eles podem permanecer invisíveis por longos períodos de tempo antes de serem ativados e podem causar danos. Além disso, são extremamente difíceis de remover, pois geralmente se instalam dentro de segmentos críticos de um sistema ou arquivos. Isso significa que não é suficiente saber se um Vírus está presente, sendo necessário saber qual Vírus está presente. É preciso saber qual o método de "infecção", quais ações ele toma e o que precisa ser feito para desativá-lo e removê-lo. Portanto, quando as primeiras ferramentas de identificação de Vírus emergiram em 1986, elas simplesmente não procuravam apenas por códigos ou mensagens genéricas, elas procuravam por indicações específicas que poderiam identificar Vírus individualizados. Isso permite ao usuário a verificação de ocorrência de "infecções" específicas e providenciar as ações apropriadas.

O processo de identificação parece ser uma tarefa simples, mas de fato não é. Existe um desacordo na comunidade de pesquisadores da área de antivírus, no que diz respeito ao que constituem Vírus diferentes. Como resultado, a comunidade tem tido problemas em aceitar o número de Vírus existentes, e os nomes associados à eles geralmente tem apenas uma vaga significância, gerando um problema de precisão.

Imagine um exemplo onde um microcomputador tenha duas ferramentas antivírus instaladas. A primeira ferramenta considera um conjunto de 350 Vírus existentes. Já a segunda, poderia considerar o mesmo conjunto como tendo mais de 900 Vírus. Isto ocorre pelo fato da primeira ferramenta agrupar um grande número de variantes sob um mesmo nome. A segunda ferramenta nomeia os Vírus de uma maneira mais detalhada. Tais problemas de precisão podem ocorrer mesmo se o fabricante da ferramenta antivírus tentar proceder a identificação com uma precisão mais alta. Uma ferramenta poderia identificar erroneamente um Vírus como outra variante de um Vírus existente por uma série de fatores. A variante poderia ser nova, ou a análise dos exemplos para a geração da identificação foi incompleta para uma perfeita identificação. A perda da precisão ocorre por diferentes razões, mas os resultados não são diferentes do exemplo apresentado. Qualquer identificação "bem sucedida" de um Vírus, precisa ser considerada junto com o grau de precisão.

5.1.4. Remoção

Com o Vírus detectado e identificado, procede-se a remoção de todos os seus vestígios, restaurando o sistema, levando-o ao seu estado original. Deve-se providenciar a remoção em todos os sistemas infectados, de forma a evitar a disseminação do Vírus.

As ferramentas de remoção tentam restaurar os arquivos "infectados" para o estado "não-infectado". A remoção é considerada bem sucedida, se o arquivo, após a "desinfecção", apresentar a mesma estrutura que o mesmo arquivo antes da "infecção", *byte* a *byte*.

O processo de remoção também pode apresentar dois tipos de falhas: *hard* e *soft*. Uma falha *hard*, ocorre se o programa "desinfectado" não mais executar ou se a remoção terminar sem remover o Vírus. Esta é uma falha grave, facilmente detectada e pode ocorrer por uma variedade de razões. Arquivos executáveis podem ser "infectados" por Vírus que os sobrescrevem, e os mesmos não podem ser removidos de forma automática, ao passo que muita informação pode ser perdida. Falhas *hard* também podem ocorrer se a ferramenta de remoção tentar remover um Vírus diferente do que realmente causou a "infecção".

No caso de uma falha *soft*, os resultados da remoção do Vírus podem modificar ligeiramente o arquivo frente a sua forma original, mas o mesmo continua podendo ser executado. Este arquivo executável modificado pode nunca apresentar problemas, mas o usuário não tem como ter certeza disso. A falha do tipo *soft*, é bem mais insidiosa, uma vez que o usuário geralmente não consegue detectá-la, sem a utilização de uma ferramenta de verificação de integridade.

Se a detecção for bem sucedida, mas a identificação ou a remoção não forem possíveis, então alternativamente descarta-se o arquivo infectado e recarrega-se uma versão livre de Vírus a partir do *backup*. Contudo, conforme [POLK, 1995, p. 66], Vírus e outras ameaças de *software*, podem permanecer em um sistema sem sua detecção de meses a anos, e portanto, poderiam estar incluídos no *backup* juntamente como os programas legítimos. Em tal situação, um programa poderia, inesperadamente, causar danos, e exigiria um grande trabalho de pesquisa nos arquivos dos *backups* realizados em busca de uma cópia não "infectada" do programa. Por isso sugere-se como um plano de contingência, no caso de programas, a restauração a partir da mídia original do fabricante do *software*. No caso de programas de dados, com informações acrescentadas pelos usuários ao longo de sua utilização, deve-se estar provido de ferramentas eficazes de remoção e mesmo assim, por vezes, é interessante uma verificação manual para garantir que os arquivos de dados estão realmente livres de Vírus e não apresentam nenhuma anormalidade.

5.2. A Evolução das Ferramentas Antivírus

Avanços em tecnologias de Vírus e antivírus são constantes. Anteriormente, Vírus eram relativamente simples, constituindo-se como simples fragmentos de código e poderiam ser identificados e eliminados através de simples pacotes antivírus. Contudo, como em uma corrida armada, ambos, Vírus e, necessariamente, ferramentas antivírus, têm se tornado mais sofisticados e complexos.

Segundo Stephenson, em seu artigo *"Preventive Medicine"* (LAN *Magazine*, *November* 1993), citado por [STALLINGS, 1999, p. 510], existem quatro gerações de software antivírus, a saber:

- ✓ Primeira geração: rastreadores/verificadores simples;
- ✓ Segunda geração: rastreadores/verificadores heurísticos;
- ✓ Terceira geração: verificadores de atividade;
- ✓ Quarta geração: proteção completa;

5.2.1. Primeira Geração

Um rastreador de Vírus de primeira geração requer uma assinatura do Vírus para corretamente detectá-lo e identificá-lo. Esses rastreadores baseados em assinaturas específicas são limitados a detectar apenas Vírus conhecidos. Um outro tipo de rastreador de primeira geração, mantém um registro do tamanho dos programas e verifica se os tamanhos originais são alterados.

5.2.2. Segunda Geração

Um rastreador de Vírus de segunda geração não se fundamenta na existência de assinaturas específicas. Eles usam regras heurísticas para pesquisar a probabilidade de uma "infecção" por Vírus. Uma classe de tais ferramentas observam por fragmentos de código que são freqüentemente associados com Vírus. Por exemplo, um rastreador poderia observar o início de um laço de criptografia usado em um Vírus polimórfico e descobrir a chave de encriptação. Uma vez que a chave seja descoberta, o rastreador pode desencriptar o Vírus para identificá-lo e se possível for, removê-lo.

Segundo [SARC c, 1998], heurísticas examinam os conteúdos dos arquivos e setores de *boot* em busca de características de Vírus, usando um conjunto de regras, ou inteligência artificial. Apesar deste método ser capaz de alertar o usuário da ocorrência de um Vírus antes que o mesmo dispare seu mecanismo "infeccioso", ele também tem a tendência de produzir falsos positivos. Na melhor das hipóteses, ele é capaz de acusar a presença de 75-85% de Vírus ainda desconhecidos.

Outra abordagem de segunda geração é a verificação de integridade. Um *checksum* pode ser anexado a cada programa. Se um Vírus infectar o programa sem

modificar o *checksum*, então um mecanismo de verificação de integridade irá capturar a mudança. Para conter um Vírus sofisticado o bastante para modificar o *checksum* quando ele "infecta" um programa, uma função *hash* pode ser utilizada. A chave de encriptação é armazenada separadamente do programa para evitar que o Vírus gere um novo código *hash*. Usando uma função *hash* em vez de um simples *checksum*, previne-se da situação de um Vírus ajustar o programa para produzir o mesmo código *hash* como antes.

5.2.3. Terceira Geração

Programas de terceira geração são aqueles que são residentes em memória e que identificam um Vírus através de suas ações, e não apenas baseando-se em sua estrutura em arquivos "infectados". Tais programas apresentam a vantagem de que não se faz necessário o desenvolvimento de assinaturas e regras heurísticas para uma grande variedade de Vírus. Na verdade, é apenas necessário a identificação de pequenos conjuntos de ações que indicam que uma tentativa de "infecção" está sendo feita e como intervir.

Alguns autores classificam as ferramentas com tais características como sendo programas que fornecem prevenção. Segundo [CIDALE, 1990, p. 39-41], os programas de prevenção geralmente permanecem residentes em memória RAM durante o uso do microcomputador. Eles acompanham todos os processos do sistema, de forma a alertar qualquer sinal de contaminação ou reprodução do Vírus. Esses programas filtram acessos a arquivos feitos por outros programas, acompanhando ainda, todos os programas que são executados e finalizados e todas as requisições ao sistema operacional. Examinam também as tabelas de alocação de arquivos e todas as estruturas de controle do sistema. Os programas antivírus ficam alertas à qualquer indicação de que um Vírus está tentando se infiltrar no sistema. Essas indicações geralmente acontecem associadas à tentativa de acesso a uma parte executável do sistema (segmento de *boot*, arquivos do sistema operacional) ou a um programa aplicativo. Quando isso acontece, o antivírus geralmente congela o sistema (antes que o Vírus cause a "infecção" e comece a se reproduzir) e apresenta uma mensagem na tela

sugerindo uma ação ou executa alguma ação de forma automática, previamente configurada.

Essas ferramentas de terceira geração podem apresentar um relativo grau de ocorrências de "falsos positivos".

5.2.4. Quarta Geração

Produtos de quarta geração são pacotes consistindo de uma variedade de técnicas antivírus atuando em conjunto. Estas técnicas incluem as já mencionadas nas gerações anteriores e em adição, incluem a capacidade de controle de acesso, limitando a habilidade dos Vírus penetrarem em um sistema e então limitam a habilidade de um Vírus atualizar arquivos e produzir "infecções".

Com os pacotes de quarta geração, uma estratégia de defesa mais abrangente passou a ser adotada, aumentando o escopo de defesa para medidas de segurança de um propósito mais geral.

6. VÍRUS EM AMBIENTES UNIX/LINUX

6.1. Introdução

Segundo [ANONIMO, 2000, p. 170], o ambiente Unix é um ambiente pobre para a disseminação de Vírus. O Unix emprega controle de acesso baseado em proprietários e grupos e controla os acessos de leitura, gravação e execução de arquivos.

Contudo, [COHEN, 1994, p.23], apresenta a real possibilidade de disseminação de Vírus em ambientes Unix, visto que muitas de suas pesquisas foram realizadas em ambientes multiusuário similares ao mundo Unix (MVS, VMS, etc.). Em ambientes Unix, quando um arquivo é "infectado" ele pode acabar "infectando" todos os arquivos a que tiver acesso. Os Vírus também poderiam residir em processos, "infectando" dados acessíveis aos processos. A disseminação que ocorre de usuário para usuário é sempre possível quando um usuário usa informações de propriedade de outros usuários, como no MVS. No MVS, por exemplo, se um usuário B executar algum programa "infectado" de um usuário A, os arquivos de B poderão ser "infectados", podendo contaminar, inclusive, tudo o que for passível de contaminação na área de B. Através desse processo, o Vírus pode se espalhar de usuário a usuário e eventualmente por todo o sistema. Todavia, em alguns casos, o problema pode se tornar ainda pior. Imagine que um usuário C possua autorização para modificar programas ou outras informações de uso comum, e que tal usuário executasse um programa "infectado", o que poderia gerar uma "infecção" nos programas de uso comum, que por sua vez, possibilitaria uma "epidemia" por todo o ambiente e áreas de usuários. No Unix, assim como no MVS, se um usuário A possui um programa "infectado", que em algum momento é utilizado por um usuário B, o usuário B, através do uso do programa de A, autoriza tal programa a modificar os seus programas. A presença de usuários privilegiados (como *root*), apenas possibilitam que o processo de disseminação seja acelerado, mas o processo seguirá os mesmos princípios.

É interessante ressaltar que, nos exemplos apresentados, um suposto Vírus não estava requerendo qualquer privilégio especial, não necessitou alterar o Sistema

Operacional, e também não dependia de qualquer falha ou defeito existente na implementação do Sistema Operacional, assim como não fez nada que não era permitido pelo sistema de segurança. O Vírus pôde se disseminar devido ao fato das políticas de proteção dos sistemas não visarem a integridade dos mesmos.

[LUDWIG, 2000, p. 253-259] menciona que diversas publicações apontam o desenvolvimento de Vírus para ambiente Unix como sendo geralmente classificado na categoria de impossível, mas, em outra mão, afirma que seu desenvolvimento não é mais difícil que para qualquer outro ambiente. Assim como o ambiente Windows é largamente utilizado como plataforma de uso pessoal, o que o torna alvo de "infecções" em massa, o Unix é o tipo de Sistema Operacional mais utilizado quando o assunto é conectividade com a Internet, o que o torna um alvo em potencial, haja visto a larga utilização da Internet, e a certeza, de em algum ponto na rede, os pacotes trafegarem através da utilização de sistemas baseados em Unix (Solaris, HP-UX, AIX, Linux, FeeBSD, etc).

Um dos agravantes do Unix, é que ele pode rodar em diversas plataformas de *hardware*, diferentemente com o que acontece com o Windows. De qualquer forma, é totalmente possível escrever Vírus em código *assembly* para Unix, muito embora tal código geralmente não seria portável entre as diversas plataformas de *hardware*. Para se escrever um vírus em *assembly*, deve-se conhecer e compreender a estrutura de um arquivo executável, bem como a linguagem *assembly* do processador alvo.

De outro modo, escrevendo-se um Vírus em C, o mesmo poderia ser executado em sistemas Unix de diferentes plataformas. Da mesma forma, se o mesmo fosse elaborado em *script shell*.

No artigo "*Computer Virus Awareness for UNIX*" da NCSA *News* Volume 3, de Maio/Junho de 1992, citado por [RADATTI a, 1996], foi exposto que uma das razões do fato de não se ter mais ataques em ambientes Unix, seria a situação de pouquíssimos programadores de Vírus terem condições de possuir ou obter acesso ao *hardware* necessário para a execução do sistema Unix. Contudo, isto não é mais verdade, visto que o Unix está amplamente disponível em Universidades e também em livrarias, por meio de mídias em livros ou revistas. O custo de estações Unix também já caiu bastante, chegando, em alguns casos, ao mesmo preço de plataformas PC. Além disso, o advento do FreeBSD, BSDI, Linux, SCO Unix, Solaris

para PC, etc., tem tornado o Unix ao acesso de todos, em função de sua gratuidade e da possibilidade de ser executados em máquinas de baixo custo, como um PC. Devido à popularidade da Internet, na qual o *backbone* e a maioria dos servidores são baseados em sistemas Unix, também ajudaram na sua disseminação, que hoje já é utilizado inclusive pelo usuário doméstico, em substituição ao Windows ou outro Sistema Operacional.

6.2. A "imunidade" dos sistemas Unix

Segundo [RADATTI b, 1996], sistemas Unix são susceptíveis a ataques de *software* hostis como qualquer outro Sistema Operacional, todavia, a comunidade Unix continua a defender a falsa idéia de que são imunes. Essa opinião é devido a uma realidade histórica. O primeiro Vírus de computador criado, em pesquisa e realmente reprodutivo, foi em ambiente Unix. Deve-se levar em consideração ainda, o episódio do *Internet Worm*, os Cavalos de Tróia, Bombas Lógicas, etc. Existe, atualmente, um número crescente de profissionais que trabalham com segurança computacional, que estão começando a se preocupar com essa questão. Essa preocupação é baseada no reconhecimento da natureza complexa do problema e do crescimento do valor das redes baseadas em Unix. Na época do acontecimento denominado *Internet Worm*, em 1988, o custo das conseqüências foi relativamente baixo. Se aquele ataque se repetisse hoje, as conseqüências e custos associados seriam muito maiores, haja visto a importância que a Internet passou a assumir, seja nos negócios eletrônicos usando EDI ou nas redes privadas, usando, de uma forma ou de outra, um ou mais sistemas baseados em Unix.

Tradicionalmente, métodos usados contra ataques em ambientes baseados em outros Sistemas Operacionais, tais como Windows, são insuficientes, devido à maior complexidade gerada por ambientes providos de sistemas Unix. Adicionalmente, o Unix provê um especial e significante problema, devido à sua natureza aberta e heterogênea.

O problema do ataque automatizado por meio de *software* existe em todos os Sistemas Operacionais. Esses ataques apresentam diferentes formas, conforme a função do ataque. Em geral, todas as formas de ataque contém um método de auto preservação, por meio de propagação ou migração, e uma carga, geralmente nociva. O método mais

comum de auto-preservação no Unix é a obscuridade. Se o programa possui um nome ou localização ocultos, ele poderá ficar livre de detecção até que sua carga tenha a oportunidade de ser executada. *Worms* se preservam através de migrações enquanto os Vírus de computador usam a propagação. Já os Cavalos de Tróia, Bombas Lógicas e de Tempo, se protegem através da obscuridade. Enquanto os algoritmos hostis que mais têm capturado a imaginação do público são os Vírus e os *Worms*, os problemas mais comuns em sistemas Unix são os Cavalos de Tróia e as Bombas de Tempo. Um Cavalo de Tróia, como já explanado em capítulos anteriores, é um programa que aparenta ser algo que não é. Um exemplo seria um programa que aparenta ser uma calculadora ou outro utilitário qualquer, mas que possui um código escondido que pode inserir uma *back door* no sistema alvo. Um simples Cavalo de Tróia pode ser criado modificando qualquer código com a adição de um código extra, com uma carga possivelmente nociva. Um dos códigos nocivos favoritos utilizados observados "in the wild" é: "/bin/rm –rf / >/dev/null 2>&1". Esse trecho de código tentará remover todos os arquivos acessíveis do sistema como um processo em *background* com todas as mensagems redirecionadas para /dev/null, ou seja, serem descartadas. Como vários sistemas apresentam um sistema de segurança fracamente instituído, existem normalmente milhares de arquivos com permissões completas de escrita, leitura e execução (777). Todos os arquivos em um sistema alvo com esse conjunto de permissões serão removidos através desse ataque. Adicionalmente, todos os arquivos com permissão de escrita, seja a nível de proprietário, grupo ou outros, serão também removidos.

A promoção do conceito de existir uma suposta imunidade em ambientes Unix, é perigoso e pode mascarar ameaças reais. Muitos defensores da existência de tal imunidade, baseiam-se principalmente no fato o Unix apresentar uma conta de supervisor (geralmente *root*), e conjuntos de permissões a arquivos e diretórios. A vida real tem provado a deficiência desses mecanismos de segurança no caso de atividades desenvolvidas por Vírus digitais. O fato é que existem Vírus que infectam sistemas Unix, sejam eles na forma de *scripts* ou como códigos executáveis.

Os ataques de Vírus e ameaças automatizadas de *software* contra sistemas Unix tornar-se-ão cada vez mais populares. Ao passar do tempo, os Vírus vêm apresentando um significante aumento no poder de disseminação, na adoção de métodos de auto-

proteção, e aproveitamento de oportunidades de "infecção". Os métodos de auto-proteção tem se tornado altamente refinados, incluindo a rápida reprodução através de "infecções", migrações com a avaliação de seus ambientes, aplicação de tecnologia *stealth* e polimorfismo. Além disso, os próprios sistemas alvos (hospedeiros) tornam-se parte do método de auto-proteção e propagação. Arquivos "infectados" por Vírus são protegidos pelo sistema operacional, da mesma forma que os arquivos não "infectados". A introdução dos Vírus em seus hospedeiros, tem sido bem refinada através da utilização da tecnologia denominada *droppers*. Um *dropper* é um Cavalo de Tróia que tem um Vírus ou mais de um como seu *payload*, ou seja, sua carga. Além do mais, tecnologias de redes extensivas como o NFS (*Network File System*), permitem a migração de Vírus entre sistemas sem esforço adicional.

Todas essas razões, apontam os Vírus como uma das principais formas de algoritmos classificados como hostis a serem utilizados massivamente no futuro, contudo, a razão mais significativa para tal conclusão é a eficácia dos elementos virais como uma forma de ataque. As experiências desenvolvidas pelo Doutor Frederick B. Cohen, ratificam tal afirmação, quando através do uso de uma conta normal (sem direitos privilegiados) em um sistema Unix, conseguiu total acesso em menos de 30 minutos. O Dr. Cohen repetiu tal experiência em diversas instalações e versões do Unix (incluindo vinte implementações comerciais) confirmando sua experiência inicial.

Diversos pesquisadores também chegaram a conclusões similares em seus experimentos. Tom Duff, citado por [RADATTI b, 1996], foi um deles. Ele demonstrou a tenacidade de Vírus em Unix mesmo quando da utilização de ferramentas de "desinfecção". O Vírus empregado por Tom Duff, foi um Vírus simples escrito em *script shell*. O Vírus era sistematicamente reintroduzido no sistema através de rotinas automatizadas de *backup* e *restore*. O Vírus "reinfectou" o sistema analisado, depois de um ano de sua detecção e eliminação original.

Sistemas Unix são muito utilizados também como servidores de arquivos via rede, podendo incubar Vírus em sua fase dormente, Vírus esses que poderiam não ter sido projetados para atacar sistemas Unix e sim outras plataformas. Isto pode tornar-se um problema muito grande, se não for identificado o foco das "infecções". Imagine um sistema Unix servindo arquivos em rede, através do NFS ou Samba, por exemplo, e com nenhum mecanismo de proteção contra Vírus instalado. As estações de trabalho e

PCs conectados a esse servidor de arquivos correriam sério risco de "infecção". Imagine, agora, que seja feita uma verificação de Vírus nessas estações e PCs e que se proceda a limpeza adequada. Não demoraria muito, e as "infeccções" poderiam voltar a acontecer, uma vez o foco das "infecções" seria o servidor de arquivos, rodando um sistema Unix, abrigando diversos Vírus em fase dormente.

No caso de sistemas Unix baseados em PC, existe a real possibilidade de Vírus projetados para uma plataforma atacar a outra. [RADATTI b, 1996] apresenta algumas situações baseadas em seus experimentos. Durante os anos de 1994 e 1995, ele desenvolveu suas observações. A primeira delas foi um ataque envolvendo um Vírus que corrompia o *file system* do Unix toda noite. O ataque foi localizado através da utilização de uma ferramenta de varredura (*scanner*) e indicou um binário Unix que era executado à meia noite pelo *cron*. O Vírus para MS-DOS se embutiu no executável do Unix, onde era executado. O Vírus não havia realizado as ações para as quais foi planejado, e os danos gerados no sistema de arquivos do Unix era resultado do Vírus "infectar" outros arquivos. O Vírus era reinstalado toda manhã após a restauração do sistema. O segundo ataque observado, envolveu um Vírus para MS-DOS que após sua execução, conseguiu "infectar" outros arquivos. Mais uma vez o sistema de arquivos foi corrompido, mas se manteve em funcionamento, permitindo a propagação do Vírus.

A "infecção" final envolveu um Vírus de setor de *boot*. Como este tipo de Vírus é executado antes da carga do Sistema Operacional, as diferenças entre o Linux e o Windows, por exemplo, passam a ser pequenas. O sistema básico de entrada e saída de um PC (BIOS), e os processadores são os mesmos em ambos os casos, e o Vírus é capaz de executar as ações para as quais foi planejado. Alguns casos foram verificados, segundo o estudo de [RADATTI b, 1996], sendo que em um dos casos uma empresa Multinacional foi envolvida e num cálculo estimado, considerando o gasto de tempo, desperdício de recursos e impacto nos negócios, os prejuízos ultrapassaram a casa dos milhões de dólares. Uma vez compreendido que as funções do processador e BIOS são as mesmas para ambos os Sistemas Operacionais, é muito fácil visualizar como os Vírus poderiam ser multiplataformas e serem projetados com essa intenção.

[RADATTI b, 1996] conclui sua abordagem afirmando que os ataques de *software* contra sistemas Unix continuarão a crescer. Os Vírus tornar-se-ão predominantes como forma de ataque a esses sistemas. Os Vírus multiplataforma

podem se tornar comum como um meio de ataque efetivo. Todos os métodos utilizados para a criação de Vírus para a plataforma Windows podem ser portados para o Unix. Isso inclui o desenvolvimento de ferramentas para a criação automatizada de Vírus. Com o advento das redes globais e crescente uso da Internet, os limites entre Vírus, *Worms*, e Cavalos de Tróia, tendem a ser reduzidos. Ataques serão criados usando todas essas tecnologias em conjunto e outras a serem desenvolvidas. Já existem, inclusive, ataques baseados no uso de ferramentas de auditoria e análise, como o COPS, SATAN e SAINT, entre outras, para a geração posterior de vírus que se aproveitem de falhas detectadas. A combinação de tais ferramentas, com os *kits* de construção de Vírus, poderia permitir um desenvolvimento funcional de vírus com alvos específicos. Com tantos problemas e possibilidades existentes, novos métodos de proteção precisam ser criados ou aprimorados. Pesquisas vem sendo desenvolvidas nesse sentido. Contudo, fica a certeza que os ambientes Unix devem permanecer em foco e serem incluídos na política de segurança e controle de Vírus, através da aplicação de metodologias e tecnologias pertinentes.

6.3. Danos e Principais Vetores de "Infecção"

A variedade de danos que são proporcionados pelas mais diversas categorias de *Malware*, é grande para qualquer ambiente em análise que apresente alguma espécie de vulnerabilidade, independentemente do Sistema Operacional, sendo em escala maior ou menor de acordo com os serviços e mecanismos de segurança implantados.

De acordo com [SPAFFORD, 1996], os danos que ameaças programadas podem provocar, variam desde absolutamente nada até conseqüências catastróficas, como a completa destruição de todos os dados em um sistema através de uma formatação de baixo nível. Os danos podem ser causados por seletiva eliminação de arquivos particulares, ou pequenas mudanças aleatórias. Muitas ameaças programadas podem objetivar alvos específicos, destruir aplicações em particular, ou simplesmente alterar uma certa base de dados para esconder evidências de qualquer outra atividade.

Revelação de informações é outro tipo de dano que pode ser resultado de ameaças automatizadas de *software*. Assim como simplesmente alterar informações em disco ou em memória, uma ameaça de *software* pode tornar alguma informação passível

de leitura, enviá-la como *e-mail*, postá-la em *sites* públicos, ou enviar para impressão. Entre essas informações, poderiam estar aquelas de caráter absolutamente sigiloso, como senhas de sistema, registros de empregados, etc. Ameaças de *software* também podem permitir acesso não autorizado ao sistema, e podem resultar na instalação de contas não autorizadas, troca de senhas, etc. O tipo de dano pode variar de acordo com as intenções de quem projetou o *Malware*.

Malwares (códigos maliciosos ou mal intencionados), também podem causar danos indiretos. Se, por exemplo, um *software* qualquer for disponibilizado no mercado contendo um Vírus ou uma Bomba Lógica, vários potenciais danos podem ocorrer. Certamente a reputação da corporação que produziu o *software* irá sofrer com tal episódio. A corporação também pode ter de arcar com prejuízos gerados aos seus clientes; licenças e selos de garantia usados com o *software*, poderão não proteger contra danos de tal esfera.

Não se pode saber, com exatidão, que danos (diretos ou indiretos) poderiam ser gerados por ataques automatizados. Se uma companhia não possui uma política de segurança bem definida ou seus empregados falharem na precaução necessária na preparação e distribuição de *software*, sérios riscos existem para que *Malwares* aproveitem-se da situação e gerem conseqüências desastrosas.

A questão mais importante que surge em uma discussão elvonvendo *Malware* é: como essas ameaças adentram em sistemas computacionais e se reproduzem? A maioria das *Back Doors*, Bombas Lógicas, Cavalos de Tróia e Bactérias aparecem nos sistemas por serem desenvolvidos nesses sistemas. Talvez, a maior ameaça à segurança de um sistema de computadores é seu prórprio grupo de usuários. Esses usuários compreendem o sistema, conhecem suas fraquezas, e sabem quais os mecanismos de auditoria e controle do sistema estão implantados. Freqüentemente, usuários legítimos têm acesso com privilégios suficientes para escrever e introduzir código de *software* mal intencionado para dentro do sistema. Ironicamente, há casos em muitas companhias em que a pessoa responsável pela segurança e controle da rede é a pessoa que poderia causar os maiores danos, e por vezes causa.

Usuários também podem ser agentes involutários na transmissão de Vírus, *Worms*, e outras ameaças. Eles podem instalar novos *softwares* de fontes externas, e instalar códigos mal intencionados embutidos ao mesmo tempo. *Software* obtido a partir

de domínio público tradicionalmente tem sido a origem da "infecção" de sistemas. Naturalmente, nem todo *software* de domínio público está contaminado; a maioria normalmente não está. Produtos comerciais também podem sofrer do mesmo mal. Dificuldades reais e críticas ocorrem quando empregados não compreendem os problemas em potencial que podem resultar da introdução de *software* que não passou por processos adequados de verificação. Tais *softwares*, podem incluir o paradigma "clique e *download*" dos *web browsers*.

Um terceiro vetor de "infecção" possível, segundo [SPAFFORD, 1996] , é o fato de uma máquina estar conectada em uma rede ou algum outro meio de comunicação entre computadores. Programas podem ser escritos externamente e "contaminar" o ambiente interno através da rede. Os *Worms* usualmente geram suas "infecções" por meio de redes interconectadas. *Worms* podem carregar Bombas Lógicas ou Vírus, introduzindo diversas ameaças e possíveis problemas ao mesmo tempo.

Ameaças programadas aproveitam-se de ambientes com baixo nível de controle, causado em parte devido a falta de treinamentos adequados de segurança e de especialistas na comunidade computacional. Pouquíssimos usuários apresentam um grau adequado de informação a respeito da segurança e possíveis ameaças e medidas de controle e contingência.

Não importando como a primeira "infecção" ou invasão ocorreu, o fato é que as coisas ficam geralmente piores quando o *software* mal intencionado se espalha através de todos os sistemas suscetíveis dentro de uma mesma organização ou planta. A maioria dos sistemas são configurados para estabelecer relações de confiança entre usuários, máquinas e serviços no ambiente local. Portanto, existem poucas restrições que venham a prevenir a disseminação de *software* mal intencionado dentro de uma rede local de computadores. Devido ao fato dos usuários de tais ambientes freqüentemente compartilharem recursos (incluindo programas, disquetes e mesmo computadores), a disseminação de código mal intencionado dentro de um ambiente assim é consideravelmente facilitada. A erradicação de *software* mal intencionado nesses ambientes é também mais difícil, devido à quase impossibilidade de identificação de todas as origens do problema assim como removê-las ao mesmo tempo.

7. MODELO DAS TRÍADES CONJUGADAS

7.1. Introdução

Através da exposição dos problemas relacionados a Vírus de computador e *Malware* em geral no decorrer da dissertação desenvolvida, percebe-se que são de uma complexidade e magnitude de notável relevância. A história dos elementos virais como protagonistas em ações que levam, em grande parte, a situações desastrosas e irreversíveis, não é tão recente como se poderia imaginar em uma primeira instância, visto que os casos passaram a ser noticiados e vinculados na imprensa apenas na segunda metade da década de 1980, mas suas primeiras ocorrências remontam à década de 1960 e ao início da década de 1970.

As ameaças automatizadas de *software*, atualmente referenciadas como *Malware*, e cujos Vírus digitais são seu principal expoente, são hoje a principal preocupação das empresas, organizações governamentais, militares e corporações em geral, além do indefeso usuário doméstico, figurando como a maior ameaça, real e constante, à segurança de sistemas informatizados.

Apesar de todo o aparato tecnológico, desenvolvido ao longo das duas últimas décadas, os sistemas e dados armazenados e processados em computadores continuam altamente suscetíveis às ações desses agentes e a desejada "imunidade" está longe de ser alcançada. Percebe-se, através do estudo apresentado nos capítulos anteriores, que as tecnologias desenvolvidas pelos *Malwares* são, na maioria das ocasiões, altamente eficazes e, além disso, bastante eficientes, provocando epidemias globais em questão de horas. Casos recentes podem ratificar essa afirmação, tais como o "Magistr", "Sircam", "CodeRed", "Nimda", entre outros. E o mais impressionante, é que na imensa maioria dos casos relatados, os ambientes alvos apresentavam uma arquitetura de proteção através da adoção de sistemas antivírus e eram considerados seguros. O problema é ainda maior, quando imaginamos e analisamos através do prisma de que as tecnologias antivirais são consideradas maduras e são utilizadas em mais de 80% dos ambientes corporativos industriais e 90% dos ambientes governamentais, sendo que esses dados são de 1994 [COHEN, 1994, p. 111], nos levando a acreditar que hoje, no ano de 2001,

a quase totalidade dos ambientes apresenta alguma ferramenta ou solução tecnológica antiviral implantada.

Começa a ficar claro que algo está errado, ou no mínimo, existem inúmeros equívocos na adoção de tais tecnologias, sendo que o maior deles é de que o uso de uma ferramenta antivírus possa defender completamente contra as ações de *Malwares*, desconsiderando-se outros aspectos essenciais para a implantação de um modelo de segurança efetivo.

[BONTCHEV, 1994, p. 3] considera que, de um ponto de vista técnico, os usuários precisam ser educados a não mais confiar essencialmente em um método ou tecnologia antiviral (exemplo: *scanning*). Pesquisas vem sendo desenvolvidas, no sentido de gerar melhores métodos de detecção de Vírus genéricos, como análises heurísticas.

7.1.1. Ambientes

Hoje, o ambiente operacional mais visado por ataques automatizados de *software* é o Windows da Microsoft. Alguns fatores poderiam ser apontados como causas desse foco, entre eles o de ser a plataforma de *software* mais utilizada no mundo, favorecendo a rápida disseminação de elementos com características reprodutivas, disseminativas ou de migração, e também o alto e freqüente número de vulnerabilidades encontradas nesse ambiente, as quais são constantemente utilizadas pelos *Malwares*, muitas vezes sem o conhecimento delas por parte do próprio produtor do *software*, no caso do Windows e demais aplicativos nele disponibilizados, a Microsoft.

Apesar dessa realidade, a verdade é de que os desenvolvedores de ferramentas antivírus focam seus produtos para tal plataforma, obviamente por ser um nicho de mercado mais proveitoso e lucrativo. Por outro lado, esse ambiente deveria apresentar então, uma proteção mais eficaz, visto que a oferta de *softwares* antivírus é substancial, gerando uma concorrência acirrada e com desenvolvimento de tecnologias cada vez mais sofisticadas. Mas a grande verdade é que tal ambiente ainda é um grande poço de ofertas para a disseminação e ataques de *Malwares*, os quais, em detrimento de todo o aparato tecnológico desenvolvido, sempre saem na frente (haja visto que a metodologia

empregada por soluções tecnológicas antivirais disponíveis no mercado, é, de uma forma ou de outra, essencialmente reativa).

De uma maneira mais imediata, uma conclusão se sobressai: se tecnologias de defesa existem e seu desenvolvimento é constante, e se através da implantação delas nunca se poderá chegar a um ambiente realmente seguro, o que se deve focar então não é apenas a segurança do ambiente, mas sobretudo o seu controle, através da adoção de metodologias que melhorem a eficácia das tecnologias a serem implantadas e seu gerenciamento, levando em consideração também outros aspectos, que não só o tecnológico.

E os ambientes que não apresentam ainda um aparato tecnológico suficiente para sua segurança, mas que também são alvos estratégicos na ação de ameaças automatizadas de *software*? O Unix de uma forma geral, e de uma maneira especial o Linux, seu expoente mais significativo nos dias de hoje, sofrem pela falta de tecnologias apropriadas para a defesa contra Vírus, *Worms*, Cavalos de Tróia e inúmeros outros *Malwares*. Até existem ferramentas, mas seu número é escasso e estão longe de serem realmente eficazes, considerando-se a complexidade do ambiente. Por outro lado, é importante frisar que o Unix apresenta características distintas do Windows, sendo mais robusto e seguro por natureza, desde sua concepção. Contudo isso é por vezes discutível. Casos recentes, reportados nos dois últimos anos (2000 e 2001), denunciam a problemática real e iminente de possíveis epidemias em ambientes baseados no Unix de uma forma geral. Citam-se, entre esses casos, o "Bliss", o "Ramen", o "Lion" e o "Cheese", todos com características interessantes e conjugando mais de um modelo de tecnologia de reprodução, disseminação, migração e/ou ataque (Vírus, *Worms*, Cavalos de Tróia, *Backdoors*, análise de vulnerabilidades, etc.).

O fato do Unix ser um sistema amplamente utilizado no que diz respeito a ser um meio de conectividade com a Internet, o torna, sem sombra de dúvida, um alvo estratégico para quem deseja criar *Malwares* que venham a causar danos em grande escala na Internet, principalmente com características de *Denial of Services*. Outro fator preocupante, quando a segurança é desconsiderada, é o do Unix ser também utilizado freqüentemente como servidor de arquivos e aplicações, usando protocolos como o NFS ou Samba, como estudado no capítulo anterior.

Conversando com Mark Ludwig [LUDWIG, 2000, p. 384], percebe-se que é praticamente impossível a adoção de um Sistema Operacional realmente seguro. A tendência, seja pelo legado de sua existência ou pela adoção de novas características e funcionalidades, é de que os Sistemas Operacionais se tornem cada vez mais complexos. Por vezes, a complexidade advém da própria implantação de mecanismos de segurança. Contudo, quanto maior for a complexidade de um sistema, maior será a probabilidade de existirem vulnerabilidades. [LUDWIG, 2000, p. 384] afirma que a segurança de um sistema apenas será considerada como uma questão absolutamente séria, quando o custo de um sistema inseguro for maior que o custo de um sistema não funcional.

7.1.2. Preocupações

Já em 1994, Vesselin Bontchev [BONTCHEV, 1994, p. 4], alertava para o significativo crescimento no desenvolvimento e disponibilização de pacotes de *software* para a construção de *Malwares*. Hoje, vivemos uma realidade em que é extremamente simples qualquer indivíduo, possuindo um acesso à Internet, adquirir uma ferramenta que permita o desenvolvimento rápido de vírus digitais com técnicas embutidas bastantes avançadas para a sua proliferação e auto-defesa. Isso faz com que o número de Vírus *in the wild* seja cada vez maior, dificultando as ações de antivírus que venham a utilizar técnicas tradicionais.

Em adição à simples existência dessas ferramentas, o crescente uso de técnicas avançadas, principalmente de forma cojugada, como o polimorfismo, técnicas *stealth*, análise de vulnerabilidades, *droppers*, etc., como apresentadas anteriormente, com rápida disseminação através da utilização de vetores adequados como a Internet, colaboram para uma crescente complexidade das tecnologias antivirais, as quais passam a exigir mais recursos, geram perda de desempenho em estações e servidores, apresentam custos mais elevados, e continuam não sendo eficazes.

Da mesma forma que os *Malwares*, as ferramentas de defesa, começam a apresentar soluções no uso de tecnologias conjugadas, usando uma estratégia de defesa em vários níveis. Na década de 1980 e 1990, havia o conceito de ferramentas especializadas ou na detecção, ou na identificação ou na remoção de vírus. Hoje o

emprego dessas abordagens em conjunto não é mais novidade, assim como o uso de monitores de atividade, verificadores de integridade, etc., de forma transparente aos usuários. Mais recente, é a adoção de camadas de proteção em um ambiente de rede.

Contudo, quando o foco na defesa é apenas tecnológico, os riscos são grandes, se consideradas as grandes possibilidades de falhas nos mecanismos implantados. [BONTCHEV, 1994, p. 15] aponta a real possibilidade de fraude na distribuição de programas antivírus, seja pela implantação de Cavalos de Tróia ou pela alteração (adulteração) nas bases de assinaturas, usadas para a detecção, fazendo com que a ferramenta seja incapaz de alertar a existência de determinados *Malwares*, mesmo estando com as últimas atualizações. A simples possibilidade de um mecanismo de verificação (*scanner*) ser "crackeado" gera um risco grave, visto que desenvolvedores de *Malwares* podem construir seus *softwares* de forma a serem capazes de ludibriar a verificação disparada pelo antivírus. Como exemplo real, recentemente foi divulgado que o mecanismo de atualização de bases de assinaturas de Vírus e outros *Malwares* via Internet disponibilizado pela Symantec (*Live Update*), era considerado inseguro pela própria Symantec, possibilitando que um atacante pudesse se passar por ela nas atualizações dos seus produtos.

E se fosse cogitada a venda de *Malwares*? Já se têm notícias do desenvolvimento de *Malwares* para serem usados em guerras. Isso já é uma realidade. A maioria dos *Malwares* encontrada, é geralmente atribuída à jovens desenvolvedores, que usam seu tempo livre em busca de desafio e promoção em seus grupos, ou devido a desvios de personalidade. Raramente é fruto de um projeto sério e com suporte financeiro. [LUDWIG, 2000, p. 390] alerta para o possível e iminente uso de *Malwares* em atividades terroristas, visto que o custo para o seu desenvolvimento é extremamente baixo se comparado com armas nucleares ou biológicas. É evidente que os resultados serão bem diferentes mas, quanto mais uma nação ou estado for dependente da tecnologia da informação, maior será o impacto. Percebe-se claramente, que tais *Malwares*, desenvolvidos profissionalmente, poderiam geram muito mais problemas que os já existentes *in the wild.*

Contra essas possíveis ameaças, [BONTCHEV, 1994, p. 18] sugere que todos os administradores de sistemas sejam educados e treinados para manterem os seus ambientes em um nível aceitável de segurança. Sempre que possível, o processo de

melhoria da segurança deveria ser automatizado, seja no uso de ferramentas adequadas, no uso de criptografia, etc. [CHESS, 1997] reforça a importância de se possuir um processo automatizado de defesa, reagindo rapidamente aos ataques e retro-alimentando o sistema de defesa.

7.2. Evolução Tecnológica

Frederich B. Cohen, no artigo *Implications of Computer Viruses*, de número 22 publicado em [DENNING, 1990, p. 381-406], aponta que o ideal seria a adoção das abordagens de Prevenção, Detecção e Cura (Remoção/"Limpeza") de forma combinada em um programa de defesa. Contudo, para a grande maioria das empresas, o custo seria considerado alto e possivelmente não tolerado, em detrimento da eficácia do programa.

Nesse mesmo artigo, Cohen, em 1990, apontava como uma estratégia futura de defesa, o foco na Integridade dos sistemas. Anunciou que a "proteção é algo que se faz, não algo que se compra", e que sistemas automatizados deveriam possuir garantias de integridade, visto que no campo da segurança da informação, a ignorância a esse respeito seria quase que suicídio.

Em 1994, [COHEN, 1994, p. 83-93], apresentava os *shells* de integridade, através da adoção de *checksum* criptográfico, com o uso de criptografia simétrica, como um mecanismo de defesa eficaz. Tomando emprestado as palavras de Cohen, um *Shell* de integridade detectaria todas as "infecções" primárias, e preveniria todas as "infecções" secundárias. Entretanto, foram levantadas algumas limitações, entre elas:

- Necessidade de um sistema de encriptação bom o bastante;
- O próprio mecanismo de verificação de integridade deveria ser seguro e confiável;
- A chave para encriptação deveria ser seguramente armazenada;
- A granularidade seria uma questão delicada: todos os arquivos possuiriam *checksums* criptográficos?;
- Como seria identificada se uma alteração é legítima ou não?

Atualmente, existem propostas de implementação de sistemas para a verificação de integridade desenvolvidas, inclusive, no Brasil. Uma delas é o SOFFIC (*Secure On-*

the-Fly File Integrity Checker), apresentado no 3º Simpósio Segurança em Informática, realizado em São José dos Campos, no Instituto Tecnológico de Aeronáutica, de 24 a 26 de outubro de 2001. O processo chave para a verificação de integridade de arquivo, de acordo com o artigo em questão publicado no simpósio [SERAFIM, 2001], é a comparação, através da geração de um *snapshot* do sistema de arquivos em um momento em que o mesmo é considerado íntegro. A partir de então, a comparação é feita periodicamente ou quando da suspeita de uma provável violação, comparando-se o estado atual com o *snapshot* anteriormente gerado. O SOFFIC emprega para a geração dos citados *snapshots*, o conceito do *hash* criptográfico, gerando uma assinatura de código de tamanho reduzido e fixo de bits, por meio dos algoritmos MD5 e SHA-1. "A assinatura de código proporciona ao usuário a segurança de um *software* lacrado, o que quer dizer que ela assegura autenticidade e integridade"[ROCHA, 2001].

O SOFFIC segue os princípios básicos introduzidos por [COHEN, 1994, p. 83-93], usando em contra-partida, a Criptografia Simétrica e Assimétrica, implementando uma série de controles para a sua própria segurança, entre eles, mecanismos de proteção de suas bases de dados, núcleo do sistema, *cache* e chaves usadas. Contudo, o SOFFIC, a mais dez anos depois das advertências de Frederick B. Cohen, é apenas ainda um protótipo, sendo que a primeira versão funcional, específica para o Linux, ainda não foi disponibilizada.

[MURILO, 2001] destaca a fragilidade da verificação de integridade por meio de *hashes* criptográficos (considerada por muitos especialistas o método mais seguro), quando da implementação de *Malware* e *Rootkits* (ferramentas utilizadas com freqüência por invasores para ocultar sua presença em máquinas comprometidas, fazendo parte inclusive de alguns *Worms* e ferramentas de DDOS – *Distributed Denial of Services*) como módulos de *kernel* (*Loadable Kernel Modules* ou LKMs). Um LKM é um componente que pode ser dinamicamente carregado no *kernel*, modificando a funcionalidade do sistema sem a necessidade de uma reinicialização. Os LKMs são implementados em vários sistemas Unix, entre eles o Linux, BSD, Solaris e HP-UX, e geralmente são usados para o carregamento de *device drivers* de maneira dinâmica. O administrador de sistema terá muitas dificuldades de detectar *rootkits* implementados como LKMs, mesmo com o uso de *hashes* criptográficos, visto que os comandos do

sistema não sofrerão modificação e normalmente as chamadas de sistema que permitem listar os módulos de *kernel* são alteradas.

O uso de assinaturas digitais também é sugerido por alguns pesquisadores para a proteção contra *Malwares*, seja diretamente ou indiretamente através do uso de infraestrutura de chave pública (PKI) e certificação digital para a distribuição de *software* e comunicação segura. No caso de *applets ActiveX* (linguagem de programação da Microsoft para criação de *applets* e DLLs – *Dynamic Link Libraries*), consoante [MCAFEE, 1997, p. 6], não existe formas de se garantir que a pessoa que assinou o *applet* é realmente quem diz ser. Em outra mão, não é possível confirmar se o *applet* é ou não hostil. É preciso confiar nas intenções do emissor e na integridade do *applet*.

Segundo [CHESS, 1997], as assinaturas digitais oferecem uma certa proteção contra Cavalos de Tróia enviados por partes não confiáveis, uma vez que o sistema não permitirá acesso a recursos importantes de autores não confiáveis. Oferecem menos proteção, contudo, contra Vírus e *Worms*. Devido ao fato das replicações ao longo de canais de comunicação existentes, as pessoas emissoras disseminadores desses *Malwares* muito provavelmente serão as mesmas presentes na base de dados de pessoas confiáveis. Os Vírus tendem a se disseminarem ao longo dos canais de comunicação estabelecidos, no curso normal dos negócios.

Independentemente da adoção de mecanismos de assinatura digital e de verificação de integridade, [COHEN, 1994, p. 57], como já estudado em capítulos anteriores, defende que defesas técnicas perfeitas, inibindo a disseminação de vírus em um sistema ou ambiente de rede, deveriam ser sustentadas na limitação do compartilhamento, na limitação da transitividade e na limitação do acesso e funcionalidade. No entanto, sugere como meios técnicos mais pragmáticos de defesa, a adoção de tecnologias usadas de forma conjugada, com um alto grau de sinergismo entre as tecnologias empregadas, adotando-se uma arquitetura em camadas. O *Moated Wall*, usado em castelos medievais, é uma possível analogia que pode ser feita, uma vez que combinava vários mecanismos de segurança, entre eles, muros, fossos, etc. No tocante à proteção contra *Malwares*, tecnologias como *shells* de integridade costumam funcionar bem em conjunto com mecanismos de controle de acesso.

Cohen conclui sua argumentação dizendo que qualquer defesa pode ser superada, mas o uso de uma arquitetura de defesa em camadas de forma sinérgica, provê

uma proteção mais efetiva. A redundância e tolerância a falhas devem ser levadas em consideração. Contudo, o fato é que tecnologias podem ser geradas, mas sempre apresentarão falhas e poderão estar susceptíveis.

A idéia de utilizar um sistema imunológico digital também é abordada em algumas iniciativas científicas. [CHESS, 1997], sugere que a reação de um sistema de defesa a qualquer tipo de ameaça, seja análoga à resposta de um sistema imunológico biológico a uma doença. [REIS, 2001], apresenta uma proposta para a modelagem de um Sistema de Segurança Imunológico de caráter mais genérico, defendendo que a analogia entre segurança de computadores e imunologia constitui uma rica fonte de inspiração para o desenvolvimento de novos mecanismos de defesa. Um esboço inicial do modelo pode ser visualizado na figura 13, logo abaixo:

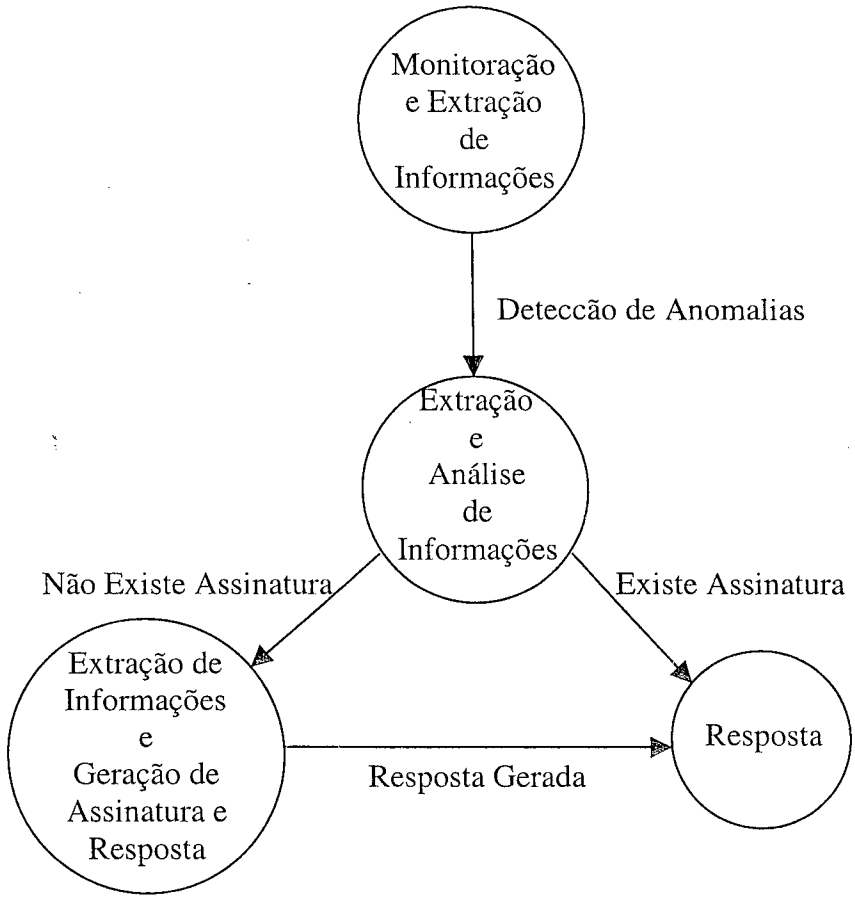

Figura 13: Modelo de funcionamento do Sistema de Segurança Imunológico

Esse sistema proposto por pesquisadores do Instituto de Computação da UNICAMP, possuiria três fases principais, a saber:

- Fase de detecção: quando o detector identifica alguma condição anormal do objeto monitorado, ativando o analisador, que iniciaria a identificação do ataque, acionando o extrator, que passaria a fornecer informações ao sistema adaptativo;
- Fase de ativação do sistema adaptativo: o analisador filtra as informações providas pelo extrator e busca por uma assinatura correspondente em sua base de dados. No caso da existência da assinatura, o analisador ativaria o agente de contenção para a execução da resposta. Caso contrário, o analisador ativaria o gerador de assinaturas para criá-la, com base nas informações obtidas pelo extrator. Em seguida, o gerador de assinaturas invocaria o gerador de respostas e passaria a lhe fornecer dados mais específicos, a serem usados na elaboração das medidas de contenção. Uma vez que os geradores concluírem suas tarefas, a base de dados seria atualizada e o agente de contenção seria acionado, pelo gerador de respostas, para combater o ataque;
- Fase de contra-ataque: o agente de contenção buscaria, na base de dados, as medidas relacionadas ao ataque e as executaria.

Embora esse modelo seja para uma segurança de forma mais genérica, não apenas para a defesa contra *Malwares*, percebe-se claramente sua aplicabilidade. Inicialmente a IBM, e agora a Symantec, desenvolveram um sistema imunológico digital, o *Digital Immune System*, para a defesa e combate a *Malwares* e de uma forma mais especial, aos Vírus digitais. O *Digital Immune System* apresentado pela Symantec Corporation [SYMANTEC, 2001], foi projetado como um sistema automatizado para tratar ameaças da classe do *Worm* Melissa, vírus em geral, e ataques de *Denial of Services*. Segundo [SYMANTEC, 2001, p. 4] o *Digital Immune System* apresenta, entre outras, as seguintes características:

- Detecta uma alta percentagem de ameaças, novas ou desconhecidas, no *desktop*, servidor ou *gateway*;
- Provê submissão segura de possíveis Vírus para análise e distribuição segura de novas definições;
- Provê um filtro inteligente de submissões, focando os recursos de sistema nas ameaças mais críticas;

- Reduz instâncias de falsos positivos;
- Provê total automação do início ao fim no que se refere à submissão, análise e distribuição de novas definições;

Uma visão macro do fluxo tratado pelo *Digital Immune System* proposto pela Symantec, pode ser melhor visualizado através da figura 14:

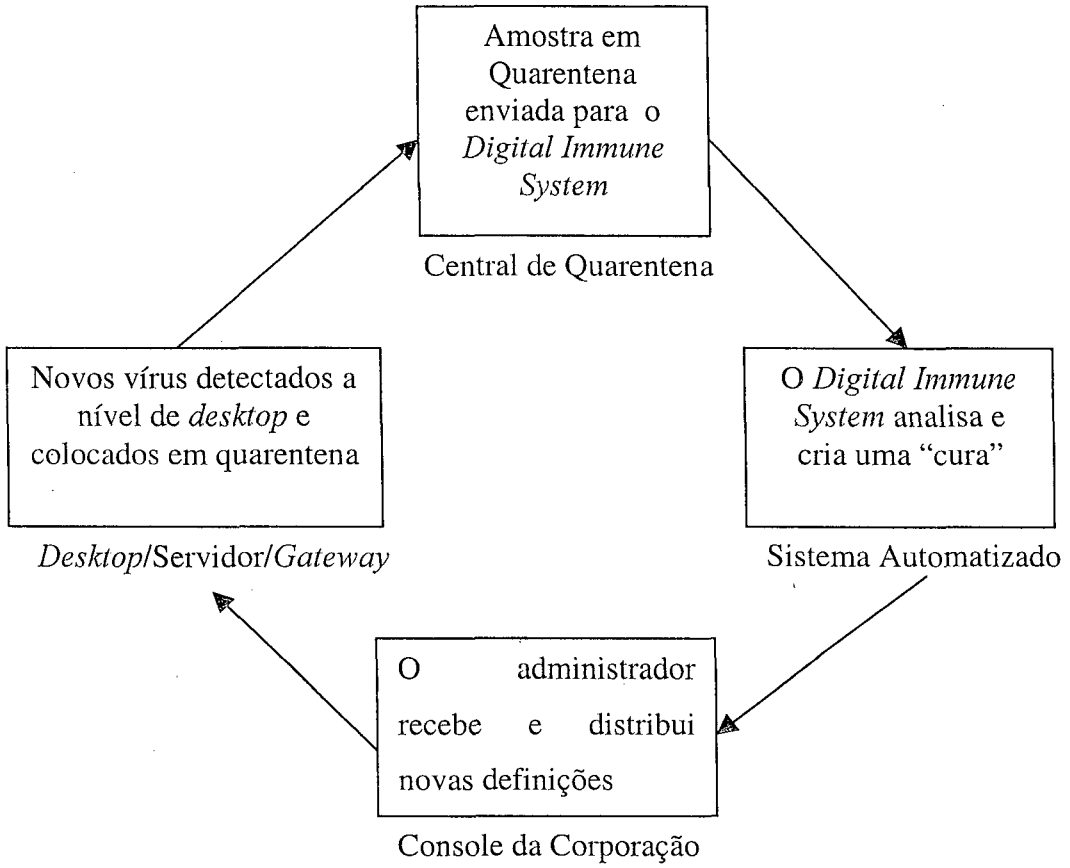

Figura 14: Diagrama Macro do Fluxo de Dados no *Digital Immune System*

Como conclusão, a Symantec coloca que o *Digital Immune System* automatiza todo o processo de criação de curas virais e sua distribuição, não dando oportunidade a erros humanos. Complementa afirmando que tal sistema pode gerar a cura para a rápida disseminação de Vírus em que o mundo se defronta atualmente.

Apesar da analogia ser bastante interessante e por vezes com alto grau de aplicabilidade, existem aspectos relacionados à analogia entre sistemas imunológicos, digitais e biológicos, que devem ser melhor analisados, para se estudar suas aplicações,

entre eles [REIS, 2001]: a) A definição de "sobrevivência" em um sistema computacional; b) É importante a adaptação do modelo a soluções "não-biológicas", pois muitas analogias não são pertinentes; c) A cobertura provida pelo sistema imunológico será sempre incompleta, apresentando vulnerabilidades a agentes "patogênicos" particulares.

Além das iniciativas e propostas de avanços tecnológicos apresentadas, podem ser citados muitos outros trabalhos, visando a defesa dos sistemas estudados e suas integridades, sendo desenvolvidos em ambientes de pesquisa, fundamentados em observações feitas no decorrer dos anos que se passaram ao advento nocivo dos Vírus digitais e demais *Malwares*, pela comunidade científica mundial, entre eles, inclusive, a proposta do uso de Redes Neurais para o reconhecimento de Vírus digitais [TESAURO, 1996].

Todavia, o que vêm a ser proposto nas próximas seções, em contrapartida, são modelos que possam vir a complementar o uso de tecnologias antivirais, com outros fatores não tecnológicos e, de uma forma mais genérica, serem utilizados de um modo pragmático e efetivo, independente da adoção ou não das tecnologias emergentes analisadas, buscando propiciar o melhor controle de um ambiente alvo e, conseqüentemente, torná-lo mais seguro.

7.3. Modelos

7.3.1. Introdução: Modelos Existentes Estudados

Embora as defesas com base tecnológica são o centro para o sucesso contra *Malwares* em ambientes modernos, segundo [COHEN, 1994, p. 97] a história tem mostrado que métodos não tecnológicos são efetivos para a proteção das informações. Qualquer sistema precisa ser utilizado por pessoas, e as pessoas precisam tomar decisões sobre quais métodos usar e como usá-los. [COHEN, 1994, p. 97] explica que na adoção de defesas não baseadas somente em tecnologias, são endereçadas políticas e procedimentos para o comportamento humano.

Jeannette Jarvis da Boeing Corporation, dos Estados Unidos, apresentou no congresso de 2001 do Vírus *Bulletin*, realizado no mês de setembro em Praga, uma estratégia denominada *Successful Anti-Virus Strategy*. Esta estratégia é baseada em quatro pontos fundamentais: produtos, processos, políticas e pessoas.

No tocante aos produtos, é ressaltada a importância da solidez das ferramentas antivírus, ou seja, sua confiabilidade, estabilidade e ausência de *bugs*. Os fornecedores, na opinião de Jeannette, precisam cooperar com os clientes na busca da satisfação de suas necessidades. Um exemplo citado, é a necessidade de gerar uma padronização nos nomes de identificação dos *Malwares*.

Em relação aos processos, é sugerido um gerenciamento de eventos sustentado por processos de gerenciamento predefinidos para momentos críticos, para conter rápida e efetivamente qualquer possível epidemia ou ataques em larga escala; todo mundo tem que saber o que exatamente fazer, como fazer e quando. Manuais claros e objetivos podem ser adotados gerando uma pró-atividade.

As Políticas são as práticas organizacionais documentadas e executadas por todos os envolvidos. É importante que primeiro, elas existam; segundo, sejam suportadas; e terceiro, que haja uma adesão incondicional a elas.

No tocante às Pessoas, pode-se analisar como sendo os pesquisadores, desenvolvedores, suporte técnico, equipe de vendas, etc., dos fornecedores de soluções antivirais, os clientes, seja no ambiente da empresa ou em casa, os consultores, etc. A relação de parceria, comunicação eficiente, troca de informações, comprometimento, disponibilidade, são alguns dos fatores extremamente importantes para o sucesso de uma estratégia antivírus, segundo Jeannette Jarvis.

[SOBERS, 2000], sugere a adoção de uma arquitetura antivírus de defesa, baseada em quatro camadas, o que ele chamou de 4-*Layered Approach*. As camadas seriam definidas da seguinte forma:

- Camada 1: o primeiro ponto de entrada em qualquer empresa, a partir da Internet, é o *gateway*. [SOBERS, 2000] sugere a implementação de algum tipo de *software* antivírus a ser aplicado no *gateway/firewall*, filtrando ativamente mensagens de correio eletrônico, *applets* Java e *ActiveX*;
- Camada 2: instalar *software* antivírus em todos os servidores de *e-mail* da organização, bloqueando extensões de arquivos anexos considerados de risco, como

por exemplo shs, vbs, exe, etc., e colocando em quarentena ou mesmo excluindo todos os arquivos anexos às mensagens que venham a ser detectados como hospedeiros de vírus;

- Camada 3: o terceiro ponto a se considerar, segundo [SOBERS, 2000], é o nível que combina servidores e estações *desktop*. Todos os servidores e estações devem possuir um antivírus instalado, com o modo de detecção em tempo real ativado. A nível de *desktop*, algumas ações adicionais podem ser tomadas para se assegurar uma melhor proteção, como a desativação de linguagens de script, como o *Windows Scripting Host*, além de se aplicar todos os *patchs* sugeridos pelos fornecedores, principalmente nos *softwares* que agem como clientes no serviço de correio eletrônico, principal canal de entrada de Vírus e *Malwares* na atualidade;
- Camada 4: de acordo com [SOBERS, 2000], é o nível mais importante e provavelmente o que gera menos custo. Trata-se da educação. Educação a nível técnico dos administradores de sistema, operacional das pessoas que venham de uma forma ou de outra a interagir com tecnologias antivirais, e educação a nível conceitual, esclarecendo os tipos de ameaça que a organização está sujeita, como ocorrem, como contê-las e reagir a elas.

[SOBERS, 2000] conclui sua abordagem, evidenciando que as disseminações virais continuarão a crescer, independentemente da adoção de tecnologias emergentes. A verdade, segundo suas palavras, é que os Vírus sempre farão parte da indústria computacional. Isso pode ser atribuído ao lado mal do ser humano. Nunca se chegará a uma conclusão definitiva das causas que levam as pessoas a desenvolverem Vírus. Todavia, [SOBERS, 2000] afirma que com a sua proposta das quatro camadas em ação, uma empresa terá tomado os passos necessários para formular uma defesa robusta que tornará a invasão de um *Malware* extremamente difícil.

[MACFEE, 1997] afirma que a melhor proteção para as redes corporativas é uma abordagem multi-camadas, onde uma camada seria a proteção a nível do *gateway/firewall*, e outra em cada computador. A segurança dos servidores viria a proteger a rede inteira a um custo efetivamente mais baixo. Produtos de segurança instalados em cada máquina de usuário provêm diversas dimensões adicionais de proteção. A proteção das estações de trabalho permite uma maior granularidade na

verificação do tráfego da Internet, do que no *firewall*; oferece ainda, proteção contra potenciais ameaças geradas pelos usuários, e protege contra intrusões provindas de ameaças internas, como Vírus sendo inadvertidamente disseminados por empregados dentro de uma organização.

Certamente, produtos como *software* antivírus são necessários no combate aos Vírus de computador e outros *Malwares*. Mas tomando emprestado as idéias expressadas em [TREND, 1997], os próprios fornecedores de antivírus concordam que, mesmo sendo uma importante parte no conjunto defensivo, os produtos sozinhos não resolvem o problema. As pessoas sim. E se as pessoas tiverem consciência da necessidade de proteção contra *Malware* e os benefícios proporcionados, muito provavelmente tomarão os devidos passos para a implantação de uma proteção efetiva. Definindo os passos específicos de uma forma operacional, têm-se os procedimentos, completando a tríade proposta pela Trend Micro, os três P's: produtos, pessoas e procedimentos. Através dos procedimentos, as pessoas podem fazer o melhor uso dos produtos disponíveis para alcançar os objetivos na proteção contra *malwares*. A figura 15, demonstra a importante distinção entre os procedimentos e toda a política corporativa, sugerida pela Trend.

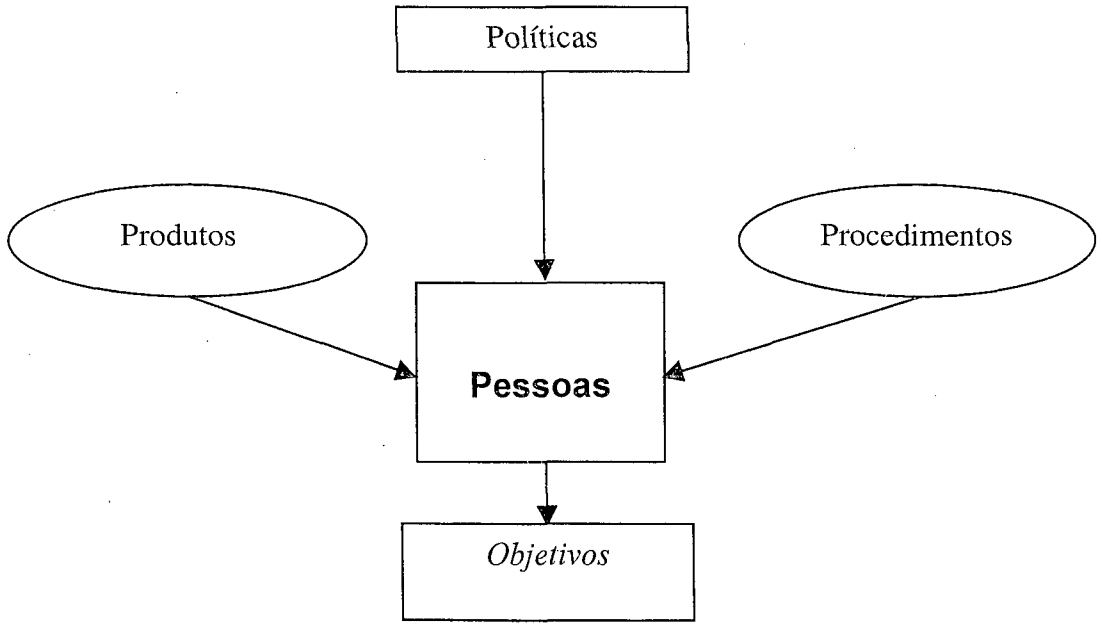

Figura 15: Proteção contra Vírus

Charles Cresson Wood [WOOD, 1997], citado por [TREND, 1997, p.7], define Políticas como "padrões de alto nível destinados a prover um guia para quem toma decisões... tipicamente incluindo declarações gerais das metas, objetivos, convicções, éticas, e responsabilidades". Wood também define procedimentos como "passos operacionais específicos que as pessoas precisam tomar para atingir as metas, geralmente esboçadas na política". Portanto, pode-se interpretar os procedimentos, apresentados em [TREND, 1997], como o modo de se implementar uma política.

Conversando com [GORDON, 1995], percebe-se que a proteção antivírus é, ou deveria ser, uma parte integrante de qualquer operação em sistemas de informação, seja pessoal ou profissional. Todavia, a observação mostra que o projeto de sistemas antivírus reais, bem como sua implementação e manutenção, pode variar de um esboço casual até algo praticamente não funcional. Sarah Gordon [GORDON, 1995], propõe idéias provindas da teoria de sistemas que possam ajudar a gerenciar melhor a segurança contra *Malwares*, ou ao menos identificar problemas antes não vislumbrados.

Na Teoria Geral dos Sistemas, se aprende que um sistema é um conjunto, ou grupo de elementos existentes relacionados formando um todo. Sistemas podem ser formados por objetos (exemplo: computadores), sujeitos (exemplo: empregados) e conceitos (exemplo: linguagem e comunicação). São sistemas reais, independentes de um observador, ou conceituais, sendo uma construção simbólica. [GORDON, 1995], enuncia que um sistema de estratégia antivírus , não é tão diferente de muitos outros, sendo composto de três elementos: computadores (objetos), pessoas (sujeitos) e conceitos (políticas e idéias). Cada sistema, possui seus próprios subsistemas. Por exemplo, um sistema de rede de computadores consiste de computadores individuais. Esses computadores são formados por mais subsistemas, como microprocessadores, resistores, etc. O sistema analisado consiste de subsistemas reais e conceituais. Conforme Gerald Weinberg [WEINBERG, 1975], citado por [GORDON, 1995], um sistema pode ser dito como um modo de ver o mundo, ou um ponto de vista.

Sarah Gordon afirma que, mesmo se for usado o melhor *software* antivírus disponível, rodando no máximo de suas capacidades, poderão ainda existir problemas. O *software* é simplesmente uma parte na estratégia. Políticas e procedimentos tomam um importante papel na estratégia como um todo. Mesmo os Vírus, fazem parte do sistema.

Sarah Gordon ainda complementa, em seu artigo, que não se pode antecipar todos os problemas que poderão ser encontrados ao se tentar manter uma organização segura contra a ação de Vírus, pelo fato de não se poder calcular as interações possíveis ao se tentar formular uma estratégia.

Usando uma abordagem modelada holisticamente (focando na "saúde" do sistema e sua manutenção), [GORDON, 1995] introduz a necessidade de se designar pessoas para garantir o sistema em um ótimo estado. O foco deve migrar da culpa para a responsabilidade. Isso requer investimento, equipamentos atualizados, empregados treinados, atualização tecnológica, manter os empregados sempre na vanguarda das informações relacionadas a ameaças virais e sistemas e metodologias de prevenção ou contenção. Além disso, precisa existir *feedback*, pois a falta de *feedback* proporciona entropia. Em termos simples, a entropia pode ser chamada de estado de degradação ou desorganização de uma sociedade ou sistema. A informação pode ser gerada a partir de qualquer tipo de comunicação entre qualquer elemento no sistema. É necessário examinar o sistema como um todo, incluindo todas as suas partes, seus relacionamentos e interações com elementos externos. No caso de um sistema de estratégia antivírus, precisa-se definir os elementos do sistema, suas fronteiras e metas.

[COHEN, 1994, p.146], esclarece que qualquer que seja o volume de informações capturadas e analisadas, elas não terão utilidade se não se souber aplicá-las. Em sua visão, existem dois componentes de planejamento críticos para uma defesa contra ameaças virais: planejamento estratégico e planejamento tático. As diferenças entre estratégias e táticas são comumente descritas em termos de tempo. O planejamento estratégico é elaborado para ações a longo prazo, enquanto que o planejamento tático é para curto prazo. O planejamento estratégico concentra-se em determinar que recursos estarão disponíveis e quais metas buscar-se-ão alcançar, sob diferentes circunstâncias, enquanto o planejamento tático concentra-se em como aplicar os recursos disponíveis para se alcançar as metas estabelecidas, em circunstâncias particulares.

Como princípios gerais, [COHEN, 1994, p.146-147], sustenta a idéia de que nenhuma simples defesa é segura em todas as situações, e nenhuma combinação possui um custo-benefício razoável em todos os ambientes. O princípio básico e fundamental é a consciência de que proteção é algo que se faz, e não simplesmente se compra.

No núcleo de qualquer estratégia está o time de pessoas que a desenvolve, a conduz, a longo prazo e prevendo situações táticas. Portanto, a primeira e mais importante coisa ao se definir uma estratégia, é montar um bom time para ajudar a implementá-la.

Uma estratégia antivírus sempre deve fazer parte de uma estratégia mais ampla, não devendo ser isolada. É comum se possuir um time de pessoas na ação contra Vírus que não exerçam nenhuma influência nas decisões de compra de *hardware* e *software*. Uma simples diferença no planejamento geral pode gerar dramáticas diferenças nos custos associados às defesas contra Vírus.

Respostas eficientes em situações táticas, ainda tomando emprestado as idéias de [COHEN, 1994, p.148-149], normalmente endereçam a completa atenção de especialistas treinados exercendo funções bem definidas. Adicionalmente, é comum a requisição de um número substancial de não-especialistas para que acompanhem os especialistas, realizando operações assistenciais. Situações táticas requerem que recursos suficientes estejam disponíveis para lidar com elas. Se não houver recursos suficientes, o custo gerado para a adoção dos recursos necessários pela base de emergência geralmente é bastante alto, gerando atrasos e problemas operacionais que podem ser causados pela inexperiência na operação de tais recursos. Por outro lado, uma pequena quantidade de recursos devidamente utilizados, geralmente supera funcionalmente, a utilização de vasta gama de recursos mal aplicados. Não é fundamentalmente necessário a geração de defesas robustas e caras, mas é fundamental e mais sábio, em situações táticas, investir equilibradamente e reagir com rapidez.

7.3.2. Modelo das Tríades Conjugadas

Através do estudo das tecnologias e metodologias apresentadas anteriormente, reflexão sobre os mesmos e uma experiência profissional desenvolvida ao longo de mais de dois anos implantando soluções antivírus (tecnologias) no mercado do sul do Brasil, desenvolveu-se um modelo genérico para a aplicação nas mais diversas circunstâncias e ambientes, sendo que seu uso independe da escolha por esta ou aquela tecnologia em específico.

Uma grande dimensão do problema, já dizia [COHEN, 1994, p.110], é uma postura pró-ativa versus reativa. Historicamente, as defesas de caráter reativo vêm sendo amplamente empregadas na segurança computacional, e as tecnologias antivirais, de uma forma ou de outra, apresentam esta característica. Um dos maiores problemas relacionado às defesas reativas é que elas dependem do poder da percepção humana, por vezes tarde demais. Além disso, a maior desvantagem de defesas de caráter reativo é que tomam muito tempo, e não se permite antecipar os problemas não havendo preparação prévia contra os mesmos. Uma das "vantagens" é que seu custo é baixo, enquanto não ocorrer nenhum ataque.

Os modelos estudados, apresentam-se bem estruturados, mas em alguns aspectos, deixam de contemplar pontos importantes no sucesso para suas aplicações. A *Successful Anti-Virus Strategy*, peca quando deixa de dar foco em um ponto extremamente importante, que é a educação das pessoas, o que pode resolver boa parte dos problemas. A 4-*Layered Approach*, apresenta uma abordagem interessante no tocante à organização tecnológica, e cita a educação como o principal fator, mas deixa de expor outros pontos trabalhados em outras propostas. Sarah Gordon, introduz uma visão sistêmica, bastante pertinente, análise que vem a alimentar o fluxo de diversos modelos e metodologias. O modelo proposto pela Trend Micro, não apresenta muitas novidades, e mais uma vez falta o foco na educação e na atualização das informações, como item importante para a manutenção e sobrevivência do modelo. Cohen, também apresenta uma análise importante, na adoção de estratégias e táticas, que vêm a complementar análises desenvolvidas nos modelos estudados.

O modelo aqui proposto, busca a pró-atividade na defesa contra *Malwares*, através da conjugação de diversos elementos cuja importância foi delineada no decorrer do documento. Os elementos básicos do modelo, aqui denominado "Modelo das Tríades Conjugadas", são seis, a saber: Pessoas, Educação, Informação, Tecnologia, Política e Processos, complementando soluções anteriormente sugeridas. Em princípio duas tríades básicas, são concebidas:

7.3.2.1. Primeira Tríade: foco nas Pessoas

A primeira tríade é constituída pelos seguintes elementos (figura 16):

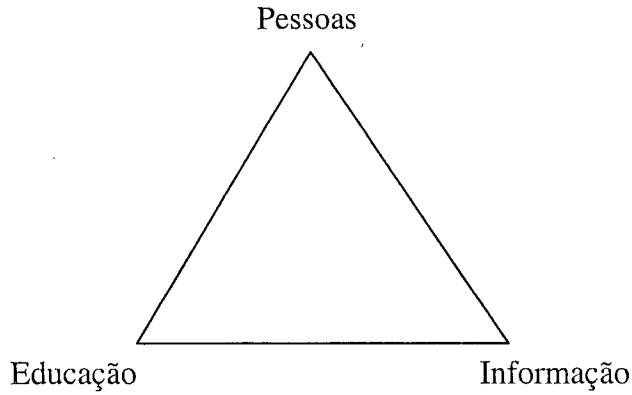

Figura 16: Primeira Tríade

- Pessoas: todas as pessoas envolvidas, direta ou indiretamente, dentro da organização ou em parceiros de negócios e fornecedores, que possam vir a ser contemplados no sistema de defesa. O usuário normal, merece uma atenção especial, visto ser um agente importante na proliferação ou não de ameaças automatizadas de *software*. A preparação de uma equipe de atendimento de emergência, formada por especialistas bem treinados e preparados, é essencial na implantação desse modelo como base para a defesa de um ou mais ambientes. Pessoas para atividades de suporte também são necessárias para ajudar os usuários no desenvolvimento de seus trabalhos e contribuir para a detecção de qualquer eventual anomalia;
- Educação: a educação é uma questão chave quando se envolve pessoas no processo como um todo. Pessoas bem treinadas e esclarecidas, contribuem diretamente para a eficácia e também eficiência de qualquer sistema de defesa ou estratégia a ser adotada. Pode-se desenvolver a educação em diversos níveis: operacional, técnico, administrativo, gerencial, ou outros níveis que venham a ser introduzidos. O treinamento da equipe de suporte e especialistas de segundo nível (equipe de emergência) deve ser constante. A conscientização periódica de todas as pessoas envolvidas, com relação às ameaças existentes e defesas implantadas é essencial para a sobrevivência do modelo;

- Informação: inicialmente as pessoas devem ser treinadas e educadas, sendo a educação um processo contínuo. A informação exerce papel fundamental pois é com base nas informações que se dispõe é que serão preparadas as pessoas. As pessoas da equipe emergencial devem constantemente buscar novas informações sobre ameaças e ferramentas de defesa e controle, repassando tais informações a nível técnico para a equipe de suporte e a um nível mais apropriado , às demais pessoas. A questão chave é a vanguarda e atualização da informação. Toda a organização deve estar constantemente conscientizada a respeito das mais novas ameaças existentes e como se defender contra elas.

7.3.2.2. Segunda Tríade: foco na Tecnologia

A segunda tríade é constituída pelos seguintes elementos (figura 17):

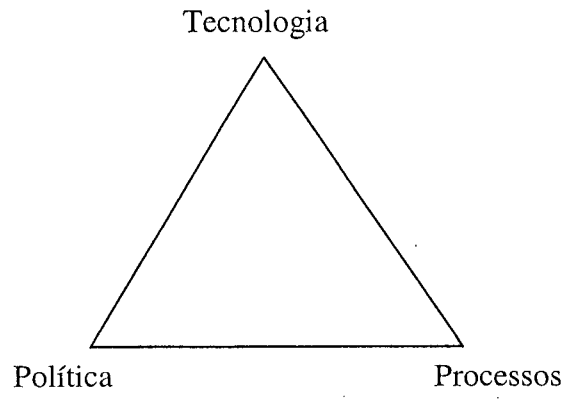

Figura 17: Segunda Tríade

- Tecnologia: toda ferramenta de *hardware* ou *software* a ser adotada como meio a auxiliar no sistema de defesa é englobada pelo elemento Tecnologia. Os meios tecnológicos contra *Malwares* não precisam ser essencialmente ferramentas antivírus. Pode ser uma conjugação de diversos tipos de *software* que venham, de uma forma ou de outra, a auxiliar na automação do processo de defesa. Podem ser citados, além dos antivírus, ferramentas para verificação de integridade, controle de

acesso, ferramentas para limitações de compartilhamento e organização funcional baseadas em regras e perfis de usuários, etc. As ferramentas para automação do gerenciamento das tecnologias de defesa, também são bastante importantes, pois propiciam um maior e melhor controle da situação, com foco na segurança. Quanto mais automatizado for o processo, melhor, pois diminui as possibilidades de erro;

- Política: todas as práticas desenvolvidas para a segurança dos ambientes da organização, devem ser documentadas. É importante frisar que, uma política documentada, mas não praticada, não passa de um mero documento. Tudo que estiver documentado em uma política de segurança, deve refletir o dia a dia da organização em relação aos procedimentos de segurança adotados. A política deve focar o aspecto estratégico em um sistema de segurança amplo a ser adotado. Deve deixar claro as metas e, em linhas gerais, como alcançá-las, vinculando-se a procedimentos táticos e operacionais. Como já mencionado anteriormente, também deve definir responsabilidades e, é importante destacar, que o foco seja no comprometimento dos indivíduos da organização e não na intimidação dos mesmos, evitando-se a geração de sentimentos de culpa no descumprimento de uma ou outra orientação, o que poderia levar à não funcionalidade da política e, como já exposto anteriormente, tornando-a um mero documento;
- Processos: os processos englobam procedimentos tático-operacionais bem definidos, ou seja, os já mencionados passos operacionais específicos que as pessoas precisam tomar para atingir as metas, geralmente esboçadas na política. Os processos precisam ser padronizados, e todas as pessoas que vierem a ser classificadas como passíveis de executá-los, assim o devem fazer. Portanto, deve haver um conjunto de pré-condições para que os processos sejam executáveis, como se fossem insumos ou entradas necessárias para os processos em específico. Por exemplo, uma pessoa sem o devido treinamento, não poderia ser considerada apta para a limpeza manual de arquivos "infectados" por um determinado Vírus. Os procedimentos, devem ser auditados e continuamente melhorados, sendo que o *feed-back* de quem os executa é fundamental para o sucesso dos mesmos. Os procedimentos que focam as tecnologias empregadas são extremamente importantes para ações rápidas, e adequadamente desenvolvidas para as ocasiões em cheque. A base de processos, pode crescer substancialmente, e seu gerenciamento e controle também é essencial

para o funcionamento do modelo. Deve-se haver a consciência de que, também, muitas vezes os processos podem gerar retro-alimentação da política, mudando rumos estratégicos, podendo também influenciar decisivamente na adoção ou não de determinadas tecnologias;

7.3.2.3. A conjugação das Tríades: o modelo final

Através da conjugação da primeira tríade, sustentada nos pilares Pessoas, Educação e Informação com a segunda tríade, sustentada nos pilares Tecnologia, Política e Processos, chega-se ao modelo final proposto, pela interligação de todos os pilares, sobrepondo suas representações gráficas triangulares (figura 18):

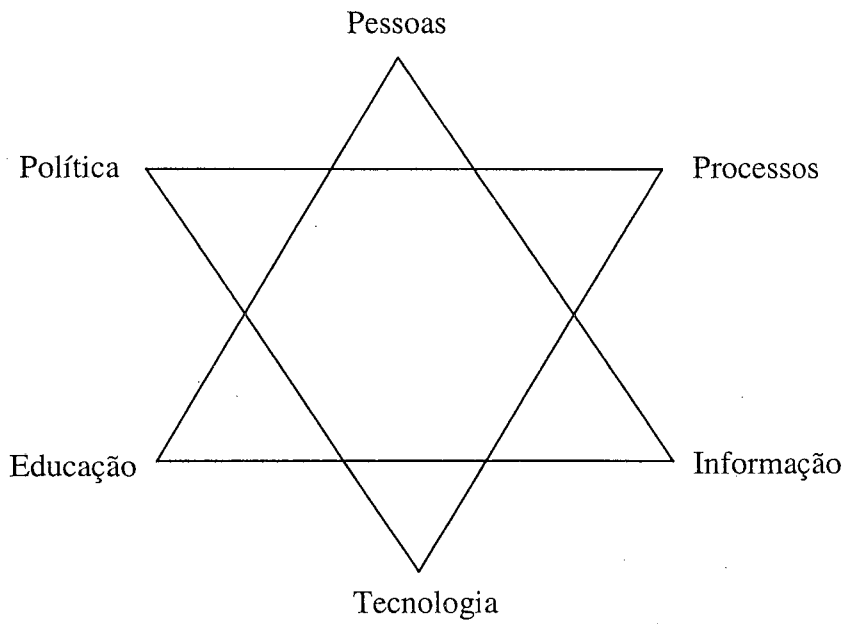

Figura 18: Modelo das Tríades Conjugadas

Todos os elementos devem estar interligados na geração de um sistema de defesa efetivo e permanente, baseado em um planejamento estratégico sólido e sustentado por planejamento tático claro, objetivo e de fácil implementação.

Os elementos do modelo proposto interagem naturalmente em qualquer situação. O que é fundamental para seu efetivo funcionamento, é o planejamento das interações entre os pontos relacionados, de forma a criar uma conjuntura sinérgica e colaborativa, fundamentadas em alto grau de controle e gerenciamento.

Para a implantação de qualquer Tecnologia, são necessárias Pessoas, que de uma forma ideal deveriam estar bem treinadas através de um sólido programa de Educação, estando também atualizadas na vanguarda da Informação, executando Processos bem definidos e previstos em uma Política bem documentada e institucionalizada, com metas e objetivos claramente apresentados.

7.3.3. Discussão e Análise do Modelo Proposto

No tocante à Política e Processos a serem adotados, na implantação de um planejamento estratégico e tático contra *Malwares*, deve-se ressaltar alguns pontos.

Primeiramente, uma política que venha a versar sobre um planejamento estratégico na defesa contra *Malwares*, deve fazer parte de uma política mais ampla de segurança, abrangendo outros aspectos que não as ameaças automatizadas de *software*, como intrusões, espionagem, controle de acesso, etc. A definição da política no tocante aos Vírus e *Malwares* deve "casar" com os demais itens de uma política de segurança mais ampla de forma coerente e sinérgica. As metas e objetivos propostos também devem ser coerentes com aqueles propostos na política de segurança da organização de um modo geral.

Segundo, a política a ser implantada, depende estritamente dos interesses da organização. É algo bem particular e que deve ser confeccionada de um modo pragmático, ou seja, uma vez sendo adotada, seja realmente seguida e praticada. Pode-se, obviamente, recorrer a modelos já existentes, mas sempre será necessária uma adaptação. A NBR-ISO-IEC 17799, baseada na norma inglesa BS-7799, liberada no Brasil a partir de setembro de 2001, é uma norma para a implementação de políticas de segurança globais, incluindo a tratativa no controle e combate aos *Malwares*, que geralmente é seguida por boa parte das organizações, sendo amplamente aceita, dando margem a auditorias e futuras certificações de conformidade.

Terceiro, da mesma forma que a Política, também os Processos são singulares a cada corporação em particular, uma vez que estão diretamente relacionados à Política de Segurança implantada. Também neste caso, pode-se recorrer a alguns modelos de implantação, seja na formatação e padronização do modelo dos procedimentos a serem elaborados, bem como os próprios procedimentos. [COHEN, 1994, p. 149-150] sugere algumas linhas gerais para a elaboração de procedimentos geralmente eficazes em boa parte das organizações onde foram implantados. Entre elas, sugere procedimentos para resposta rápida a incidentes de uma forma centralizada, através da adoção de equipes de emergência bem preparadas. Sugere ainda, a captura e análise de registros de incidentes, melhorando o planejamento tático propiciando reduções de custos futuros, por meio da melhor adaptação dos procedimentos às ocorrências a serem enfrentadas. Adicionalmente, propõe a adoção de procedimentos alternativos no caso de falha nos procedimentos principais ou na execução dos mesmos. Procedimentos de contingência devem sempre fazer parte do planejamento tático.

Outra fonte interessante de consulta para se elaborar bons procedimentos, é a ICSA. A ICSA disponibiliza guias para a elaboração de procedimentos táticos e operacionais, como em [ICSA, 1999], onde são contempladas diversas situações, ambientes e aplicações, sendo um modelo que também pode ser útil no fornecimento de subsídios para a implantação da política de segurança

Em relação à Educação, é importante relevar que não é um processo barato, e deve-se sempre analisar a relação custo-benefício e também escolher bem as pessoas envolvidas. De uma forma geral, todas as pessoas devem ser educadas contra as ameaças dos *Malwares*, cada uma ao nível adequado. Os treinamentos técnicos costumam ser dispendiosos, e por isso, as pessoas a serem preparadas, tendo em vista o importante papel que exercerão, devem ser bem selecionadas, com base em seus princípios éticos e capacidade técnica, além da sua qualificação e talento para resolverem problemas de alta complexidade, sendo exigido um alto grau de comprometimento. Isto, com certeza, exigirá desembolso de maiores salários e melhores condições de trabalho.

A experiência mostra que para um conjunto de 300 usuários em ambientes corporativos, é altamente recomendado a existência de pelo menos um profissional qualificado para, de forma integral e com dedicação exclusiva, administrar o ambiente

contra a ameaça dos *Malwares*. Este profissional deverá planejar a adoção de tecnologias adequadas, administrá-las, manter-se atualizado na vanguarda da informação e disponibilizá-las de um modo eficiente, planejar a sua educação e a educação necessária aos usuários, elaborar os processos pertinentes e administrá-los bem como auditar suas execuções periodicamente, enfim, diversas atividades extremamente importantes. Pode-se fazer um grosso planejamento de necessidade de mão de obra, como se segue:

Tabela 1: Sugestão para volume de Mão de Obra

Número de Usuários	Número de Profissionais (Time de Emergência)
Até 300	01
Até 1000	02
Até 2500	03
Acima de 2500	04

Adicionalmente, ao número de profissionais especialistas que formarão o time de emergência, é interessante formar uma equipe de suporte técnico, que viria a prestar suporte de primeiro nível a qualquer situação eventual. Esta equipe não necessitará de dedicação exclusiva. Vale frisar, que abaixo de 300 usuários existem inúmeras empresas. Muitas delas não chegam a 50 usuários, e a colocação de um profissional dedicado pode ser além de dispendiosa, desnecessária. Cada organização, pequena ou grande, é que tem que avaliar quantos profissionais utilizará. Os números apresentados são apenas uma sugestão, que por muitas vezes poderão estar além das expectativas e necessidades ou mesmo aquém. O que se pretende mostrar, é que o modelo para ser eficaz precisa de pessoas para implementá-lo e administrá-lo, em uma evidente postura pró-ativa.

No tocante às Informações, é imprescindível que se proceda consultas regulares em freqüências no mínimo diárias, aos meios adequados, como os *sites* de fornecedores antivírus, revistas especializadas (obviamente dependendo de sua periodicidade), *sites hackers*, organizações internacionais (CERT, SANS, VirusBTN, etc.), etc. Percebe-se que a Internet exerce papel fundamental nesse ínterim. Da mesma forma, a Internet,

mais precisamente uma Intranet e/ou Extranet, pode ser um canal eficiente de divulgação de informações importantes a todos os membros da organização, seja por meio de correio eletrônico ou publicação em páginas eletrônicas. Meios mais rudimentares também podem ser empregados, como afixações de orientações em murais nos departamentos ou confecção de documentos circulares. O importante é que a informação chegue rapidamente ao seu destino, mais uma vez buscando uma postura pró-ativa e preventiva.

O aspecto tecnológico é talvez um dos mais dispendiosos, sendo sua eficácia diretamente proporcional às soluções empregadas, a forma como foram empregadas e como é feita sua manutenção. Altamente pertinente, é uma análise custo-benefício, uma vez que conforme Frederick B. Cohen mencionou [COHEN, 1994, p.117], "o custo de ataques e defesas combinadas não pode exceder o custo de ataques sem nenhuma defesa". Isso leva a uma discussão a respeito de quanto pode valer o que será protegido. Será que os meios de proteção não custam mais caro de que perder todo o patrimônio protegido? [COHEN, 1994, p. 117], afirma que provavelmente a maioria das defesas contra vírus usadas por usuários são mais dispendiosas que o custo para a recuperação a partir de ataques no caso de não existir nenhuma defesa instituída.

Hoje, pode-se citar a Internet, como já discutido anteriormente, como sendo o principal vetor de entrada de *malwares* em uma organização. Para melhor se analisar a configuração mais adequada a ser adotada em um determinado ambiente, é interessante descriminar as diversas camadas em que as tecnologias antivirais podem ser adotadas. Atualmente, existem soluções antivírus que protegem os seguintes elementos em um ambiente corporativo:

- *Gateway*: ferramenta que possibilita a verificação de todo o tráfego de entrada e saída da e para a Internet. Entre os protocolos que tais ferramentas atuam, está o SMTP (correio eletrônico), HTTP (hiper texto) e FTP (transferência de arquivos). Comumente, estas ferramentas interagem com o *firewall*, se existir;
- *Groupware*: ferramenta especializada que protege servidores internos de correio eletrônico e aplicações de grupo em uma corporação;
- Servidores: ferramenta antivírus que visa a proteção de servidores de dados e aplicações em uma organização, principalmente no que diz respeito à integridade do sistema de arquivos;

- Estações: ferramentas que protegem o ambiente do usuário, a nível de *desktop*, e disponíveis móveis.
- Filtros: ferramentas que podem ajudar em uma metodologia preventiva, filtrando mensagens ou aplicações, baseando-se no conteúdo ou tipos de documentos ou arquivos em anexo;

Para melhor analisar as diferentes abordagens na aplicação dessas tecnologias, é interessante lançar mão de um modelo hierárquico em camadas, como descrito na figura 19:

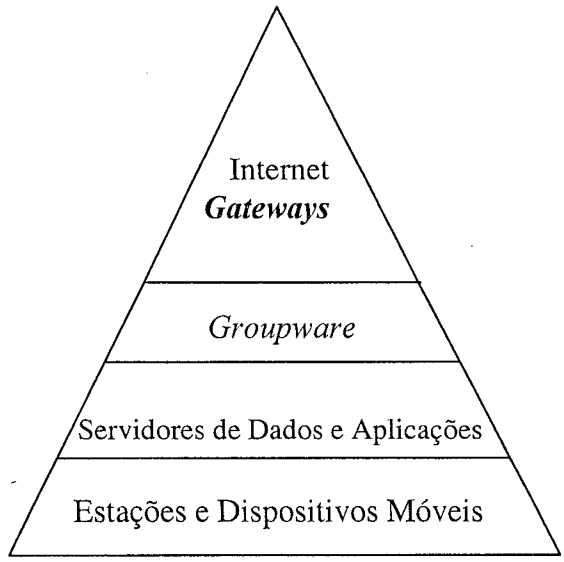

Figura 19: Modelo Hierárquico Piramidal

O modelo em camadas proposto, é organizado de uma forma diferenciada frente aos modelos utilizados para redes de computadores (ex.: OSI, TCP/IP), onde se organiza as camadas em níveis lógicos de comunicação, do plano físico às aplicações. Aqui, a idéia é organizar e agrupar as diversas entidades, com base em parâmetros como o volume (número de elementos) e o fluxo usual de acesso à Internet, ou seja, os acessos à Internet, de uma forma geral, seguem o fluxo no sentido das Estações de Trabalho em direção ao *gateway*. Na base da pirâmide têm-se as Estações de Trabalho e Máquinas Móveis, como *notebooks*. Esta camada abrangeria a grande maioria dos usuários no uso de seus microcomputadores. Uma camada acima, estariam os Servidores de Dados e Aplicações, onde se concentrariam grandes quantidades de dados de grande parte dos

usuários e da organização, possíveis aplicações utilizadas, etc. Na camada do *Groupware*, teria-se os servidores de correio eletrônico e serviços de grupo, como Lotus *Notes* ou Microsoft *Exchange*. Por fim, no topo da pirâmide, estaria o Internet *Gateway*, como porta de entrada e saída para a Internet.

Diversas organizações não teriam poder econômico para investir na proteção de todas as camadas descritas, pelo menos não ao mesmo tempo. É comum, a um consultor de segurança, ser indagado sobre qual das camadas deve ser protegida primeiro. A resposta inicial poderia ser: depende.

Em um primeiro aspecto, deve ser analisado o ambiente e tecnologias já implantadas para a proteção desse ambiente, e por fim, se as tecnologias levantadas continuarão a ser adotadas. A partir de então, o importante é verificar o orçamento que a empresa dispõe para investir e estudar alternativas de implantação.

Discorrendo sobre as alternativas de implantação de tecnologias antivirais para a proteção das camadas enumeradas, no modelo hierárquico piramidal, pode-se tomar emprestado dois termos utilizados na Engenharia de *Software*: *top-down* e *bottom-up*.

Em uma abordagem *top-down*, colocaria-se em foco, primeiramente, as camadas superiores da pirâmide. Esta abordagem vem sendo bastante adotada haja visto que o maior vetor no processo de contaminação e disseminação de *Malwares* tem sido, de forma absolutamente destacada, a Internet. Em 2000, a ICSA já anunciava que o correio eletrônico era o meio empregado em 87% das "infecções" registradas. Atualmente, em 2001, esse número é ainda maior. Se juntarmos ao correio eletrônico, os demais serviços disponibilizados na Internet (como http e ftp), fatalmente poderá se chegar a um percentual extremamente significativo, beirando a casa dos 100%. Assim, poderia se chegar à conclusão de que, ao se tentar filtrar códigos mal intencionados na entrada ou saída de uma rede em relação à Internet, estatisticamente, a rede estaria protegida contra mais de 90% dos possíveis casos de ocorrência.

Contudo, essa relação não deve ser analisada de uma forma tão direta. Deve-se lembrar, que hoje a Internet é uma tecnologia ao alcance de muitos, e muito provavelmente, a maioria dos funcionários de grande parte das organizações, faz uso da Internet em seus ambientes domésticos. Não é difícil imaginar que, em uma organização que se protege apenas através de tecnologias antivirais instaladas no Internet *Gateway*, possivelmente filtrando os tráfegos smtp, ftp e http de fora para dentro e de dentro para

fora, desconsiderando as demais camadas desse modelo, as estações dos usuários poderiam sofrer de uma epidemia sem precedentes, no caso de um usuário trazer uma mídia contaminada de sua casa, contaminação essa que bem poderia ter sido gerada através do uso da Internet. A implantação de uma tecnologia de segurança na camada de *Groupware*, poderia diminuir os riscos de uma contaminação em massa, visto que o correio interno e os demais serviços dessa camada estariam protegidos. Contudo, continuariam a persistir os riscos nas camadas inferiores, uma vez que os Servidores de Dados e Aplicações poderiam funcionar como vetores altamente eficientes na proliferação dos *Malwares*. As relações de confiança estabelecidas entre os recursos, principalmente através do conceito de compartilhamento, também implicariam em situações que poderiam se elevar a nível crítico no caso de contaminação por um *Malware* que fizesse proveito dessas características.

Em uma abordagem *bottom-up*, nesse caso, ao contrário da Engenharia de *Software*, considerada mais tradicional, o foco seria as camadas de base da pirâmide, em direção às camadas superiores. Embora menos considerado atualmente, visto que comercialmente já é um mercado mais amadurecido (antivírus para *desktop* e servidores são mais antigos que soluções para *gateways* ou *groupware*. Ainda em 1999, [STALINGS, 1999, p. 519] afirmava ser uma limitação do *firewall*, a impossibilidade de rastrear os códigos que entram ou saem em busca de vírus ou outros *malwares*. Hoje, tal capacidade já é desenvolvida por várias soluções antivírus que podem se integrar aos *firewalls*), é um modelo relativamente seguro. Ao se proteger todas as estações de trabalho dos usuários, protege-se também da maioria das possibilidades de um usuário introduzir um *Malware* na organização. No caso, ele até poderia receber uma arquivo infectado via correio eletrônico, mas o seu sistema antivírus o detectaria antes de que qualquer prejuízo fosse causado.

Contudo a abordagem *bottom-up* apresenta alguns problemas. Um deles é o de assumir uma postura tipicamente reativa, ou seja, permite-se a entrada do *Malware* na organização e depois trata-se de contê-lo. Além do mais, se houver algum problema na ferramenta antivírus ou a mesma estiver desatualizada, o *Malware* poderá fazer seu estrago tranqüilamente e o que é pior, poderá fazer uso de algumas características de ação remota, que poderão gerar estragos, inclusive, em máquinas devidamente protegidas. Outro problema grave é a questão do gerenciamento: todas as estações

devem estar sempre atualizadas e todas devem reportar as ocorrências a algum meio de gerenciamento centralizado. Se por acaso, alguma máquina estiver com o antivírus desatualizado, desabilitado, ou de uma forma ou de outra, com problemas em seu funcionamento, toda a rede estará correndo o risco. A figura 20 ilustra as abordagens.

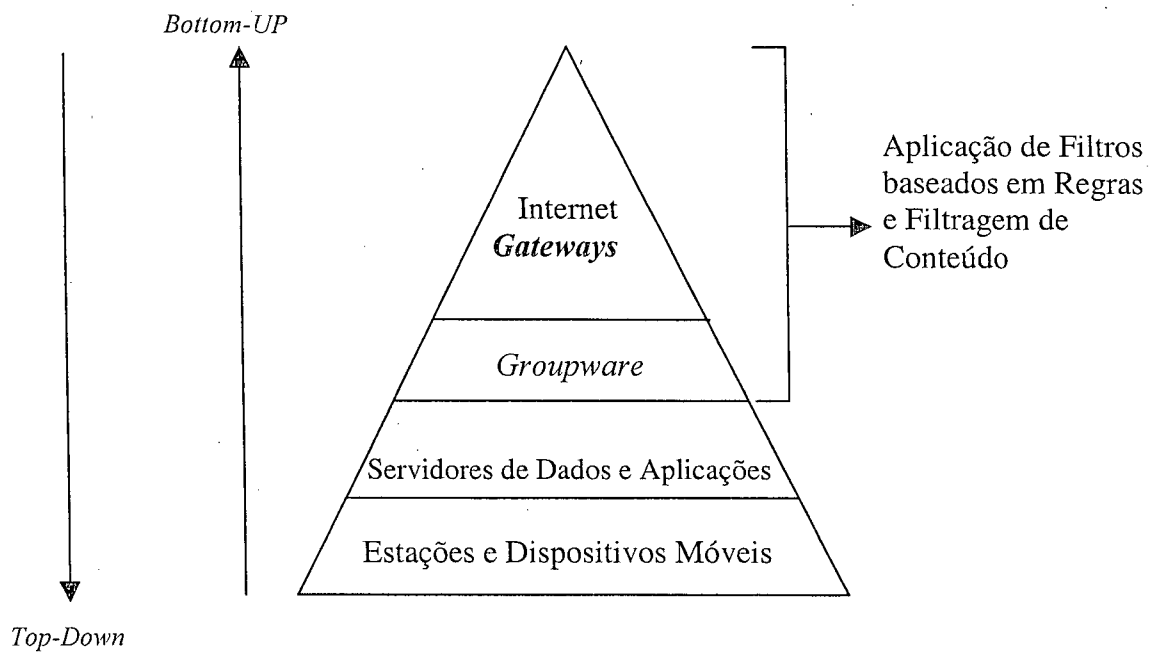

Figura 20: Abordagens *Top-Down* e *Bottom-Up*

A figura ilustra também, a possibilidade de se implantar filtros nas camadas superiores. Esses filtros, geralmente são instalados para a filtragem de mensagens de correio eletrônico no tocante a seus anexos, em busca de assinaturas de códigos virais ou de *Malwares*, ou em relação a seus conteúdos, podendo, também, serem utilizados para o filtro de outros serviços da Internet, como http e ftp. Esses filtros são extremamente interessantes na adoção de uma postura preventiva, uma vez que, munido da informação adequada, o administrador de sistema poderia conter o ataque de um *Malware* específico proibindo sua entrada por meio dos serviços da Internet mais utilizados, não havendo a necessidade da ação propriamente dita da ferramenta antivírus.

Apesar de se ter dividido a análise da implantação de tecnologias antivirais em duas abordagens, tais abordagens são apenas conceituais. Não existe padronização de implantação em nenhum dos sentidos e é delegado à cada empresa ou organização, com o apoio ou não de um consultor especializado, a decisão de escolher o que implantar

primeiro. O que se deve ter em mente contudo, é uma relação custo-benefício bem desenvolvida para melhor gerenciar o risco e aumentar os recursos de controle.

Muitas organizações já possuem um legado de tecnologias implantadas no passado e mesmo, por vezes, no caso de tais ferramentas não serem mais eficazes, a organização persiste em mantê-las, por causas muitas vezes de caráter subjetivo. Isso tem favorecido a adoção de ferramentas a nível de *gateway* e de *groupware* como de primeira necessidade. Essas duas camadas realmente são extremamente importantes, mas dificilmente serão eficazes num estratégia antivírus se não forem sustentadas pelas camadas inferiores, principalmente a nível de *desktop*. Alternativamente, a nível de *desktop*, por exemplo, poderia-se adotar estações desprovidas de dispositivos como unidades de disquete ou de CD, para evitar contaminações por esses meios, e com limitações nos compartilhamentos. Empresas que adotam abordagens alternativas, como o *thin-client*, não focam a segurança no *desktop* e possivelmente não enfrentarão problemas, desde que as demais camadas sejam contempladas e sejam considerados outros fatores não tecnológicos, como aqueles discutidos no Modelo das Tríades Conjugadas. Entre outras alternativas estaria a adoção de ferramentas que propiciam um melhor controle de acesso e geração de perfis para limitar a funcionalidade, adotando a estratégia de que cada um só deve ter acesso somente àquilo que possa ajudá-lo no desempenho de suas atividades no trabalho.

A necessidade do emprego de tecnologias está relacionado diretamente, de uma forma ou de outra, com os demais elementos do Modelo das Tríades Conjugadas. Por exemplo, em um ambiente em que determinados grupos de usuários são capacitados em aplicação de estratégias antivírus, a necessidade de tecnologia a níveis extremos não seria grande, visto que se estabeleceria um certo grau de confiança nos usuários em questão. Por outro lado, usuários que não estivessem tão bem preparados, precisariam estar munidos de ferramentas que viessem a automatizar mais o processo de defesa. Dessa forma, quanto mais for investido em Pessoas, Educação, Informação, Política e Processos, menos investimento será necessário em Tecnologia. Contudo, o bom senso deve estar sempre presente e um ponto de equilíbrio deve ser estabelecido, para que situações que possam levar ao caos, não sejam fomentadas. Somente a organização, orientada por especialistas capacitados e altamente competentes, poderá definir os esforços e investimentos necessários em cada um dos elementos do modelo, para que o

mesmo seja eficiente e venha a propiciar a concretização do objetivo maior que é a segurança plena contra *Malwares*, com níveis de controle e gerenciamento do risco bem estabelecidos.

CONCLUSÃO

O mercado atual, embora repleto de soluções antivírus com base tecnológica, é carente de profissionais que propiciem uma adequação pertinente dessas soluções nos ambientes corporativos. A palavra antivírus, anteriormente com a conotação de um pacote de *software*, atualmente precisa ser encarada como a agregação de serviços às tecnologias existentes. É nesse sentido que o modelo apresentado vem a contribuir na concepção de sistemas e estratégias de defesa mais eficazes.

Não é negado, em nenhum momento, que a Tecnologia exerce um papel fundamental no processo de defesa, visto que, fundamentando-se inclusive com trabalhos de outros autores, a automação da defesa é algo extremamente indicado e esforços nesse sentido terão respaldo nos resultados colhidos. Ameaças automatizadas de *software*, nascem todos os dias, e a cada dia exploram características e vulnerabilidades diferentes, e em diversos casos é praticamente impossível, sem o uso adequado de tecnologias, se proteger contra essas ameaças por meio apenas de algumas metodologias.

Contudo a realidade vem comprovando que as organizações precisam mais do que simples *softwares*. Precisam de Pessoas capacitadas a implantá-los adequadamente, de Pessoas para gerenciá-los e fornecer a manutenção necessária, e de Pessoas preparadas para o pior: o ataque de um *Malware*. A prevenção é, sem dúvida, a melhor estratégia para um sistema de defesa. Entretanto, prevenir não é tão simples como se possa imaginar. Envolve mais do que Pessoas e Tecnologias, envolve também Educação, Política e Processos, e muita Informação.

Um sistema estratégico de defesa, deve buscar além da prevenção, um planejamento envolvendo todos os elementos base do Modelo das Tríades Conjugadas proposto, para todas as situações e ocorrências imagináveis. Planos de contingência são imprescindíveis. Como Mark Ludwig alertou na introdução do livro *The Giant Black Book of Computer Viruses* [LUDWIG, 2000], em um sistema com a probabilidade de ser bem sucedido em 99% das ocorrências, teria-se as seguintes probabilidades ao se considerar diferentes freqüências de ataques: a) um ataque por semana: $P = (0.99)^{52} =$

0.59; b) um ataque por dia: $P = (0.99)^{365} = 0.026$. Ou seja, no caso de se ter um ataque por dia, em um sistema 99% seguro, existe a chance de 97,4% , em um ano, de que um ataque obtenha êxito. Assim sendo, no melhor dos sistemas, quase ideal, imaginando-se uma segurança de 99% (100% é utopia), existe a real possibilidade de que o sistema seja invadido pela ação de pelo menos um *Malware*, no intervalo de um ano. Isso são apenas dados probabilísticos, podendo variar para baixo ou para cima dependendo da eficácia do *Malware*. Portanto, fundamenta-se a necessidade dos planos de contingência para todos os ambientes analisados e protegidos.

Um modelo para ser efetivo deve ser discutido e documentado, e mais do que isso, praticado. Todos devem ser conscientizados da sua existência e implantação e cooperar para que o mesmo surta os efeitos desejados. Com relação ao Modelo das Tríades Conjugadas o caso não é diferente. Tudo em relação ao modelo deve ser concebido de acordo com os interesses da organização, através de amplas discussões e com a geração da documentação adequada.

Um aspecto final e essencial para o sucesso do modelo proposto é o gerenciamento. Tudo deve ser passível de controle. Deve-se focar na confiabilidade dos sistemas e processos, pois "a força de uma corrente se mede pela capacidade de resistência do elo mais fraco". Com a consciência dos pontos críticos do sistema, é possível fazer um gerenciamento adequado do risco, gerando processos adequados para contorná-los e diminuir o seu poder de impacto no sistema como um todo.

O modelo sugerido é genérico e necessita de adaptações onde for aplicado. Mas os princípios suportados pelos elementos das duas tríades conjugadas sempre serão mantidos. Trata-se de um modelo amplo e independente, permanecendo sempre atualizado e sobrevivendo a avanços tecnológicos em ambos os sentidos, antivírus e *Malware*. Trata-se de um modelo heterogêneo, buscando acoplamento entre elementos que precisam se comunicar, gerando um alto grau de sinergia no gerenciamento do sistema de segurança como um todo. Trata-se, por fim, de uma espécie de *Moated Wall* moderno.

Como sugestão para trabalhos futuros, pode-se citar, por exemplo, a confecção de um modelo em níveis, que venha a estabelecer parâmetros para o enquadramento de uma empresa analisada em um estágio/nível de segurança, ou mesmo maturidade, no combate e controle contra *malware*. Com base em um enquadramento dessa espécie,

poderia-se delinear e padronizar uma margem de investimento necessário para se progredir de nível ou para se atingir um determinado nível desejado, facilitando a implantação do modelo desejado e proporcionando crescimentos relativos e conscientes na segurança do ambiente, deixando claro os possíveis riscos que cada nível pode apresentar, e as soluções tecnológicas e outros esforços necessários ao seu alcance e manutenção.

REFERÊNCIAS BIBLIOGRÁFICAS

[ANDE, 1980] – Aderson, J. *Computer Security Threat Monitoring and Surveillance.* Fort Washington, PA: James P. Anderson, Co., Abril de 1980.

[ANONIMO, 2000] – Autor Anônimo. Segurança Máxima para Linux. Rio de Janeiro: Editora Campus, 2000. 761 p.

[BERNSTEIN, 1997] - Bernstein, Terry; et all. Segurança na Internet. Rio de Janeiro: Campus, 1997. 461 p.

[BONTCHEV, 1994, p. 3] – Bontchev, Vesselin; *Future Trends in Virus Writing*. Virus Test Center, University of Hamburg, Abril de 1994.

[CHESS, 1997] – Chess, David; *The Future of Viruses on the Internet*. Virus Bulletin, 1997.

[CIDALE, 1990] - Cidale, Ricardo A. Vírus Digital. São Paulo: McGraw-Hill, 1990.

[COHEN, 1994] – Cohen, Frederick B.; *A Short Course on Computer Virus* 2nd ed. New York –USA: Wiley, 1994. 250 p.

[DENNING, 1990] - Denning, Peter..*Computers Under Attack: Intruders, Worms, and Viruses*. ACM Press, 1990. 554 p.

[DRAKOS a, 1993] - Drakos, Nikos ; *History of Virus* Computer Based Learning Unit, University of Leeds http://csrc.ncsl.nist.gov/nistir/threats/subsubsection3_3_1_1.html#SECTION000311 0000000000000

[DRAKOS b, 1993] - Drakos, Nikos ; *History of Worms* Computer Based Learning Unit, University of Leeds http://csrc.ncsl.nist.gov/nistir/threats/subsubsection3_3_2_1.html#SECTION000321 0000000000000

[FIPS 191, 1994] – Federal Information Processing Standards Publications 191. *Specifications for Guidelines for the Analysis Local Area Network Security.* November, 1994.

[GARGAGLIONE, 1999] – Gargaglione, Bruno Diegoli; De Paula, Pedro Castanheira; Vírus: uma Ameaça Letal. Rio de Janeiro: Brasport, 1999. 187 p.

[GORDON, 1995] – Gordon, Sarah; *The Anti-Virus Strategy System*. Virus Bulletin, 1995.

[ICSA, 1999] - *TruSecure Anti-Virus Policy Guide Version 3.12*; *Specific and Detailed ICSA Anti-Virus Policy Suggestions*. Revised December 10, 1999.

[KATZAN, 1997] - Katzan Júnior, Harry. Segurança de Dados em Computação. Rio de Janeiro: Livros Técnicos e Científicos, 1977. 136 p.

[LUDWIG, 2000] - Ludwig, Mark. *The Giant Black Book of Computer Viruses.* 2nd. ed. USA: Lexington & Concord Partners, Ltd., 2000. 462 p.

[LUNDELL, 1989] – Lundell, Allan. Vírus! O mundo secreto dos invasores dos computadores que se reproduzem e destroem. Portugal: Edições CETOP, 1989. 178 p.

[MAX, 2000] - Autor Anônimo. Segurança Máxima. Rio de Janeiro: Campus, 2000. 826 p.

[MCAFEE, 1997] – McAfee *White Paper on New Trends in Internet and Intranet Security*; *Protecting Networks and PCs from Hostile Java and ActiveX Applications*. McAfee Associates, Inc. Novembro/1997. 10 p.

[MURILO, 2001] – Murilo, Nelson; Klaus Steding-Jessen. Métodos para Detecção Local de Rootkits e Módulos de Kernel Maliciosos em Sistemas UNIX. São José dos Campos: Anais do Simpósio Segurança em Informática: CTA/ITA/IEC. 2001. p. 133-139

[PITKOWSKI, 1992] Pitkowski, André; Vírus: prevenção, planejamento e controle, proteção de rede local, vírus mais comuns; São Paulo: Atlas, 1992.

[POLK, 1995] - Polk, W. Timothy, et al. *Anti-Virus Tools and Techniques for Computers Systems.* New Jersey: Noyes Data Corporation, 1995. 79 p.

[RADATTI a, 1996] - Radatti, Peter V. *The Plausibility of UNIX Virus Attacks* USA: CyberSoft, Incorporated , 1996. http://www.cyber.com./papers/plausibility.html

[RADATTI b, 1996] – Radatti, Peter V. *Computer Viruses In Unix Networks* USA: CyberSoft, Incorporated , 1996. http://www.cyber.com./papers/networks.html

[REIS, 2001] - Reis, Marcelo Abdalla dos; Fernandes, Diego de Assis Monteiro; Paula, Fabrício Sérgio de; Geus, Paulo Lício de. Modelagem de um Sistema de Segurança Imunológico. São José dos Campos: Anais do Simpósio Segurança em Informática: CTA/ITA/IEC. 2001. p. 91-100.

[ROCHA, 2001] - Rocha, João Luiz Francalacci; Custódio, Ricardo Felipe. Proteção de *Software* por Certificação Digital. São José dos Campos: Anais do Simpósio Segurança em Informática: CTA/ITA/IEC. 2001. p. 17-26

[SANKARY, 1989] – Sankary, Tim. *Developing Virus Identification Products*, 1989. http://futon.sfsu.edu/~kk/virus/history.txt

[SARC a, 1998] - SARC Virus Expert Training Program. *Level 1: Concepts and Foundation. Module 1 - Computer Virus Concepts, Ethics, and Trends.* Symantec Corporation, 1998.

[SARC b, 1998] - SARC Virus Expert Training Program. *Level 1: Concepts and Foundation. Module 2 – Virus Functionality and the Infection Process.* Symantec Corporation, 1998.

[SARC c, 1998] - SARC Virus Expert Training Program. *Level 1: Concepts and Foundation. Module 3 – Virus Handling and Removal* Symantec Corporation, 1998.

[SERAFIM, 2001] – Serafim, Vinícius da Silveira; Weber, Raul Fernando. Um Verificador Seguro de Integridade de Arquivos. São José dos Campos: Anais do Simpósio Segurança em Informática: CTA/ITA/IEC. 2001. p. 169-174

[SOARES, 1995] - Soares, Luiz Fernando Gomes; LEMOS, Guido; COLCHER, Sergio. Redes de Computadores: das LANS, MANS e WANS as redes ATM. Segunda Ed. Rio de Janeiro: Campus, 1995. 705 p.

[SOBERS, 2000] – Sobers, Larry; *Anti-Virus Architecture: A 4-Layered Approach*. SANS Institute. Outubro/2000. www.sans.org/infosecFAQ/malicious/anti-virus.htm

[SPAFFORD, 1989] – Spafford, Eugene. *The internet worm program: An analysis.* Computer Communication Review. 19(1), January 1989.

[SPAFFORD, 1996] – Spafford, Gene; Garfinkel, Simson. *Pratical Unix & Internet Security*. 2nd ed. USA: O'Reilly, 1996. 1004 p.

[STALLINGS, 1999] - Stallings, William; *Cryptography and Network Secutity: principles and practice* 2nd ed. New Jersey-USA: Prentice Hall, 1999. 569 p.

[SYMANTEC a, 1997] – The Symantec Enterprise Papers Volume XXX *Understanding Symantec's Anti-Virus Strategy for Internet Gateways.* Symantec Corporation, 1997.

[SYMANTEC, 2001] – Symantec *Virus Protection Technical Brief. The Digital Immune System: Enterprise-Grade Anti-Virus Automation in 21st century.* Symantec Corporation, 2001.

[TANENBAUM, 1997] - TANENBAUM, Andrew S. Redes de Computadores. Terc. Ed. Rio de Janeiro: Campus, 1997. 923 p.

[TESAURO, 1996] – Tesauro, Gerald; Kephart, Jeffrey O.; Sorkin, Gregory B.; *Neural Networks for Computer Virus Recognition*. IBM Thomas J. Watson *Research Center*. New York – USA: IEE *Expert*, , Abril de 1996.

[TREND, 1997] – Trend Micro, Inc; *Designing and Implementing a Virus Prevention Policy:Key Issues and Critical Needs.* Setembro/1997.

[VASCONCELLOS, 1999] – Vasconcellos, Márcio José Accioli de; A Internet e os Hackers: Ataques e Defesas. 2ª ed. Chantal: 1999. 336 p.

[WEBER, 1989] – Weber, Raul Fernando; Vírus de Computador.
PGCC/II – Universidade Federal do Rio Grande do Sul
Revista de Informática Teórica e Aplicada – Outubro de 1989 – Vol. 1. Num. 1.
p. 79 – 118.

[WEINBERG, 1975] - Weinberg, Gerald; *An Introduction to General Systems Thinking*. p.51,. John Wiley and Sons, 1975.

[WOOD, 1997] - Wood , Charles Cresson. *Policies From the Ground Up, Infosecurity News.* Março/Abril 1997, pp 24-28.