Universidade Metodista de Piracicaba - UNIMEP

Faculdade de Ciências Exatas e da Natureza - FACEN

Mestrado em Ciência da Computação

SISTEMAS DE MISSÃO CRÍTICA: ANÁLISE DA CAPACIDADE DE SOBREVIVÊNCIA FRENTE A ATAQUES INTERNOS

DAGOBERTO GANÉO DELLAI

ORIENTADOR: PROF. DR. MÁRCIO MERINO FERNANDES

PIRACICABA
2006

Universidade Metodista de Piracicaba - UNIMEP

Faculdade de Ciências Exatas e da Natureza - FACEN

Mestrado em Ciência da Computação

SISTEMAS DE MISSÃO CRÍTICA: ANÁLISE DA CAPACIDADE DE SOBREVIVÊNCIA FRENTE A ATAQUES INTERNOS

DAGOBERTO GANÉO DELLAI

ORIENTADOR: PROF. DR. MÁRCIO MERINO FERNANDES

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação, da Faculdade de Ciências Exatas e da Natureza, da Universidade Metodista de Piracicaba – UNIMEP, como requisito para obtenção do Título de Mestre em Ciência da Computação.

PIRACICABA
2006

À

minha esposa, Marta, e filhos, Beatriz e Felipe.

Especialmente aos meus pais Sérgio e Maria Alice, por terem me proporcionado uma Graduação e Pós-Graduação.

AGRADECIMENTOS

Ao Prof. Dr. Márcio Merino Fernandes, pela orientação, compreensão e incentivo dispensado ao desenvolvimento deste trabalho.

Ao Prof. Dr. Abimael Aranha Netto, Coordenador da Diretoria de Apoio Didático Científico e Computacional da Faculdade de Ciências Médicas – Unicamp, pelo apoio, incentivo e amizade.

Ao programador Thomas dos Reis Zambotti, funcionário do Laboratório de Informática da Faculdade de Ciências Médicas – Unicamp, pela ajuda com a ferramenta EASEL.

Aos Profs. Drs. Fernando Cendes e Dr. Li Li Min, do departamento de Neuroimagem da Faculdade de Ciências Médicas – Unicamp, por terem disponibilizado um Mac com Sistema Operacional Mac OS X, para rodar a ferramenta EASEL.

Ao Fabrício Ramos Silvestre Pereira, responsável pelo Mac, no departamento de Neuroimagem.

RESUMO

A metodologia SNA (Survivable Network Analysis) destina-se à análise de segurança de redes com ênfase na sobrevivência do sistema, e não na simples prevenção de todo e qualquer ataque. Visa melhorar as chances de manutenção (sobrevivência) de operações consideradas como missão crítica para a organização. Neste documento, mostra-se que a sobrevivência de sistemas de missão crítica é também ameaçada por ataques e falhas internas, e não apenas externas, como normalmente se acredita. A metodologia descrita nesta dissertação visa propor recomendações para configurações e procedimentos utilizados na administração e operação de sistemas baseados em redes de computadores. Duas situações hipotéticas foram utilizadas para a aplicação da metodologia. As recomendações adotadas são definidas através de simulações com a ferramenta EASEL, especialmente desenvolvida pelo CERT para trabalhos nessa área. A combinação do método SNA com a utilização de simulações mostrou-se efetiva para a definição de procedimentos e configurações de segurança, uma vez que permite quantificar o nível de eficiência esperado.

Palavras-Chave: Segurança em computadores, Sobrevivência de sistemas, Redes de computadores, Ameaças internas, Simulação, Easel.

ABSTRACT

The SNA methodology (Survivable Network Analysis) can be used on network security analysis, with emphasis on system survivability, as opposed to simply attack prevention. It aims to improve the system chances of surviving, specially of those operations considered to be mission critical for the organization. This document shows that the survivability of mission critical systems is also threatened by *internal* attacks and failures, and not only by external ones, as it is usually believed. The methodology presented in this dissertation could be used to define recommendations for configurations and procedures used in the administration and operation of systems based on computer networks. Two hypothetical situations were used to illustrate the application of the methodology. The suggested recommendations were defined through simulation with EASEL, a tool specially developed by the CERT organization. The combination of the SNA method with EASEL has shown to be effective for the definition of security procedures and configurations, as it allows to quantify the expected effectiviness level.

Key-words: Computers Security, System Survivability, Computer Networks, Internal Threats, Simulation, Easel.

SUMÁRIO

LISTA DE FIGURAS

PÁG.

LISTA DE TABELAS

LISTA DE GRÁFICOS

1 – INTRODUÇÃO

1.1 – CONTEXTO DA PESQUISA

Nos últimos anos assistiu-se a um aumento da utilização de computadores enquanto seu preço e tamanho diminuíram drasticamente. Com este aumento, observou-se um crescente número de equipamentos conectados em rede, ampliando a infra-estrutura de comunicação. Em especial, destaca-se a tecnologia Ethernet e a rede Internet.

Esses dois crescimentos abriram caminhos para novos cenários como, redes de computadores, estrutura cliente servidor, etc. Com esses novos cenários foi-se encadeando novas preocupações com a segurança de sistemas. Apesar de questões de segurança serem muito discutidas e trabalhadas, aspectos relacionados com a segurança interna das organizações não têm sido muito explorados. Para auxiliar no processo de garantir a segurança de sistemas de uma organização, estudaremos o conceito de sobrevivência (survivability), que pode ser definido como a capacidade de manter processos de missão crítica em operação, mesmo na presença de ataques ou falhas de diversos tipos (equipamentos, software, humanas).

Estudos de sobrevivência de sistemas podem ser feitos através do método SNA (Survivable Network Analysis). Este constitui-se na análise dos chamados “3 Rs”, que são: resistência, reconhecimento e recuperação. No caso, os principais pontos de vulnerabilidade da rede são identificados. A partir disso, estratégias para resistir a ataques, reconhecer ataques, e recuperar-se de ataques são definidas. Com os pontos de vulnerabilidades obtidos, aplicaremos a ferramenta de simulação EASEL, que simulará os cenários de intrusão, definidos pelo método SNA, guiando decisões para a definição de procedimentos para cada um dos “3 Rs”.

1.2 – OBJETIVOS

No contexto descrito acima, o objetivo principal deste trabalho é:

Avaliar possíveis descrições e procedimentos antes de executá-los, principalmente no que diz respeito à eficácia dos mesmos. Isso é necessário porque, na prática, envolvem tempo e dinheiro. Para isso utilizamos a metodologia SNA e Simulação.

Para esse fim, pretende-se trabalhar nos seguintes pontos:

- Avaliar a adequação da metodologia SNA em conjunto com a ferramenta EASEL para o tratamento de problemas de segurança interna de redes de computadores.
- Recomendar algumas ações preventivas em ambientes ou sistemas sujeitos a ataques internos
- Disponibilizar um conjunto de experiências e orientações para aplicação da metodologia

1.3 – JUSTIFICATIVAS PARA ESTE TRABALHO

As organizações investem milhões, tanto em sistemas, equipamentos como em pessoal da área de informática, mas problemas de segurança continuam a ocorrer, por mais bem preparada que a organização possa ser.

Os sistemas controlam praticamente toda a organização, tornando-as dependentes, uma pequena falha, e toda organização pára, sem perspectiva de voltar a oferecer seus serviços, deixando usuários sem condições de utilizar os serviços oferecidos por ela. A falta desses serviços pode causar conseqüências que chegam à perda de credibilidade do usuário em relação à organização.

Apesar de diversos avanços terem ocorrido na área de segurança de sistemas, ainda não é possível garantir que as medidas adotadas sejam suficientes para manter níveis de operação mínimos exigidos. Novas abordagens são necessárias para tratar o problema, sendo esta a principal justificativa desta pesquisa.

1.4 – RESULTADOS E CONTRIBUIÇÕES ESPERADAS

Após a execução do trabalho que está sendo proposto, pretendemos fazer uma análise da metodologia empregada, sendo que por ser nova, não disponibiliza muitas informações e resultados.

Com esta análise, contribuiremos com um conjunto de procedimentos para a aplicação da metodologia e ferramenta de simulação EASEL.

1.5 – ORGANIZAÇÃO DO TRABALHO

Este documento está organizado da seguinte forma:

- Capítulo 1 – Mostra o contexto da pesquisa, objetivos, justificativas e os resultados e contribuições esperados.
- Capítulo 2 – Neste capítulo é mostrado alguns Conceitos Básicos de Redes de computadores.
- Capítulo 3 – Neste capítulo falaremos de seguranças, como seus problemas e alguns mecanismos de proteção.
- Capítulo 4 – Neste capítulo falaremos da Sobrevivência de um Sistema de Missão Crítica, de uma nova disciplina de sobrevivência que está surgindo e do método de análise de segurança de rede (SNA).

- Capítulo 5 – Neste capítulo aplicaremos o método SNA em um caso real e utilizaremos a ferramenta EASEL para simular dois casos hipotéticos.

- Capítulo 6 – Conclusão, experiências e recomendações para a aplicação da metodologia e da ferramenta de simulação EASEL e relacionar alguns trabalhos para pesquisas futuras.

2 – REDES DE COMPUTADORES

2.1 – SISTEMAS DE INFORMAÇÕES CORPORATIVAS

Conforme [CIDRAL,2000], as organizações contemporâneas possuem na Tecnologia da Informação um elemento estratégico, na medida que as soluções tecnológicas automatizam processos organizacionais e são fonte de vantagens competitivas através da análise de cenários, apoio ao processo decisório, definição e implementação de novas estratégias organizacionais. Assim, cresce a preocupação com a coleta, armazenamento, processamento e transmissão da informação na medida que a disponibilidade da informação certa, no momento certo, é requisito fundamental para a melhoria contínua da qualidade e competitividade organizacionais.

Por isso, podemos dizer que a Missão Crítica de uma Organização está relacionada com o Sistema de Informação, e este apóia-se em redes de computadores, que não são seguras.

Neste sentido, de acordo com [MEC,1998], tem-se que:

> *"Sistemas de Informação podem ser definidos como uma combinação de recursos humanos e computacionais que inter-relacionam a coleta, o armazenamento, a recuperação, a distribuição e o uso de dados com o objetivo de eficiência gerencial (planejamento, controle, comunicação e tomada de decisão) nas organizações. Adicionalmente, os sistemas de informação podem também ajudar os gerentes e os usuários a analisar problemas, criar novos produtos e serviços e visualizar questões complexas."*

Desta forma, a área de sistemas de informação envolve dois grandes níveis:

a) aquisição, desenvolvimento e gerenciamento de serviços e recursos da tecnologia da informação:

b) desenvolvimento e evolução de sistemas e infra-estrutura para uso em processos organizacionais.

Sistemas de informação apóiam-se quase que 100% dos casos em redes, nas suas diversas formas, conforme descrito na seção 2.2.

2.2 – REDES

Devido à grande utilização de computadores no mundo todo, em instituições de ensino, no setor comercial e em residências tornou-se interessante a interconexão destes equipamentos, formando um poderoso e eficiente meio de comunicação que permite usufruir de simples compartilhamento de recursos, como impressoras e espaço em disco (mídia fixa ou removível).

Conforme a evolução das tecnologias aplicadas, acredita-se que as redes de computadores poderão ser os principais veículos de comunicação [TANENBAUM,1997].

A comunicação para a humanidade é algo natural, pois nenhum ser humano consegue viver isoladamente e, desde seu surgimento, pode-se observar o desenvolvimento de técnicas para suprir esta necessidade de contato com outrem.

Isto pode ser exemplificado com a evolução dos toques de tambor, uso de sinais de fumaça e por pombos-correio, o surgimento do telégrafo em 1838 e seu desenvolvimento até a presente data com o uso de rádios, televisores e até as comunicações via satélite.

Centrando a atenção no desenvolvimento da informática (que antigamente oferecia riscos às organizações, pois defeitos em equipamentos causavam a paralisação de todo o serviço) para o surgimento de minicomputadores e, posteriormente, microcomputadores, em que a utilização de redes de informação proporcionou melhorias na estruturação organizacional.

Dessa forma, a descentralização do processamento também permitiu o compartilhamento de recursos (como meios de armazenamento de

dados, impressoras, softwares, por exemplo), maior confiabilidade, modularidade dos sistemas, novos serviços, entre outras vantagens, que facilitaram a comunicação entre pessoas.

Existem basicamente dois tipos de redes:

Redes Confinadas: são aquelas cujas distâncias entre os MPs (módulos processadores) não são maiores que alguns poucos metros.

Redes Geograficamente Distribuídas: São aquelas que ultrapassam os limites das Redes Confinadas.

As redes se dividem em três grandes classes:

- Local Area Networks (rede local)- LANs

Redes locais surgiram para viabilizar a troca e o compartilhamento de informações e dispositivos periféricos (recursos de hardware e software), preservando a independência das várias estações de processamento, e permitindo a integração em ambientes de trabalho cooperativo. Pode-se caracterizar uma rede local como sendo uma rede que permite a interconexão de equipamentos de comunicação de dados numa pequena região. Outras características típicas encontradas e comumente associadas a redes locais são: altas taxas de transmissão e baixas taxas de erro, outra característica é que em geral elas são de propriedade privada.

- Metropolitan Area Networks (redes metropolitanas) - MANs

Quando a distância de ligações entre vários módulos processadores, começa a atingir distâncias metropolitanas, chamamos esses sistemas não mais de rede locais, mas de MANs.

Uma rede metropolitana apresenta características semelhantes às redes locais, sendo que as MANs em geral, cobrem distâncias maiores que as LANs [SOARES,1995].

- Wide Area Networks (rede geograficamente distribuída) - WANs

Surgiu da necessidade de se compartilhar recursos especializados por uma maior comunidade de usuários geograficamente dispersos. Há um custo de comunicação bastante elevado (circuitos para satélites). Em face de várias considerações em relação ao custo, a interligação entre os diversos módulos processadores, em uma rede, determinará a utilização de um arranjo topológico específico e diferente daqueles utilizados em redes locais. Ainda por problemas de custo, as velocidades de transmissão empregadas são baixas: da ordem de alguns kilobits/segundo, embora alguns enlaces cheguem hoje à velocidade de megabits/segundo.

2.2.1 - ARQUITETURAS DE REDES

A tarefa de permitir a comunicação entre aplicações, executados em máquinas diferentes, envolve uma série de detalhes que devem ser cuidadosamente observados, para que esta comunicação ocorra de maneira precisa, segura e livre de erros. Por exemplo, detalhes de sinalização dos bits para envio através dos meios de transmissão, detecção e correção de erros de transmissão (pois a maioria dos meios de transmissão são passíveis de interferências), roteamento das mensagens, desde sua origem até o seu destino, podendo passar por várias redes intermediárias, métodos de endereçamento tanto de *hosts* quanto de aplicações, cuidar da sintaxe e semântica da informação, de modo que, quando uma aplicação transmitir um dado do tipo inteiro, a aplicação destino possa entendê-lo como do tipo inteiro, isto é, preservar a confiabilidade dos dados de um emissor até um receptor.

Para reduzir a complexidade de projeto, a maioria das redes de computadores são estruturadas em camadas ou níveis, onde cada camada desempenha uma função específica dentro do objetivo maior que é a tarefa de comunicação. As camadas são construídas umas sobre as outras e cada camada oferece seus serviços para as camadas superiores.

A camada N, em uma máquina, para desempenhar suas funções estabelece uma conversação com a camada N em outra máquina. As regras utilizadas nesta conversação são chamadas de protocolo da camada N. As funções de cada camada são executadas por entidades (processos, que podem ser implementados por software ou por hardware), Figura 2.1. Entidades que executam em camadas correspondentes e em máquinas distintas são chamadas de processos pares *peers*. São os processos pares que se comunicam, utilizando o protocolo de sua camada.

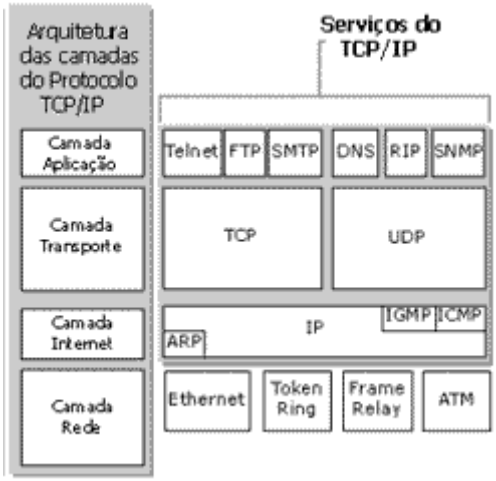


Figura 2.1 – Modelo TCP/IP e suas camadas

Na verdade, nenhum dado é transferido diretamente da camada N de uma máquina para a camada N de outra máquina. Em vez disso, cada camada passa dados e informações de controle para a camada imediatamente abaixo, até encontrar o meio físico, através do qual a comunicação de fato ocorre. Na máquina destino a mensagem percorre o caminho inverso, da camada mais inferior para a mais superior, com cada camada retirando e analisando as informações de controle colocadas pela sua camada correspondente na máquina origem. Após esta análise, a camada decide se passa o restante dos dados para a camada superior. Essas informações de controle correspondem ao protocolo da camada e também são conhecidos como cabeçalho do protocolo.

Para ilustrar, o conceito de comunicação através de múltiplas camadas, Figura 2.2, consideremos o seguinte exemplo:

- Duas pessoas em países diferentes desejam trocar informações sobre um projeto de software. Um analista só fala português e o outro só se comunica em inglês. Para se comunicarem eles decidem utilizar um tradutor;

- Considere ainda, que o idioma comum entre os tradutores seja o alemão e que o meio utilizado para transmissão dos dados seja o telégrafo;

- Assim, o analista que fala português passa suas informações para seu tradutor que as traduz para o alemão. A mensagem em alemão é então passada ao telegrafista que as transmite para um telegrafista no outro país;

- Ao receber a mensagem, o telegrafista passa para o tradutor que a traduz para o inglês e a entrega para o analista.

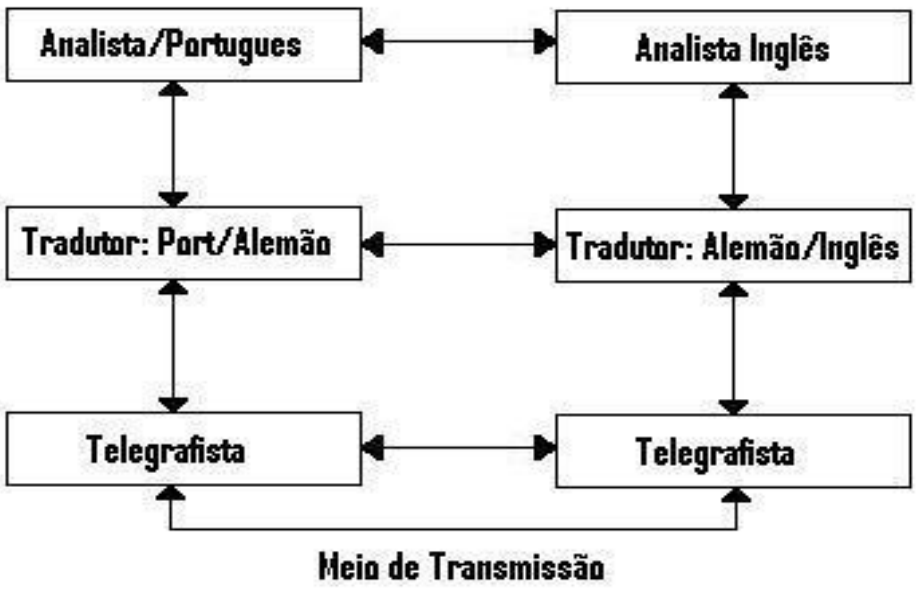


Figura 2.2 - Comunicação através de múltiplas camadas.

Nota-se que existe uma interface entre cada par de camadas adjacentes. É ela que definirá quais e como as funções oferecidas pela camada

inferior podem ser acessadas pela camada superior. Esta interface deve ser bastante clara, de modo que, ao trocar-se a implementação de uma camada por outra completamente diferente, não seja necessária modificar as outras camadas. Isso é possível desde que a interface entre as camadas seja mantida. Por exemplo, trocando-se linhas telefônicas por transmissão via satélite, a implementação da camada responsável por manipular o acesso ao meio de transmissão deverá modificar completamente sua implementação, porém as demais camadas não sofrerão estas modificações desde que os mesmos serviços anteriores e o modo como são oferecidos sejam mantidos.

Neste contexto, o conjunto das camadas e protocolos é chamado de ARQUITETURA DE REDE.

2.2.2 - PADRÕES PARA INTERCONEXÃO DE REDES DE COMPUTADORES

As primeiras arquiteturas de rede foram desenvolvidas por fabricantes de equipamentos, os quais desenvolviam soluções para interconexão apenas de seus produtos, sem se preocuparem com a compatibilidade de comunicação com equipamentos de outros fabricantes. Assim o fizeram, por exemplo, a IBM (International Business Machines Corporation) ao anunciar sua arquitetura de rede SNA (System Network Architecture), e a DEC (Digital Equipament Corporation) com sua DNA (Digital Network Architecture). Essas arquiteturas são denominadas proprietárias.

Desse modo, computadores de fabricantes diferentes não podiam se comunicar, impondo uma grande limitação aos consumidores, pois ficavam “amarrados” aos produtos de um único fabricante, caso quisessem que seus equipamentos se comunucassem.

Torna-se evidente a necessidade de um conjunto de regras que permitam a comunicação ou interconexão entre dois sistemas quaisquer, sem considerar seu fabricante. Surgem as arquiteturas para interconexão de sistemas abertos: a Arquitetura Internet, desenvolvida por pesquisadores patrocinados pelo Departamento de Defesa dos Estados Unidos, e a Arquitetura OSI (Open Systems Interconnection) desenvolvida pela

comunidade internacional sob a coordenação da ISO (International Standards Organization).

O TCP (transmission control protocol – protocolo de controle de transmissão) e o IP (internet protocol – protocolo da internet) são os protocolos mais importantes da Internet, o protocolo IP especifica o formato da informação que é enviada e recebida entre os roteadores e os sistemas finais. O caminho que a informação transmitida percorre do sistema final de origem, passando por uma série de enlaces de comunicação e roteadores, para o sistema final de destino é conhecido como rota ou caminho pela rede.

2.2.3 – A PERIFERIA DA REDE

Os computadores que utilizamos (PC em casa, ou no trabalho) são conhecidos na área de informática como hospedeiros ou sistemas finais. São chamados de hospedeiro *hosts* porque rodam aplicações como: um browse da Web, programa de e-mail, etc. Também são chamados de sistemas finais *end system* porque se situam na periferia da internet como mostra a Figura 2.3. Os sistemas finais são subdivididos em duas categorias: clientes e servidores. Clientes são os PCs e os servidores são computadores mais poderosos.

Um programa cliente que roda em um PC pede e recebe informações de um servidor que roda em um computador mais poderoso. Esse modelo é, sem dúvida, a estrutura que predomina na Internet.

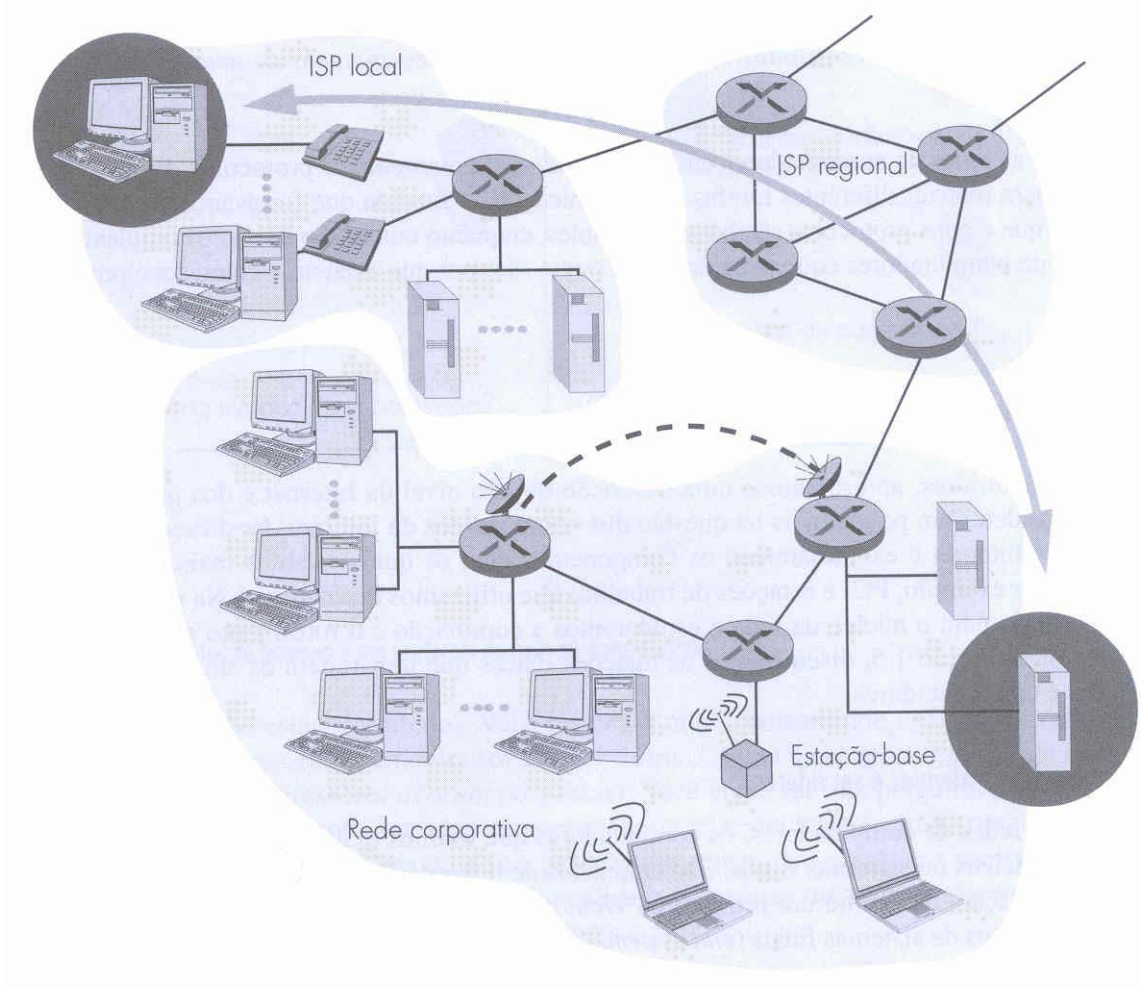


Figura 2.3 - Interação entre Sistemas Finais.[KUROSE,2003]

2.2.4 – REDES DE ACESSO

Existem três tipos de redes de acesso na internet:

- **Rede de acesso residencial**, que liga um sistema final (PC) à rede. A forma mais comum de acesso residencial é o uso de um modem com uma linha discada, ligado por meio da rede pública de telefonia a um ISP (Provedor). Para fazer esta conexão do PC ao roteador de borda é utilizado uma rede de acesso ponto a ponto *point-to-point*. O modem consegue atingir uma velocidade de até 56Kbps. Já a tecnologia ISDN (integrated services digital network – rede digital de serviços integrados) transmite dados

digitais de um ponto final (PC) para a central da companhia telefônica. Sua velocidade de acesso é bem maior do que da ISP, (por exemplo de 128Kbps). A ISDN pode ser considerada como um modem melhorado.

A linha digital assimétrica para assinantes (ADSL), é, conceitualmente, semelhante ao modem discado, transmitem em par de fios da linha telefônica só que atinge uma velocidade de até 8 Mbps. A taxa de transmissão de dados na direção do sistema final doméstico até o roteador da central é menor do que 1 Mbps.

Uma das características da ADSL é que o usuário pode receber/fazer uma chamada telefônica e ao mesmo tempo navegar pela internet.

As redes de acesso de linha híbrida de cabo de fibra e cabo coaxial (HFC) são redes de cabo existentes para as transmissões de TV a cabo. As operadoras de TV a cabo disponibilizam também acesso à internet. A ligação entre a operadora e o usuário é feito por meio de cabo, isto é, sai da operadora vai até um repetidor e deste repetidor é levado o sinal para a casa do usuário.

- **Rede de acesso institucional**, que liga um sistema final de uma empresa ou instituição educacional à rede. Nas redes de acesso corporativo, uma rede local é usada para ligar um sistema final a um roteador de borda. As redes corporativas usam a tecnologia Ethernet. Esta tecnologia permite uma velocidade de 10 Mbps a 1 Gbps, suas ligações são feitas por cabos de par trançado.

- **Rede de acesso móvel**, que liga um sistema final móvel à rede. A rede de acesso móvel usa o espectro de rádio para conectar um sistema final portátil, por exemplo, um laptop com modem

sem fio, a uma estação-base. Esta estação base, por sua vez, está conectada a um roteador de borda de uma rede de dados.

Um padrão emergente para rede de dados sem fio é o pacote de dados celular digital (cellular digital packet data – CDPD). Como o nome sugere, uma rede CDPD opera como uma rede sobreposta (isto é, uma rede virtual separada, menor, como parte da rede maior) na rede de telefonia celular. Assim ela usa o mesmo espectro de rádio do sistema de telefonia celular e opera em velocidade na faixa de dezenas de Kbits por segundo. Como acontece com as redes de acesso a cabo e com Ethernet compartilhada, os sistemas finais CDPD compartilham o meio de transmissão com outros sistemas finais CDPD que estejam dentro da célula coberta por uma estação base. Um protocolo de controle de acesso ao meio é usado para arbitrar o compartilhamento de canal entre os sistemas finais do CDPD.

O sistema CDPD suporta o protocolo IP e, assim, permite que um sistema final IP troque pacotes IP com uma estação base IP por meio de canal sem fio. A rede CDPD não provê nenhum protocolo acima da camada de rede. Do ponto de vista da Internet, a rede CDPD pode ser vista como uma extensão da capacidade de transmitir pacotes IP para um enlace de comunicação sem fio entre um sistema final móvel e um roteador de Internet.

2.3 – COMENTÁRIOS FINAIS

Neste capítulo foram colocados alguns conceitos, como a importância da informação para o crescimento de uma organização. Para que a informação possa ser correta, precisa e rápida, a necessidade de uma rede de computadores na Organização é essencial. Como foi visto, redes são complexas e tecnologia de ponta, o que permite a sua utilização eficiente para a implantação e operação de sistemas de informação. Porém, sistemas de informação geram um alto custo para a organização, isto é, quanto mais eficiente maior o custo. Tendo-se uma eficiência menor, problemas de segurança surgem com mais freqüência e estes problemas de segurança serão vistos no próximo capítulo.

3 – SEGURANÇA: PROBLEMAS E MECANISMOS DE PROTEÇÃO

3.1 – SEGURANÇA: FÍSICA, USUÁRIOS E SOFTWARE

Cuidar da segurança de um servidor é como cuidar de um jardim, ou seja, requer paciência, dedicação e cuidados diários [Brand,2000]. Realmente cuidar da segurança de um servidor não é uma tarefa fácil, e não existe uma receita de bolo para tal tarefa. O que estaremos apresentando são apenas informações que poderão ser utilizadas para que se possa tomar os devidos cuidados na montagem de um servidor.

Problemas de rede podem ser de naturezas diversas: físicas, usuário, software, etc. A seguir falaremos destes problemas e dos mecanismos de segurança em redes de computadores.

3.1.1 – PROBLEMAS FÍSICOS NA SEGURANÇA DE SISTEMAS

Vamos relatar abaixo alguns problemas e suas soluções.

- **ATAQUE BASEADO NO ACESSO FÍSICO AO SERVIDOR**

Este tipo de ataque é um tanto raro, mas se seu servidor estiver localizado em um local com grande fluxo de pessoas, ele será uma vítima em potencial. Se um intruso tiver acesso físico ao seu servidor, ele poderá com um simples boot (iniciar o servidor) comprometer o sistema porque, em determinados sistemas operacionais, como no FreeBSD ou Linux por default, ao se dar um **boot em "single user"**, ou seja, usando a opção *-s* no momento do boot, você terá acesso administrador *root* ao sistema, sem a necessidade de fornecer uma senha e, como podemos perceber, um intruso poderia causar danos irreparáveis ou mesmo furtar informações do servidor. Esta vulnerabilidade pode ser corrigida facilmente, configurando o sistema para solicitar a senha do administrador *root*, no boot em modo *single user*.

Outra possibilidade seria se o intruso puder dar um **boot no servidor através do drive A:**, pois, desse modo, ele poderia utilizar, por exemplo, um disco contendo a imagem do boot.flp e outro com a imagem do fixit.flp que, normalmente, são utilizados na restauração de sistemas danificados, para ter acesso ilegal ao conteúdo do seu HD. Esta vulnerabilidade é contornada, desabilitando na BIOS do servidor, o boot através do drive A:, ou seja, forçando o boot apenas pelo HD. A configuração de uma **senha para o setup da BIOS** também é muito importante, pois o intruso não terá como alterar as configurações.

De nada adiantarão as precauções acima, se os usuários tiverem acesso irrestrito ao servidor. Isso não quer dizer que você deva manter seu servidor isolado em um cofre, mas sim controlar quem tem acesso a ele.

Uma vez tomadas as precauções acima, podemos dizer que o servidor está seguro contra os ataques físicos? Não podemos responder. Dizemos isso porque se o intruso quiser muito os dados armazenados em discos ele vai dar um jeito de pegar. O que queremos dizer com isso é que o intruso sempre **poderá remover o HD do servidor** e levar para "olhar" em outro lugar.

- **ATAQUE BASEADO NA CLONAGEM DE IP´S**

Este ataque é muito comum em redes de computadores, um indivíduo, de posse de um micro, cabo de rede e um número de IP, de um outro micro, pode ter acesso à rede de computadores sem uma autenticação na rede. Isto pode gerar um problema de duplicação de IPs na rede, e também pode causar transtornos para o usuário que possui o registro do IP. Pois quaisquer problemas que houver no IP, é ele quem vai responder. Este caso será visto como um problema em nossa simulação de casos hipotéticos.

3.1.2 –PROBLEMAS REFERENTES A USUÁRIOS NA SEGURANÇA DE SISTEMAS

Para começar podemos dizer que a segurança de um servidor, depende diretamente da escolha de uma boa senha para conta de administrador (root) e para as contas dos demais usuários. Com isso podemos tomar as seguintes precauções, que foram obtidas através de resultados de análises experimentais feitas ao longo do trabalho, Tabela 3.1, quando vamos fazer a escolha de uma senha :

Tabela 3.1 – Recomendações para a escolha de uma senha.

Não utilizar o mesmo nome do seu login;
Não utilizar seu nome ou de qualquer outra pessoa; (principalmente o de sua esposa ou de seus filhos);
Não utilizar nenhuma palavra existente em um dicionário, independente do idioma;
Não utilizar nenhuma informação pessoal, como exemplo: número de RG, CPF, telefone, alguma data importante, etc;
Não utilizar nenhuma seqüência de letras presente em seu teclado, como exemplo lkjhgf;
Não utilizar senhas que contenham apenas números;
Não utilizar senhas com menos de 8 caracteres.

Com as informações acima podemos agora ver as possibilidades para se escolher uma boa senha, apenas seguindo os passos a seguir:

- Procurar utilizar uma mistura de números, letras em caixa alta e baixa e caracteres especiais;
- Escolher os caracteres de sua senha de forma aleatória.

Uma outra forma de se escolher uma boa senha, para aquelas pessoas que tenham dificuldades em escolher uma, seria utilizar senha semi-aleatória, construída a partir de uma frase que só tenha sentido para ele. Por exemplo, tomando a primeira letra de cada palavra da seguinte frase: "Não existe Segurança neste mundo; só Oportunidade.", poderíamos extrair a seguinte senha: NeSnm;sO.

Após a escolha da senha, vamos ver as possibilidades que um intruso poderia ter, para conseguir burlar o sistema.

Existem vários métodos para um intruso conseguir burlar um sistema, alguns deles serão mostrados a seguir:

- **ATAQUE BASEADO EM ENGENHARIA SOCIAL**

Esse tipo de ataque pode não ser muito usual. Mas não podemos garantir que ele não possa acontecer. Infelizmente, a prevenção desse tipo de ataque depende muito mais do bom senso dos usuários que de um administrador. O único jeito de se prevenir esse tipo de ataque seria informar aos usuários, e deixar bem claro, a política adotada pela empresa. Como exemplo, deixar claro que nunca ninguém vai ligar para um usuário e dizer:

> *Olá, aqui é o administrador do sistema XYZ. Estou ligando porque ocorreu um problema em nosso servidor, e seu cadastro foi removido acidentalmente do nosso banco de dados. Eu preciso do login e da senha que senhor(a) utilizava, para que possamos restaurar sua conta.*

São, nestes casos, que um intruso consegue, na base da *boa fé* do usuário, uma brecha para entrada em um sistema.

- **ATAQUE BASEADO EM INFORMAÇÕES OBTIDAS NO "LIXO"**

Esse é um dos tipos de ataque que ninguém acredita que ocorra.

Mas, infelizmente, eles acontecem, e o sucesso desse tipo de ataque se baseia no fato de que muitos usuários, não seguem os conselhos

para a escolha de uma boa senha, como demonstrado acima, e utilizam dados pessoais, estes dados podem ser obtidos pelo intruso, no próprio lixo do usuário, e mais, como nome ou data de aniversário dos filhos, número de telefone, número de documentos, etc.

O único modo de sanar este problema, seria informar aos usuários para que estes escolham uma boa senha.

Existem ainda os usuários que, quando mudam de senha, anotam a nova senha em um papel até que o memorizem, após memorizar a senha, jogam o papel fora, muitas vezes antes de alterarem para uma nova senha.

3.1.3 –PROBLEMAS DE SOFTWARE NA SEGURANÇA DE SISTEMAS

- **ATAQUE ATRAVÉS DE UMA INTERFACE DE REDE (LAN, WAN, INTERNET, ETC)**

Pode-se dizer que esse tipo de ataque é o mais comum, pelo simples fato do intruso poder atacar seu sistema, estando em uma rede local ou até mesmo em um país diferente do alvo.

Um dos grandes problemas em ter um ataque baseado em software, seria ter um serviço rodando em um servidor que estaria com o software desatualizado. Um exemplo pode ser dado, em que um servidor de Web (servidor Apache), que disponibiliza as páginas de um site para a internet, esteja com uma versão desatualizada do software. Disponibilizar sites para a internet abre, no servidor, a porta 80 que, quando liberada se torna uma porta de entrada para intrusos. O software Apache é quem controla tudo o que passa pela porta 80 e se o software estiver desatualizado, ele pode possibilitar a invasão do servidor. Segundo [ARBAUGH, 2000], em um estudo de caso, que visa mostrar as vulnerabilidades do MS-Windows, é mostrada uma preocupação em manter o sistema operacional sempre atualizado.

Este problema é solucionado, quando o administrador fizer atualizações sempre que surgirem novas versões.

Um outro problema existe, quando um software é instalado em um servidor, e contém um erro (bug) de programação que possibilita a invasão no servidor.

A solução deste problema é sempre que possível o administrador estar lendo sobre o assunto e fazendo atualização do software quando sair nova versão ou quando for detectado erro na versão que estiver sendo utilizada.

Com tantos problemas existentes, também há vários mecanismos para melhorar a segurança de redes, veremos alguns no próximo item.

3.2 – MECANISMOS PARA MELHORAR A SEGURANÇA DE REDES

Para se ter segurança em redes de computadores é primordial a utilização de comunicação segura, criptografia, autenticação, integridade e distribuição de chaves e certificação.

Para explicar segurança em redes de computadores, utilizaremos três atores para descrever um cenário de comunicação segura em redes de computadores: Alice, Bob e Trudy, a intrusa.

Neste exemplo, Alice e Bob que vão se comunicar podem ser, dois roteadores que querem trocar tabelas de roteamento com segurança, dois hospedeiros que querem estabelecer uma conexão de transporte segura ou duas aplicações de e-mail.

Privacidade é a palavra-chave neste contexto pois, quando existe uma comunicação entre partes, espera-se: sigilo, autenticação e integridade da mensagem.

3.2.1 - COMUNICAÇÃO SEGURA

Para que Alice e Bob possam comunicar-se com segurança, isto é, para que não exista nenhum intruso (Trudy) para interceptar, ler e registrar qualquer dado que seja transmitido de Alice para Bob, e também para que Bob tenha certeza de que a mensagem que recebeu tenha mesmo sido enviada por Alice, enquanto que Alice quer ter certeza de que a pessoa com quem ela está se comunicando é de fato Bob. Ambos também querem ter certeza de que o conteúdo da mensagem que Alice enviou a Bob não foi alterado no caminho. A seguir, mostraremos as propriedades que garantirão a comunicação segura entre Alice e Bob.

- *SIGILO.*

Uma mensagem que seja enviada por Alice deve chegar até Bob sem que nenhum interceptador consiga entender a mensagem. Pois esta mensagem tem que ser cifrada de alguma maneira para que se a mesma for interceptada ela não possa ser decifrada (entendida) pelo interceptador.

- *AUTENTICAÇÃO.*

A autenticação visa garantir ao remetente que a mensagem que chegou para ele é realmente da pessoa *send* que enviou a mensagem. Se você está tendo uma comunicação pessoal então você não precisa de autenticação, ou melhor, você está autenticando visualmente. Vamos exemplificar: Uma pessoa liga para você e diz que é de seu banco, ele pergunta a você sua conta corrente, senha e saldo, pois o sistema havia dado problema, você com certeza não passaria nenhum dado pois, quem garante que esta pessoa é mesmo de seu banco, neste processo, você fez uma autenticação.

- *INTEGRIDADE DE MENSAGEM.*

Mesmo o remetente como o destinatário se conhecendo, isto é, se autenticando, se validando, eles querem garantir que o conteúdo da mensagem não seja alterado, por acidente ou por má intenção de algum intruso, durante a

transmissão. Quem vai garantir isto são os protocolos de transferência que veremos mais adiante.

Agora iremos discutir o que é um canal inseguro. A que informação Trudy terá acesso e que ações poderão ser tomadas sobre os dados transmitidos.

3.2.2 – PRINCÍPIOS DA CRIPTOGRAFIA

A criptografia vem dos tempos da Grécia, onde existe uma cifra chamada César (devido a Julio César). As modernas técnicas de criptografia, hoje em dia utilizadas em qualquer comunicação, exemplo Internet, darem-se com anos de pesquisa.

As técnicas criptográficas possibilitam que o remetente mascare os dados que serão enviados ao destinatário, para que um intruso não consiga desmascarar, ou entender a mensagem. Sendo assim, apenas o destinatário conseguirá decodificar ou entender a mensagem.

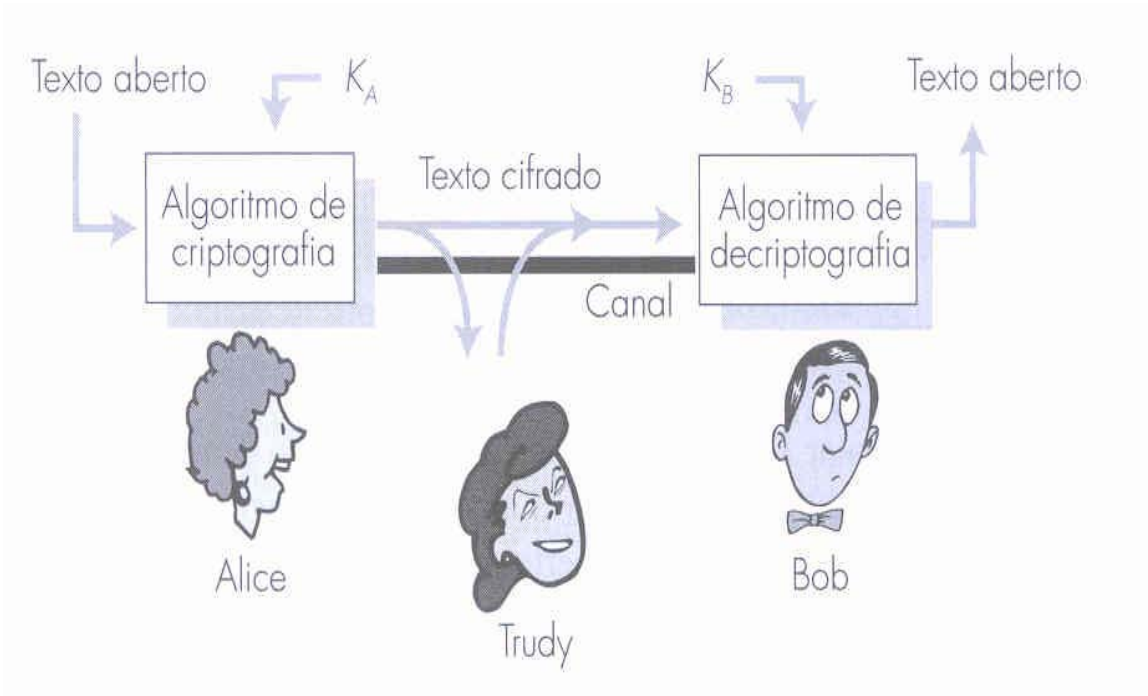


Figura 3.1 – Componentes criptográficos.[KUROSE,2003]

A técnica de codificação é conhecida por todos, padronizada [RFC 1321, RFc 2437, RFC 2420] e disponível para qualquer pessoa, mesmo para Trudy, a intrusa. Já que todos conhecem o método, alguma parte da chave deve ser secreta, senão não nos servia, aí é que entra o segredo da chave.

Alice fornece a chave, K_A, uma seqüência de caracteres, alfanuméricos, para o algoritmo de criptografia. O algoritmo pega como entrada a chave e o texto aberto e os transforma em um texto cifrado como saída. Igualmente, Bob fornecerá uma chave K_B, ao algoritmo de decriptografia, que pega o texto cifrado e a chave de Bob como entrada e transforma o texto em aberto como saída.

Sistemas de chaves simétricas são quando as chaves de Bob e Alice são idênticas e secretas, já nos sistemas de chave pública, é usado um par de chaves, em que uma das chaves é conhecida pelo mundo inteiro, e a outra chave é somente conhecida ou por Bob ou por Alice, ambos não podem conhecer a chave.

3.2.3 – AUTENTICAÇÃO

O processo de autenticação, em nossa vida real, é feito em vários momentos, pessoas que há muito tempo não se encontravam, a um primeiro momento procuram características marcantes na outra pessoa para fazer o reconhecimento, isto é, procuram se lembrar desta pessoa como ela era e fazem o reconhecimento (autenticação). Está é uma autenticação visual, existe também o reconhecimento pela voz, quando falamos por telefone, uma música tocando em um rádio, reconhece-se a voz do cantor.

O processo de autenticação, na rede de computadores, é um pouco diferente da que estamos acostumados, voz, características físicas, etc., quem faz a autenticação são roteadores e processos cliente/servidor, que se autenticam mutuamente. Partes de um protocolo de autenticação, base de mensagens e de dados é que fazem a autenticação.

Agora vamos mostrar o desenvolvimento dos vários tipos de protocolos de autenticação *(pa)*.

O protocolo de autenticação mais simples que existe é o *pa1.0*, quando Bob vai comunicar-se com Alice.

No protocolo *pa2.0* é feita a passagem do número de IP fixo da máquina de Bob, junto com a mensagem.

A senha seria uma alternativa para definitivamente impedir que Trudy não conseguisse se passar por Bob para trocar mensagem com Alice. Mas, *pa3.0,* mesmo utilizando uma senha, não consegue impedir que Trudy consiga comunicar-se com Alice, passando-se por Bob. Trudy simplesmente utilizaria um sniffer (software que vasculha a rede, e descobre senhas, textos) para ler todos os pacotes que passam pela rede e assim pegaria a senha, que não é criptografada. Faria uma mensagem para Alice e passaria com a senha que ela conseguiu na rede.

O protocolo *pa3.1* faz a criptografia da senha, utilizando uma chave simétrica secreta, $K_{A\text{-}B}$, Bob codifica a senha e envia a mensagem para Alice com a senha criptografada.

A senha criptografada impede que Trudy consiga descobrir a senha, mas não garante a autenticação.

O protocolo *pa4.0* utiliza-se de um **nonce**, que é um número que um protocolo vai utilizar apenas uma vez e nunca mais utiliza. A chave simétrica fica assim $K_{A\text{-}B}(R)$.

1- Bob envia uma mensagem para Alice;

2- Alice escolhe um **nonce**, *R*, e envia a Bob;

3- Bob criptografa o **nonce** com a chave simétrica secreta, que combinou com Alice e envia o **nonce** cifrado $K_{A\text{-}B}(R)$ de volta a Alice. Alice então vai poder comunicar-se com Bob, passando mensagens, pois este protocolo garante a Alice que Bob está ao vivo.

4- Alice decifra a mensagem, se o **nonce** decifrado for igual ao **nonce** que enviou a Bob, então ele estará autenticado.

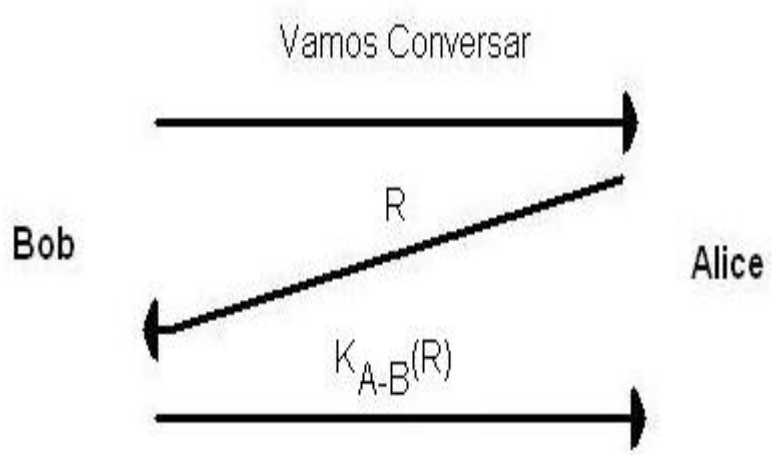


Figura 3.2 – Protocolo *pa4.0*.[KUROSE,2003]

Agora vamos ver como funciona o protocolo de autenticação *pa5.0*, com a criptografia de chaves públicas:

1- Bob envia a mensagem *"Vamos conversar"* para Alice;

2- Alice escolhe um **nonce**, *R*, e envia a Bob;

3- Bob usa seu algoritmo criptográfico privado, com sua chave privada d_A e o **nonce** para gerar o valor $d_A(R)$ (que só ele pode gerar, pois só ele conhece a chave privada) e enviar a Alice.

4- Alice decriptografa a mensagem com a chave pública de Bob, e_A, isto é ela processa $e_A(d_A(R))$. Assim, Alice calcula *R* e autentica Bob.

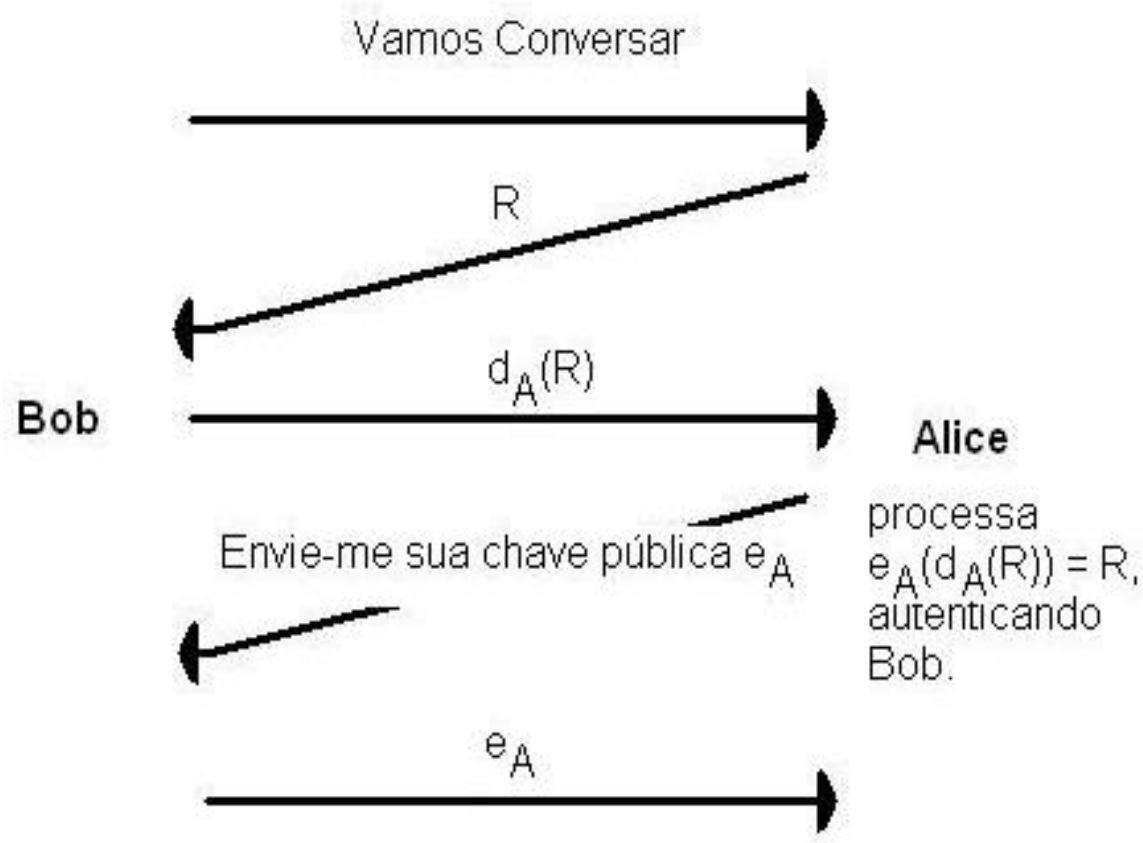


Figura 3.3 – Protocolo *pa5.0*.[KUROSE,2003]

3.2.4 – INTEGRIDADE

No dia a dia, inúmeras vezes você tem que provar que você é você mesmo. Como se faz isto? Através de identificação. Exemplos de identificação, Carteira Nacional de Trânsito, Cédula de Identidade, CIC, assinatura, etc. No meio digital tem que se identificar também, a assinatura digital é a técnica criptográfica usada para fazer esta identificação. Ela deve provar que você é você mesmo, como uma assinatura de próprio punho. A criptografia de chaves públicas pode garantir esta segurança.

3.2.4.1 – COMO GERAR ASSINATURAS DIGITAIS

Vamos partir do princípio que Bob queira comunicar-se com Alice, mas Bob em sua mensagem *m* quer assinar a mensagem. Para Bob assinar esta mensagem ele irá utilizar sua chave criptografada privada d_B para processar $d_B(m)$.

Alice recebe a mensagem, pega a chave pública de Bob, e_B, e aplica a fórmula da assinatura digital $d_B(m)$ associada ao documento *m*. Em outras palavras, ela executa $e_B(d_B(m))$ que é igual a *m*, e produz a mensagem original que Bob a enviou.

Um outro método para gerar assinatura digital é o **resumo de mensagem**, como vimos no exemplo acima, faz-se a codificação e decodificação da mensagem em um todo, gerando um alto processamento. O algoritmo resumo da mensagem pega uma mensagem *m* que Bob vai enviar a Alice, de qualquer tamanho e gera uma impressão digital do comprimento fixo dos dados da mensagem, representada pela fórmula *H(m)*. O algoritmo de resumo de mensagem protege os dados de tal maneira que quando *m* é enviado a Alice, e Trudy intercepta a mensagem, trocando esta mensagem por outra mensagem *m'*, Alice receberá a mensagem e executará a fórmula *H(m)*, mas o resultado não será igual à mensagem original.

O objetivo principal é que Bob execute a fórmula $d_B(H(m))$, para assinatura digital, somente no resumo de mensagem. Sendo que o conjunto *m* (não é criptografada) e $d_B(H(m))$ (criptografada) é a mensagem completa.

Resumo de mensagem é um exemplo de funções de *hash*, que pega uma entrada de dados (mensagem) e a transforma em uma cadeia de tamanho fixo, conhecido como *hash*.

3.2.5 – DISTRIBUIÇÃO DE CHAVES E CERTIFICAÇÃO

Na criptografia de chaves simétricas umas das desvantagens é que as partes comunicantes têm que concordar com sua chave secreta previamente escolhida. Já na criptografia de chaves públicas, não existe este problema, mas há outro que é obter a chave pública verdadeira da outra parte.

O intermediário de confiança pode solucionar o problema da criptografia de chave pública. Já para a criptografia de chave simétrica, este intermediário de confiança é chamado de **central de distribuição de chaves** (*key distribution center* – KDC), esta é uma entidade de rede única e de

confiança com quem os usuários estabelecem uma chave secreta compartilhada.

No caso da criptografia de chaves públicas, o intermediário de confiança é chamado de **autoridade certificadora** (*certification authority* - CA). A CA certifica que uma chave pública pertence a uma determinada entidade (uma pessoa ou uma rede). Isto é, uma vez que uma chave pública é certificada, ela pode ser distribuída de qualquer lugar, incluindo um servidor de chave pública, uma página Web ou um disquete.

3.2.6 - FIREWALL E FERRAMENTAS PARA PROTEÇÃO DA REDE

Aqui utilizaremos o sistema operacional FreeBSD como exemplo, que é o sistema operacional utilizado pela Faculdade de Ciências Médicas (FCM) , uma instituição fictícia, ele disponibiliza o **ipfw** que é um sistema de filtragem de pacotes com base no IP de origem/destino. Ele deverá ser utilizado para restringir os *hosts* (micros) que terão acesso ao servidor, bem como os serviços que cada um poderá acessar.

Se for configurado de maneira adequada, ele será um grande aliado na prevenção de ataques provenientes da rede.

TCP_WRAPPERS + TCP_Dump - A segunda linha de defesa.

O TCP_WRAPERS está disponível no *ports* e no *packages* (sistema de instalação de pacotes encontrado no sistema operacional FreeBSD). Com ele é possível monitorar as conexões feitas no servidor, o horário de cada conexão e o mais importante o IP de origem. Com ele também é possível monitorar e controlar os acessos ao servidor, feitos através de: FINGER, FTP, TELNET, RLOGIN, RSH, EXEC, TFTP, TALK, etc.

O **TCP_Dump** é um utilitário que permite analisar o tráfego de sua rede, um sniffer. O sniffer é um software que consegue ler os dados que passam pela rede sem estarem criptografados. Com estes dois utilitários é possível dizer quem acessou o servidor, de onde ele acessou e o que ele fez.

Para evitar problemas, sempre que o administrador configurar o servidor, é muito importante que **desative todos os serviços que não serão utilizados**, como: rlogin, rsh, tftp. Caso algum usuário necessite utilizar os serviços de rsh, rlogin e telnet, será aconselhável que os substitua pelo ssh (secure shell). Isso lhe dará um pouco mais de segurança quanto à privacidade dos dados trafegados pela rede, pois serão criptografados, diferente de uma sessão telnet, onde os dados não são criptografados.

O **fingerd** é um serviço que pode ser utilizado por um intruso, para descobrir informações de usuários do servidor e se existem contas inativas no servidor, o que seria uma boa ferramenta para o intruso.

Se um usuário for utilizar apenas o serviço de e-mail (POP3, IMAP, etc) do servidor, não forneça a ele uma shell real, defina "nologin" como shell para este usuário, pois deste modo ele não poderá acessar o servidor por telnet, por FTP ou SSH, isto é, o usuário não terá acesso ao servidor.

O comando **last**, é muito útil, pois permite que identifique atividades anormais no servidor, como exemplo, o usuário que acessou o servidor, a máquina de onde ele acessou, data e hora que permaneceu no servidor.

Consultar diariamente os **arquivos de log** do sistema, é um hábito que os administradores devem seguir. Neles teremos registro de tudo o que ocorre no servidor. Se alguém tentar violar o servidor, será nesses *logs* que iremos encontrar os indícios.

Estes são alguns exemplos de problemas mais freqüentes, envolvendo a segurança das informações que trafegam na rede de computadores.

3.3 – COMENTÁRIOS FINAIS

O objetivo desse capítulo é mostrar a real necessidade da utilização de procedimentos e critérios para melhorar a segurança do sistema. Estes critérios vão garantir a fluência e segurança das informações que estiverem trafegando na rede de computadores. Fica claro que os problemas e alternativas para melhoria da segurança são muitos. Infelizmente, apesar de muitos gastos e esforços, nem sempre é possível garantir que um sistema, operando em redes de computadores esteja livre de problemas.

Por isso, alternativas para lidar com essa situação têm sido propostas. Uma delas trata de garantir a sobrevivência da missão crítica de um sistema, e não do sistema como um todo. Esses são alguns dos focos deste trabalho, e será detalhado nos próximos capítulos.

4 – SOBREVIVÊNCIA DE SISTEMAS DE MISSÃO CRÍTICA

Nos tempos atuais, com uma sociedade cada vez mais utilizando o computador, recursos de rede e principalmente conectados à Internet, surge a grande preocupação das Organizações em proteger seus sistemas de rede, seja de negócios, governo ou defesa. Isto é, estas Organizações não mais estão limitadas em uma rede interna, possibilitando assim inúmeras brechas de segurança.

4.1 – CONCEITO DE SOBREVIVÊNCIA

A sobrevivência fornece um novo conceito e uma perspectiva de negócio em segurança que pode nos guiar para uma compreensão melhor da natureza e da estrutura moderna, em sistemas altamente distribuídos, e pode conduzir-nos às soluções dos problemas de segurança que parecem, até hoje, intratáveis.

Muitos negócios têm planos de contingência para tratar de interrupções por causa de desastres naturais ou por acidentes. Embora a maioria dos cyber-ataques seja relativamente menor, em razão de danos naturais, cyber-ataque, num sistema de informação crítica de rede de uma Organização pode causar severos estragos e rompimentos prolongados do negócio. Se cyber-ataque destrói funções críticas do negócio e interrompe os serviços essenciais de que os usuários dependem, a solução está na sobrevivência do negócio.

A sobrevivência surge como uma nova disciplina que se preocupa com a missão crítica da Organização, sendo que depende ou trabalha com a área de segurança. Isto é, para se fazer um planejamento da missão crítica da Organização, será preciso juntar os profissionais não só da área técnica, mas também os gerentes de riscos e da gerência executiva. A Tabela 4.1 [HOWARD,2000] apresenta uma comparação entre técnicas e métodos utilizados pela abordagem convencional (segurança) e pela sobrevivência, respectivamente.

Tabela 4.1 – Técnicas e métodos utilizados em Segurança e Sobrevivência

SEGURANÇA:	**SOBREVIVÊNCIA:**
Modelo Fortress: firewall, política de segurança.	Técnicas de segurança e onde aplicar.
Confiança nos usuários do sistema.	Diversidade, redundância.
Criptografia, autenticação e controle de acesso.	Validação da confiança.
Detecção do intruso (recuperação secundária).	Recuperação (automatizada pela maior parte).
Critério do sucesso: O ataque tem sucesso ou falha.	Gerência de riscos especifica a missão.
	Planejamento da contingência (desastre).
	Critérios do sucesso: Degradação vagarosa. Os serviços essenciais se mantêm. Soluções podem ultrapassar os limites do sistema.

As Organizações, hoje em dia, não se preocupam em colocar sua gerência em contato com os técnicos, deixam a parte de segurança das informações apenas para o pessoal técnico, isto é, o pessoal técnico confia nos equipamentos que estão disponíveis no mercado, para utilizar nas Organizações, ou mesmo em sua própria experiência profissional. Equipamentos únicos (centralizados) podem ser um ponto fraco, podendo

falhar e provocar uma brecha para um intruso, ou até mesmo quebrar. Nas duas situações, a Organização deixará de oferecer serviços.

Com isto pode não ser uma boa forma de conduzir uma Organização, a disciplina de sobrevivência irá mostrar a diferença que existe entre segurança e sobrevivência. A sobrevivência preocupa-se quando a Organização sofre algum tipo de risco, como ataques, acidentes técnicos, acidentes provocados pela natureza, etc. e ela continua fornecendo o mesmo serviço, sem interromper por um período a sua missão.[Howard, 1999] As estratégias adotadas pela sobrevivência é que garantem que a Organização não pare de oferecer seus serviços. A sobrevivência é um algoritmo emergente, pois você pode destruir um número específico de componentes do sistema, e o sistema alternativo cumprirá sua missão.

4.2 – MÉTODO DE ANÁLISE DE SEGURANÇA DE REDE SOB O ENFOQUE DA SOBREVIVÊNCIA.

Aplicação do método Survivable Network Analysis (SNA) é dividida em quatro etapas, conforme mostrado na Figura 4.1. A equipe, no **primeiro passo,** preocupa-se com as definições do sistema que é dividido em: definição das exigências do sistema; definição da arquitetura e descrição do sistema.

No **segundo passo,** a equipe identifica serviços essenciais e recursos, baseada no objetivo da missão e na conseqüência das falhas. Para isso a equipe usa cenários para caracterizar serviços essenciais e uso dos recursos. A definição essencial da potencialidade é dividida em: serviços essenciais/eleição dos recursos/cenário; identificação de componentes essenciais.

No **passo três,** a equipe seleciona cenários de intrusão, com base no ambiente do sistema e em uma avaliação de risco e potencialidades de intruso. A definição da potencialidade de comprometer é avaliada em: seleção do cenário de intrusão e identificação dos componentes comprometidos, em que o intruso pode entrar ou danificar.

No **passo quatro,** a equipe identifica a arquitetura dos componentes do ponto da vulnerabilidade com os componentes que são essenciais e comprometidos, relativos aos passos 2 e 3. A equipe analisa então os pontos de vulnerabilidade dos componentes e suas arquiteturas que suportam as propriedades chaves da sobrevivência que são: **estratégias de resistência, reconhecimento e recuperação,** também conhecidas como “3Rs”.

A análise dos três *Rs* é mostrada em um mapa da sobrevivência. Este mapa é uma matriz que mostra, para cada cenário de intrusão e seus efeitos correspondentes aos pontos de vulnerabilidade, uma tarefa que já é feita e uma recomendação para as três estratégias de resistência, reconhecimento e recuperação.

Um exemplo do mapa gerado a partir desse método pode ser visto na Figura 4.2. No caso, trata-se de um problema referente a um usuário desautorizado, que corrompe o Banco de Dados, levando à perda da confiança em todos os Planos de Tratamento dos clientes de um hospital.

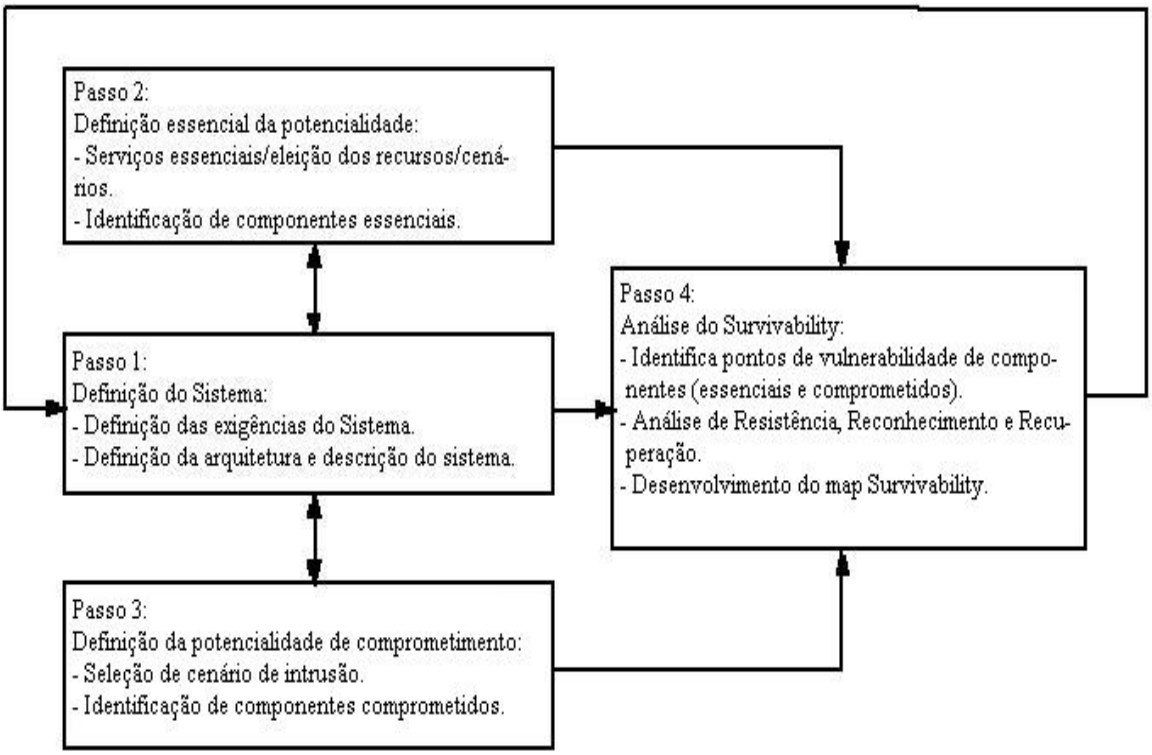


Figura 4.1 – Método de Análise de Rede pela Sobrevivência.[ROBERT,1999]

Cenário de Intrusão	**Estratégia de Resistência**	**Estratégia de Reconhecimento**	**Estratégia de Recuperação**
Um usuário desautorizado corrompe o DB que conduz a perda da confiança em todo o PTs validado de todos os clientes. **Ponto vulnerável:** **Planos de tratamento.**	**Atual:** O modelo de segurança protege o DB encontrando os PTs corrompidos.	**Atual:** Nenhuma, exceto quando um cliente percebe que seu PTs está corrompido.	**Atual:** Recuperar em um Backup ou reconstrução dos PTs de risco.
	Recomendado Implementar replicação dos DBs para validação Transversal (suportada por vários DBs).	**Recomendado** Adicionar e verificar crypto-checksums em TPs no DB.	**Recomendado** Reduzir o ciclo de Backup rapidamente para construir um DB corrompido ,quando detectado.

Figura 4.2 – Exemplo de um mapa gerado pelo método SNA

Em 1999 [ELLISON, 1999] já utilizava a disciplina de sobrevivência, para proteger sistemas críticos. A disciplina ajudava a armar estratégias que manteriam um sistema funcionando mesmo que ele sofresse danos.

Um modelo de ciclo de vida para sistemas sobreviverem foi criado por [MEAD, 2000], apresentado em um Workshop em Boston MA, USA.

Já [FISHER, 1999] utilizou a disciplina de sobrevivência, para um método novo, visando realçar a sobrevivência em sistemas não delimitados. E utilizou a ferramenta EASEL para o projeto e implementação de uma linguagem para simular sistemas distribuídos.

As metodologias utilizadas podem variar, segundo [CRISTIE,2002], em sua pesquisa para verificar a possibilidade de propagação de vírus na rede internet, utilizou a disciplina de sobrevivência, para desenvolver estatísticas de redes válidas para a análise e, como um exemplo de seu uso, aplicando-as à simulação da propagação do vírus na internet.

4.3 – LINGUAGEM DE SIMULAÇÃO EASEL

[EASEL,2000] (Emergent Algorithm Simulation and Language) está sendo desenvolvida como uma ferramenta para a pesquisa em segurança e sobrevivência de sistemas não-delimitados (isto é, atores individuais possuem uma visão restrita da informação do sistema como um todo), sistemas frouxamente acoplados e infra-estrutura altamente distribuídos. Os mecanismos de simulação da ferramenta Easel fornecem um ambiente simulado dos atores frouxamente acoplados, que interagem sem o controle central nem visibilidade global. O controle central e a visibilidade global estão disponíveis para os observadores e os facilitadores fora da simulação.

Como exigências chaves incluem uma semântica de execução consistente com redes não-delimitadas e um rico conjunto de tipos de dados, para suportar amplas aplicações simuladas, bem como, monitoração, levantamento de dados e as análises da simulação.

Outras exigências, são as facilidades de descrição que ajudam usuários a visualizar propriedades e algoritmos emergentes, as diversas características originais, a segurança e a sobrevivência.

Algumas características importantes da linguagem de simulação EASEL: os algoritmos e os protocolos usados nas simulações podem ser idênticos àqueles da aplicação distribuída real, que está sendo simulada. Assim é possível simular sistemas, em níveis diferentes de granularidade, sendo que a linguagem pode processar descrição abstrata dos atores em quaisquer níveis desejado da simulação.

4.4 – COMENTÁRIOS FINAIS

Neste capítulo conhecemos um novo conceito que vem surgindo, de uma nova disciplina que visa manter sempre em funcionamento a missão crítica da Organização, por mais que o sistema tenha sido afetado.

Também foram mostrados os quatro passos da metodologia SNA, que podem ser utilizados como parte do processo de garantia para a sobrevivência de um sistema. Após fazer a análise dos quatro passos, será desenvolvida a tabela, isto é, o mapa de sobrevivência com a análise dos três *Rs*, nas tarefas atuais e recomendadas.

A Linguagem de Simulação EASEL será utilizada para elaborarmos situações que poderão contribuir para um melhoramento da construção do mapa SNA e também a simulação do caso real, que será mostradas no próximo capítulo.

5 – APLICAÇÃO DO MÉTODO SNA COM O AUXÍLIO DE SIMULAÇÕES

5.1 – EXPERIMENTOS A SEREM CONDUZIDOS

Neste capítulo iremos aplicar a metodologia e conceitos anteriormente discutidos em um cenário Real e mais dois cenários hipotéticos (Figura 5.1). Será utilizada a ferramenta de Simulação EASEL para auxiliar a tomada de decisões referente à tabela SNA, utilizada para prever recomendações, visando melhorar a capacidade de sobrevivência do sistema.

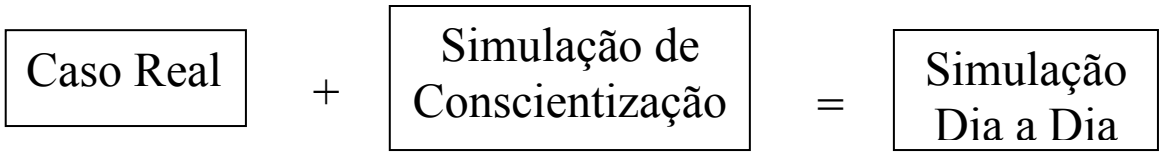


Figura 5.1 – Caso Real e Simulações

No estudo do **Caso Real**, não será feita nenhuma simulação EASEL, e sim a aplicação do conceito de sobrevivência e a metodologia SNA para, posteriormente, montarmos a tabela SNA e coletarmos os dados em um período de tempo.

A **Simulação de Conscientização** fornecerá dados que nos auxiliarão na obtenção de informações referentes ao tempo necessário para efetuar um trabalho de conscientização de usuários, objetivando a diminuição de certas práticas que comprometem a segurança do sistema. O trabalho de conscientização será focado em aspectos e circunstâncias identificados quando da aplicação do método para o caso real.

A situação do caso real, assumindo-se que os usuários estão conscientizados sobre práticas mais seguras, será então simulada com o

auxílio da ferramenta EASEL, etapa essa que chamaremos de **Simulação Dia-a-Dia**. Objetiva-se com isso avaliar a efetividade do trabalho de conscientização em um cenário real anteriormente analisado. Os resultados obtidos na simulação do dia-a-dia, que terá um período maior que os do caso real, serão comparados com os dados do caso real que foram gerados para a construção da tabela.

A seguir, iremos analisar cenários hipotéticos para a aplicação da metodologia, em três casos relacionados à FCM.

5.2 – Cenário Hipotético para Aplicação da Metodologia

Este estudo preocupa-se com aspectos relativos à segurança interna de sistemas de redes de computadores. Analisaremos possibilidades de intrusões que possam afetar nossa Organização, sendo que ela não se preocupa com a segurança interna, o que acontece na maioria dos casos em Organizações.

Um exemplo de intrusão em Organizações seria o intruso destruir, roubar informações da Organização, sendo que o mesmo é um funcionário da própria Organização. Sem medidas de seguranças internas, as Organizações estão tão vulneráveis como se estivessem sem firewall.

Vamos estudar algumas situações em que a primeira é um caso real e as demais serão simulações feitas na ferramenta EASEL.

5.2.1 – Cenário analisado

O trabalho a ser desenvolvido toma como cenário um ambiente hipotético, o qual será identificado como Faculdade de Ciências Médicas (FCM). A FCM tem como objetivo formar alunos nos seguintes cursos: Medicina e Enfermagem, para tanto ela possui uma estrutura de redes de computadores, com servidores e sistemas Tabela 5.1, onde se encontram mais de 800 microcomputadores, para um total de 1.200 usuários, dá suporte para

área Administrativa, docentes e discentes, e mantém três laboratórios de informática para os alunos.

No item 5.2.2 vamos conhecer o parque de equipamentos que a FCM possui, e que é disponibilizado para seus usuários.

TABELA 5.1 - Serviços que a informática presta para a Faculdade:

• Servidor de Web, onde são disponibilizadas as páginas da faculdade, com informações que os departamentos enviam para o web design, para serem publicadas.
• Sistema de ordem de serviço On-Line, onde os departamentos podem solicitar agendamento de equipamentos e laboratórios e consertos de equipamentos.
• Servidor de autenticação e e-mail, eles não podem parar, nem dar problemas, pois as autenticações são feitas nele, tanto de funcionários como de alunos, ele trabalha como PDC (Primary Domain Controller). Nosso meio de comunicação, isto é, a faculdade mantém comunicação com todos os departamentos e alunos, é feito via e-mail.
• Servidor de dados, neste servidor o banco de dados que roda é o MySQL, onde todas as informações são guardadas.
• Servidor de Ensino à Distância, ele mantém o software de ensino à distância e os dados das disciplinas.
• Dois servidores de backup, onde são armazenadas algumas das informações, que temos em outros servidores.

5.2.2– EQUIPAMENTOS UTILIZADOS NA LAN

Os equipamentos utilizados na rede local e servidores são:

- Pentium III com dois Processadores de 1.4, memória RAM de 1 giga, 2 HDs de 140 giga, compartimento para 5 HDs HotSawp e Fita Dat.
- Storadge com capacidade de 1 Tera, para backup dos demais servidores.
- Os equipamentos para estação de trabalho variam de Pentium 100 para Pentium IV. Os micros utilizados como Gateway são Pentium 100. A ligação entre os servidores-micros é feita por cabos ópticos e par trançado, com Switch de 10/100/1000 megabytes.

5.2.3 – PONTOS DEFINIDOS COMO MISSÃO CRÍTICA, A SEREM PRESERVADOS

Existem três pontos que são cruciais para que a missão crítica da FCM possa manter-se em funcionamento. Dois destes pontos foram descritos na Tabela 5.1, de serviços que a informática presta para a faculdade, que são:

- Servidor de autenticação.
- Servidor de e-mail.

Estes dois servidores são importantes para a sobrevivência de nossa missão crítica, eles não podem parar muito menos dar problemas, pois as autenticações dos usuários são feitas no servidor de autenticação, este servidor trabalha como um PDC (Primary Domain Controller), isto é, ele também faz autenticação dos micros.

O servidor de e-mail é o meio de comunicação com os usuários, isto é, a faculdade mantém comunicação com todos os departamentos e alunos.

- A rede de computadores também é muito importante para a FCM.

Sem a rede de computadores não seria possível manter nossa missão crítica e muito menos os dois serviços citados acima.

No item a seguir, iremos ver a montagem dos quatro passos da metodologia SNA e, posteriormente, a montagem da tabela do mapa SNA.

5.3 – ESTADO INICIAL DO SISTEMA DA FCM

A faculdade, no intuito de oferecer vários serviços para seus usuários (que são os alunos, professores, funcionários e convidados), não se preocupou com alguns pontos cruciais que podem prejudicar o nome da instituição, bem como estar vulnerável a qualquer tipo de ataque que possa sofrer de um intruso, ou mesmo um usuário que venha a descobrir brechas no sistema, e não possui nenhum tipo de identificação, autenticação dos usuários de rede interna.

5.3.1 - PONTOS VULNERÁVEIS NOS SISTEMAS DA FCM

Abaixo estão relacionadas situações típicas de problemas de *segurança interna*:

- Um usuário, na ausência de um funcionário, utiliza-se do micro do mesmo para obter informações preciosas que estão no micro.

- Um usuário ou mesmo um intruso cria um e-mail, em um destes servidores de e-mail gratuito, e vai até o laboratório de informática da FCM, que não possui nenhum tipo de identificação dos usuários, utiliza-se de um micro para enviar um e-mail difamador a um destinatário, ou mesmo utiliza-se deste micro para invadir um outro servidor. O que é preocupante, é que no cabeçalho do e-mail e mesmo em logs de

servidores que possam ser invadidos, o IP (identificação do micro na Internet) do micro da FCM estará lá. Como será feita a identificação do usuário que utilizou este micro?

- Um usuário ou mesmo um intruso aproveita-se de uma falha do servidor de autenticação para apossar-se de informações de outros usuários.

- Um intruso usa da técnica de ludibriar um usuário para utilizar-se do micro, enquanto o usuário sai de sua sala.

- Um usuário ou mesmo um intruso conecta um micro na rede da FCM e consegue utilizar-se deste serviço de rede.

5.3.2 – ANÁLISE DO CASO REAL

Utilizando-se o conceito de sobrevivência (Survivability), que visa manter a missão da rede de computadores sempre funcionando, isto é, sem ser comprometida, aplicaremos a metodologia SNA para as três situações. Montaremos a tabela do mapa SNA, podendo levantar e analisar as informações ocorridas nos servidores, no dia-a-dia de trabalho dos usuários e da própria rede. Para que com estas informações, possamos simular na ferramenta EASEL, as recomendações feitas na tabela dos três Rs.

Para a dissertação desenvolvida com a metodologia SNA, serão utilizados quatro tipos de intrusão, com cada uma delas será feita uma análise do dia-a-dia, das tentativas de acesso à rede interna, com êxitos e falhas. Os resultados destas análises servirão para montagem da tabela do mapa SNA. Poderemos recomendar itens para melhoria nas estratégias dos três Rs, que são utilizados para a montagem da tabela do mapa SNA.

Para a análise, foi levado em conta que a população que acessa a rede é de 1.200 usuários, pegas em logs e informações do dia-a-dia, no período de 25 dias úteis. E também foi feito uma média de 563 micros que se conectam diariamente na rede de computadores.

Vamos agora realizar os quatros passos da metodologia SNA, Figura 4.1, para que, posteriormente, possamos montar a tabela de mapeamento SNA.

5.3.2.1 – PASSO 1: DEFINIÇÃO DO SISTEMA

- Definição das exigências do Sistema.

- Manter a rede de computadores interna funcionando e disponibilizando os serviços que nela estão sendo utilizados pelos usuários.

- Definição da arquitetura e descrição do sistema.

- Criação de logins e senhas para acesso e demais serviços disponíveis na rede.
- Cadastros de microcomputadores para acessar a rede de computadores.
- Duplicação de servidores para *backup*.
- Gerenciamento de Alta-Disponibilidade em servidores Web.
- Criação de *scripts* para segurança da Extranet.
- Criação de formulários web para cadastro de usuários.
- Checagem de validação de usuários no banco de dados.
- O sistema funciona com a verificação de vínculo do usuário com a organização, se existir este vínculo é criado um login e senha, para que o mesmo possa ter acesso à rede de computadores, e-mails e outros serviços.
- O usuário de posse de seu login, vai precisar de um microcomputador, que será colocado na rede, só que, para

isto, ele precisa ser liberado no firewall, incluído no servidor de nomes e servidor de autenticação (PDC).

Após estes estágios, o usuário poderá acessar a rede e demais serviços da Organização.

Após as etapas anteriores é possível ter suporte para que se consiga um passo-a-passo sobre acessos na rede tanto de usuários como de micros.

5.3.2.2 - PASSO 2: DEFINIÇÃO DAS POTENCIALIDADES ESSENCIAIS

- Serviços essenciais, eleição dos recursos e cenários.

- Acesso à rede e á internet.

Para que os usuários possam se conectar na rede de computadores é preciso fazer uma identificação, esta identificação também serve para os microcomputadores que se conectarão à rede. Na estratégia de resistência, os usuários terão de ter um login e senha, para que possam ser autenticados no servidor de PDC. Os micros terão que ser liberados no firewall. Na estratégia de recuperação, existem duas possibilidades, a primeira através de um backup em fita e a outra através de um servidor de BDC (Backup Domain Control).

- Acesso ao servidor de e-mail.

A estratégia de reconhecimentos determina, via formulário eletrônico, que o usuário faça sua inscrição para utilização do e-mail, este formulário será encaminhado eletronicamente para o coordenador de curso, se o usuário for aluno, ou para o chefe de departamento se o usuário for funcionário. Tanto o chefe de departamento como coordenador de curso só validarão o cadastro se a estratégia de resistência for verdadeira, isto é, se o usuário tiver algum vínculo com a organização. Na estratégia de recuperação, alguma informação do usuário pode ser encontrada no banco de dados, ou se o servidor de e-mail parar, o servidor redundante entrará automaticamente, através do dispositivo de alta disponibilidade.

Acesso à Extranet.

O servidor onde fica a página web da extranet, não possui informações sobre dados dos usuários para que seja feito autenticação, esta estratégia de reconhecimento é feita através do servidor de PDC, onde um script CGI roda no servidor, verifica-se e autentica-se o usuário, após é redirecionada para o servidor web onde se encontra o site da extranet, com a abertura de uma sessão para que seja impossível um usuário ter acesso diretamente ao site da extranet sem passar pelo script de autenticação do servidor de PDC. A estratégia de resistência implica que usuários cadastrados no servidor de PDC podem utilizar a extranet. Para tornar o servidor web mais resistente a falhas, aplica-se o dispositivo de alta disponibilidade como estratégia de recuperação.

- Intrusão corrompe informações do Banco de Dados.

Os usuários terão acesso ao banco de dados. Para que os registros possam estar confiáveis e não corrompidos, adiciona-se e faz-se a checagem de criptografia e checksums nos registros do banco de dados, como uma estratégia de reconhecimento. Para a estratégia de resistência é recomendado implantar a replicação do banco de dados, com sistema de checagem da validação. Isto possibilita que se reconstrua o banco de dados redundante, apenas no lugar corrompido e rapidamente, sendo esta a estratégia de recuperação.

- Identificação dos componentes essenciais.

- Backup de servidores.
- Backup em fitas.
- Alta disponibilidade nos servidores de e-mail e web.

5.3.2.3 - PASSO 3: DEFINIÇÃO DAS POTENCIALIDADES DE COMPROMETIMENTO

- Seleção de cenários de intrusão.

- Acesso à rede e à internet.

  1- O intruso pode colocar um micro na rede com um IP de um outro micro, podendo navegar na rede.

  2 - Duas máquinas ocupam o mesmo número de IP, na rede ao mesmo tempo.

  3 - Roubo de senha de usuários.

- Acesso ao servidor de e-mail.

  1 - Tanto o coordenador como o chefe de departamento pode liberar um usuário que não tenha vínculo com a Organização (falha humana).

  2 - Intruso rouba senha do usuário.

  3 - Intruso aproveita-se da ausência do usuário e utiliza o micro.

- Acesso à Extranet.

  1 - Intruso rouba senha do usuário.

  2 - Intruso aproveita-se da ausência do usuário e utiliza o micro.

- Intrusão corrompe informações do Banco de Dados.

  1 - Intruso corrompe parte de um registro.

  2 - Intruso rouba senha do usuário.

3 - Intruso aproveita-se da ausência do usuário e utiliza o micro.

- Identificação de componentes comprometidos.

- Acesso à rede e à internet.

  1 – Só saberemos se alguém presenciar o ocorrido.

  2 – Aviso ao administrador de rede através do sistema operacional.

  3 – Saberemos apenas se o usuário nos avisar.

- Acesso ao servidor de e-mail.

  1 – Só será identificado se houver algum problema com a conta do usuário, que foi criada indevidamente.

  2 - Saberemos apenas se o usuário nos avisar.

  3 – Saberemos apenas se o usuário presenciar o intruso utilizando sua conta, ou se surgir algum problema com a conta que foi utilizada indevidamente.

- Acesso à Extranet.

  1 - Saberemos apenas se o usuário nos avisar.

  2 – Saberemos apenas se o usuário presenciar o intruso utilizando sua conta, ou se surgir algum problema com a conta que foi utilizada indevidamente.

- Intrusão corrompe informações do Banco de Dados.

  1 – De posse de uma conta de usuário no sistema, o intruso corrompe parte de um registro.

  2 - Saberemos apenas se o usuário nos avisar.

5.3.2.4 - PASSO 4: ANÁLISE DA SOBREVIVÊNCIA

- Identificar pontos de vulnerabilidade de componentes (essenciais e comprometidos).

Essenciais:

- Recuperação de backup não funciona.
- Sistema de alta-disponibilidade não funciona.

Comprometidos:

- Com a senha de outro usuário, pode-se comprometer a rede e banco de dados.
- Com IP´s duplicados, é necessário um prazo de dois dias para encontrarem o micro que está utilizando o outro IP.

- Análise de Resistência, Reconhecimento e Recuperação.

- Desenvolvimento do mapa de Sobrevivência.

5.3.2.5 – MONTAGEM DO MAPA SNA

Vamos analisar o sistema de rede de computadores da FCM sob o conceito de sobrevivência (survivability). Com isto, utilizaremos os dados conseguidos na dissertação da metodologia SNA, para fazermos o mapeamento com a estratégia dos 3 R´s. O mapa resultante para o cenário analisado encontra-se na Tabela 5.2.

Tabela 5.2 – Mapa gerado pelo método SNA para o cenário analisado.

Cenário de Intrusão	**Estratégia de Resistência**	**Estratégia de Reconhecimento**	**Estratégia de Recuperação**
Impedir o acesso à rede Internet.	*Atual:* Nenhum	*Atual:* Nenhuma	*Atual:* Nenhuma
	Recomendado Checar Login e senha do usuário para autenticar na rede.	*Recomendado* Identificar usuário e micro por um servidor de PDC.	*Recomendado* Backup em fita ou outro servidor, do arquivo que contenha os usuários e micros registrados.
Servidor de e-mail	*Atual* O usuário tem que ter algum vínculo com a FCM.	*Atual* Formulário assinado pelo coordenador de curso ou chefe responsável.	*Atual* Em arquivos, de formulários de papel
	Recomendado Verificar via formulário web o banco de dados dos Recursos Humanos.	*Recomendado* Preencher formulário web, enviando e-mail ao coordenador ou chefe responsável, para aprovação do mesmo. Arquivando em um banco de dados	*Recomendado* Backup do banco de dados.
Extranet	*Atual* Firewall	*Atual* Nenhum	*Atual* Nenhum
	Recomendado Além do Firewall, serviço de autenticação dos usuários internos em um servidor PDC.	*Recomendado* Script cgi, para verificar usuários que estão cadastrados no servidor PDC.	*Recomendado* Replicação do servidor web.
Corromper Informações do Banco de Dados	*Atual* Nenhuma	*Atual* Nenhuma	*Atual* Localizar no backup e restaurar, ou arrumar diretamente no registro manualmente.
	Recomendado Implementar replicação do DB, com sistema de checagem para validação	*Recomendado* Adicionar e checar criptografia, *checksums* em registros no DB.	*Recomendado* Em um backup redundante reconstruir o DB apenas no lugar corrompido, rapidamente.

5.3.2.6 – COMENTÁRIOS FINAIS

O cenário do caso Real foi monitorado durante o período de 25 dias úteis (de 06-10-2005 a 17-11-2005). O sistema recebeu quatro ataques (16%). Obteve-se nas estratégias de Resistência, Reconhecimento e Recuperação uma eficiência de 100%. Apesar de uma eficácia de 100% do sistema, foi detectado um tempo de 8% no período, para localização do microcomputador com problema. Há de se considerar que isto está devidamente relacionado com o tempo de permanência do microcomputador, com IP duplicado, ligado na rede, distância física e agilidades na locomoção de pessoal técnico.

Considerando que se tem 1200 usuários, um parque de 563 microcomputadores, no período de 25 dias úteis os ataques foram poucos, não comprometendo a rede interna, e demonstrando a eficiência das recomendações utilizadas.

5.4 – SIMULAÇÃO DE CONSCIENTIZAÇÃO

Iremos agora simular com a ferramenta EASEL, o programa (Anexo 1) que será o mesmo exemplo do item 5.2, só que com o intuito de saber quanto tempo será necessário para fazer um trabalho de conscientização com os usuários da FCM, em que poderemos verificar se as recomendações feitas no mapa de sobrevivência, também irão dar resultados para esta situação.

Sabendo que, na situação real, tivemos problemas de ordem humana, vamos simular um período de três dias para resolvermos os problemas com senhas e IP´s, fazendo nestes dias um trabalho de conscientização com os nossos usuários.

5.4.1 – CENÁRIO ANALISADO

Vamos utilizar o mesmo cenário descrito no item 5.2.1.

Pretendemos com a utilização deste cenário, verificar qual período seria necessário para fazermos um trabalho de conscientização, da utilização da rede de computadores na FCM, com o objetivo de melhorar aspectos de segurança interna, e garantir a sobrevivência.

5.4.2 – APLICAÇÃO DA SIMULAÇÃO COM A FERRAMENTA EASEL

Como descrito no cenário anterior, utilizaremos duas variáveis que terão maior importância em nossa simulação, pois foi na duplicação de IP´s e nos problemas com senhas que tivemos maiores problemas.

Vamos assumir que, no período dos três dias, tempo necessário para recuperarmos a tentativa de intrusão com senhas ou IP´s duplicados, será feito um trabalho de conscientização com os usuários da rede de computadores. Tornando assim no prazo de três dias, mais o trabalho de conscientização de usuários, os micros passarão a recuperados e usuários informados, sem mais o perigo destes problemas comprometerem nossa rede de computadores. Fazendo também este trabalho nos departamentos que não tiveram nenhum dos problemas acima.

Iremos identificar estes dois problemas (tipos) na simulação como sendo: micros para problemas com IP´s e usuários para problemas referente a senhas.

- Problemas com micros;

Para demonstrarmos a simulação, precisaremos de algumas variáveis, em que o número máximo de dias que uma intrusão poderá ficar na estratégia de resistência é de dez dias. E para a estratégia de reconhecimento é de quatro dias. Este total de 14 dias foram propostos devido ao levantamento que fizemos em comparação com o caso Real. O cálculo foi feito da seguinte

forma: é sabido que, no caso Real, em 25 dias obtivemos 4 problemas; aplicando-se a regra de três, temos que em 14 dias vezes 100, dividido pelos 25 dias do caso Real chegamos ao percentual de 56. Os 4 problemas que tivemos, no caso Real, vezes 56%, chega-se a 2,24 problemas no período de 14 dias. Este valor encontrado da quantidade de problemas é suportada pela equipe técnica da FCM. Estas quantidades de dias foram baseadas, nas condições de tamanho físico, quantidade de funcionários e estrutura de equipamentos que a FCM possui, variando para outras Organizações. Sendo que a distância máxima para a estratégia de reconhecimento será de 62 vizinhos, em virtude das sub-redes que estamos utilizando, apenas 62 micros por grupos.

Se a permanência de uma intrusão na estratégia de resistência for maior que 80% do período de 10 dias, então ele passa para a estratégia de reconhecimento, e se for menor ele passa para a estratégia de recuperação.

Mas, se a intrusão ficar 92% do período de quatro dias na estratégia de reconhecimento, ela passará para *status* de rede parcialmente comprometida. Esta porcentagem de 8% foi obtida, aplicando-se a regra de três, na situação real, em 25 dias, que demorou quatro dias para resolver um problema. Como foi feito um treinamento com os técnicos para a resolução deste problema, baixamos para 2 dias a resolução do problema. Assim, aplicando a regra de três, 25 dias equivalem 100%, dois dias são iguais a 8%.

Sendo que se no período de três dias, o comprometimento parcial da rede, for maior que 10% de um total de 563 micros, ela tornará a rede comprometida, impedindo assim de se cumprir a missão de sobrevivência. Senão, automaticamente, ele passa para a estratégia de recuperação.

A seguir, na Tabela 5.3, é mostrado o período de dias que a porcentagem de comprometimento da rede foi maior.

Dias	reconhecimento	resistência	recuperação	Comprometi-mento Parcial	% Compro-metimento
43	27	66	237	4	0.71048%
44	25	68	247	5	0.8881%
45	23	66	259	5	0.8881%
46	23	55	276	3	0.53286%
47	23	45	291	3	0.53286%
48	21	43	300	2	0.35524%
49	17	39	311	3	0.53286%
50	10	43	321	3	0.53286%
51	18	40	325	2	0.35524%
52	18	41	334	2	0.35524%
53	14	40	348	1	0.17762%

Tabela de Micros

Tabela 5.3 – Problemas com Micros

- Problemas com usuários

Para os problemas com os usuários assumiremos os mesmo dados utilizados nos problemas com micros. Só que a porcentagem de 10% será sobre o valor de 1.200 usuários, para o período de três dias no comprometimento parcial da rede. A Tabela 5.4 mostra dados do período com maior comprometimento da rede de computadores.

Dias	Reconhecimento	resistência	recuperação	Comprometi-mento Parcial	% Compro-metimento
43	63	167	784	8	0.66667%
44	55	154	824	8	0.66667%
45	56	152	850	4	0.33333%
46	56	132	874	8	0.66667%
47	63	115	899	9	0.75%
48	50	108	928	13	1.08333%
49	42	104	964	7	0.58333%
50	49	86	987	4	0.33333%
51	37	88	1015	2	0.16667%
52	46	77	1026	2	0.16667%
53	39	70	1047	6	0.5%

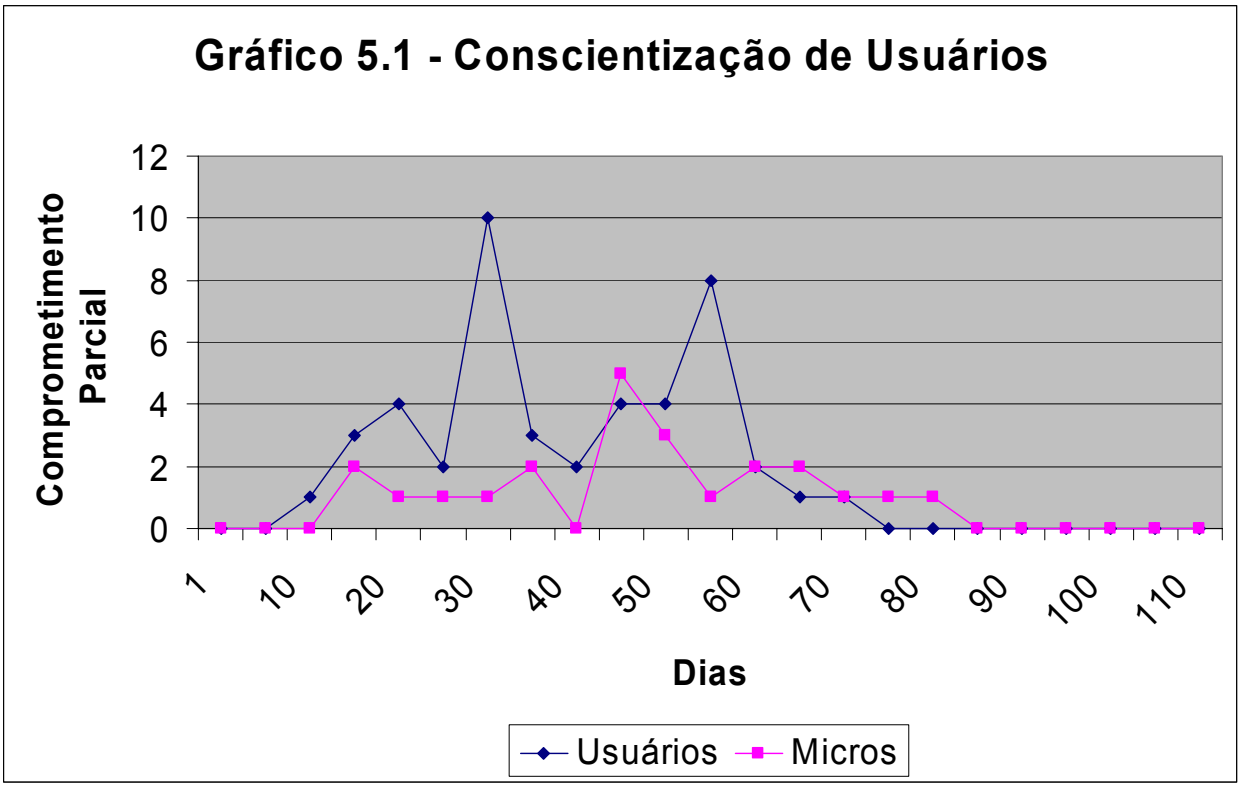
Tabela de Usuários

Tabela 5.4 – Problemas com usuários

Agora que já temos as variáveis e porcentagens dos problemas descritos acima, vamos descrever as variáveis em comum.

O comprometimento da rede de computadores e, consequentemente, o da missão também dar-se-á, com a somatória das porcentagens do comprometimento parcial, nos dois problemas de micros e usuários, isto é, se a porcentagem ultrapassar os 10%.

Nos resultados obtidos no item 5.3.2.6, que são dados reais, temos dados de 25 dias, em que utilizamos 563 micros e 1.200 usuários. E, nestes 25 dias, tivemos três problemas na estratégia de reconhecimento relacionados a problemas de micro e um problema na estratégia de resistência, relacionado a problemas de usuários.



Podemos verificar que para a simulação de conscientização, o Gráfico 5.1 mostra que tivemos apenas dez comprometimentos parciais de usuários e cinco comprometimentos parciais de micros.

5.5 – SIMULAÇÃO DIA-A-DIA

Objetivos: - Usuários já conscientizados;

- Equipe de suporte treinada para os dois problemas de senhas e IP´s.

Com base nestes objetivos, podemos dar início à próxima simulação. Sendo que neste cenário, a simulação não terá fim, será contínua, isto é, programamos a simulação para que não tenha apenas uma estratégia de recuperação, ela terá também simulações de ataques aleatórios. A situação de intrusão que for recuperada pela estratégia de recuperação, poderá voltar como uma situação de intrusão novamente, isto possibilita uma simulação sem fim, proporcionando assim, uma situação parecida com a do dia-a-dia real.

Tendo os objetivos, aplicaremos, a seguir, a simulação do dia-a-dia, para podermos comparar os resultados com o caso Real.

5.5.1 – APLICAÇÃO DA SIMULAÇÃO COM A FERRAMENTA EASEL

Nesta simulação, estaremos nos baseando, nos mesmos dados da simulação anterior. A diferença é que entre a estratégia de resistência e estratégia de reconhecimento, o programa (Anexo 2) gera um valor aleatório entre zero (0) e mil (1000). Com este valor é verificada uma condição, se o valor for menor que dois (2) ele passará para a estratégia de resistência, senão continuará a seqüência lógica do programa, proporcionando o surgimento de uma nova tentativa de intrusão.

Com isto faremos com que a simulação se torne infinita, podendo assim simular uma quantidade de dias que seja do agrado do responsável pela simulação. Ou a própria Organização quer saber se em cinco anos, os requisitos que ela já possui como, equipamento, pessoal, estratégias adotadas, irão causar algum risco para a mesma.

A simulação que fizemos, neste item, não causou nenhum risco para a Organização, como é verificado nas Tabelas 5.5 e 5.6.

Tabela 5.5 – Problemas com micros

Dias	reconhecimento	resistência	recuperação	Comprometi-mento Parcial	% Compro-metimento
114	2	10	28	0	0,00%
115	0	10	30	0	0,00%
116	2	6	30	2	0,71%
117	2	6	30	2	0,71%
118	2	6	30	2	0,71%
119	0	0	40	0	0,00%
120	0	0	40	0	0,00%
121	0	1	40	0	0,00%
122	0	1	40	0	0,00%

Tabela de Micros

Nestes 200 dias de simulação, podemos dividi-lo em oito meses (períodos), isto é, tomando por base 25 dias úteis como foi feito no caso real.

Sendo que nestes oito meses podemos verificar que tivemos 32 casos de comprometimento parcial, sendo que 17 casos foram problemas referentes a usuários e 15 casos foram problemas referentes a micros.

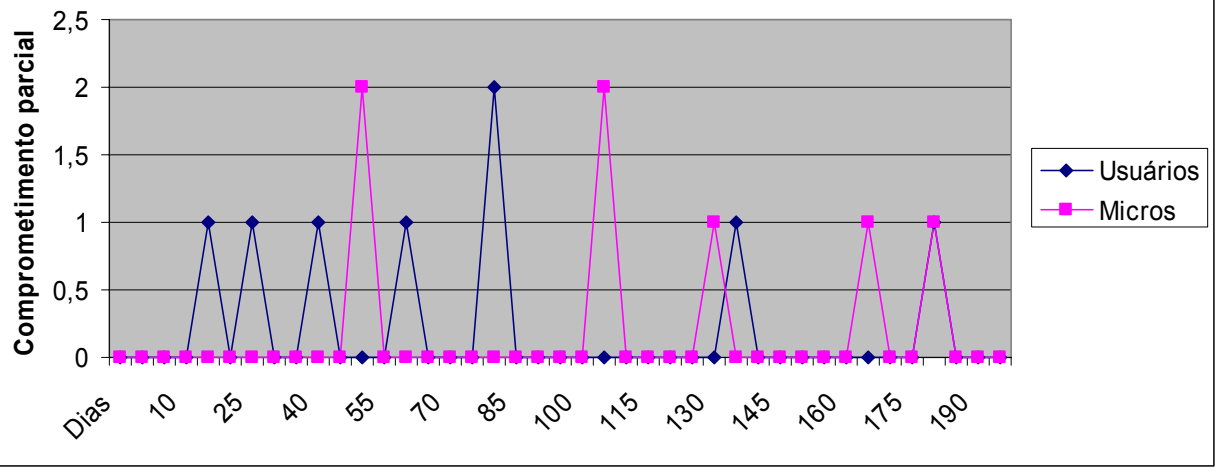
Tabela 5.6 – Problemas com usuários

Dias	reconhecimento	resistência	recuperação	Comprometi-mento Parcial	% Compro-metimento
114	0	12	23	0	0,00%
115	0	12	23	0	0,00%
116	6	0	23	6	0,50%
117	6	0	23	6	0,50%
118	6	0	23	6	0,50%
119	0	0	35	0	0,00%
120	0	0	35	0	0,00%
121	1	0	35	0	0,00%
122	1	0	35	0	0,00%

Tabela de Usuários

Se observarmos o Gráfico 5.2, o comprometimento parcial tanto para usuários como para micro, em um período de duzentos dias não ultrapassa dois problemas por dia.

Gráfico 5.2 - Simulação Hipotética dia a dia



O resultado da simulação forneceu-nos um total de 200 dias e nestes dias foram simuladas intrusões que atenderam a três estratégias, chegando ao comprometimento da rede em 0,50% por usuários e 0,71% por micros. A somatória do comprometimento resultou em 1,21%, o que não prejudicaria o funcionamento da rede de computadores e a sobrevivência do sistema.

5.6 – ANÁLISE DAS SIMULAÇÕES

As simulações realizadas, acima, visam testar as recomendações feitas na Tabela 5.2. Para tanto foram realizadas duas simulações que visam demonstrar, aproximadamente, que é possível utilizar a metodologia SNA e simular tarefas com a ferramenta EASEL, para que sejam aplicadas no dia-a-dia de uma Organização.

No item 5.4, é possível chegar a duas conclusões: Analisando os resultados obtidos na simulação com a ferramenta EASEL, podemos verificar que no dia 48, Tabelas 5.2 e 5.3, a somatória dos comprometimentos parciais, que não poderiam passar de 10%, não interferiram no comprometimento da

rede de computadores, pois atingiram apenas 1,61% e, ao total de 112 dias, todos os usuários foram comunicados, concluindo assim a estratégia de conscientização. Com estes fatores, a missão de sobrevivência para a continuidade da rede de computadores teve picos de problemas, Gráfico 5.1, mas no final foi normalizado e mostra que funcionou.

Com o trabalho de conscientização, pretende-se também que o índice de problemas do tipo de senhas e IP´s, diminua mais.

Já na simulação do dia-a-dia da rede de computadores da Organização FCM, item 5.5, o período de 200 dias, é transformado em oito meses (períodos). Pois dividimos os 200 dias por 25 dias, os mesmos 25 dias úteis que utilizamos no caso real.

Tabela 5.7 – Comparação de dados reais e simulados.

	Problemas em 25 dias	
Dados	***micros***	***usuários***
Real	3	1
Simulação	1,88	2,13

Com base na Tabela 5.7, que contém valores dos dados reais e dados utilizados na simulação do dia-a-dia feita pela ferramenta EASEL, em um período de 25 dias úteis, podemos verificar que a diferença entre dados reais e simulados, são pequenas, por se tratar de um período extremamente curto da análise dos 25 dias úteis do caso real. O que nos motiva é que, em períodos maiores, períodos de dois anos, poderíamos diminuir e detectar possíveis intrusões internas com mais facilidade, sendo que com o trabalho de conscientização e mapeamento das áreas com maiores índices de problemas, é possível combater e diminuir os problemas da Tabela 5.7, visto que em períodos maiores, Gráfico 5.2, no dia-a-dia da Organização é possível notar uma variação, de período para período, nos problemas que surgem na rede de computadores.

5.7 – AVALIAÇÃO DA APLICAÇÃO DA METODOLOGIA

O método SNA (Survivable Network Analysis method) foi aplicado em um caso real, onde o resultado foi satisfatório, pois na Tabela 5.2 com os quatro cenários de intrusão, as recomendações feitas foram atendidas nas três estratégias de resistência, reconhecimento e recuperação. Apesar de termos quatro problemas de ordem humana, é possível refazer e ir melhorando a Tabela 5.2. Nas simulações realizadas com a ferramenta EASEL, conseguimos chegar a resultados que se aproximam da realidade, como mostrado na Tabela 5.7.

E com a simulação realizada no item 5.4, podemos constatar que a utilização da ferramenta EASEL pode melhorar e muito o preenchimento da tabela do método SNA. O preenchimento da tabela poderia prever o trabalho de conscientização dos usuários e mapeamento de áreas com maior índice de problemas encontrados na rede de computadores da Faculdade de Ciências Médicas – FCM.

5.8 – COMENTÁRIOS FINAIS

Neste capítulo mostramos que através da ferramenta EASEL é possível montar simulações, através dos mapas de sobrevivência que criamos com a metodologia SNA, e fazer simulações como as do item 5.4, que nos darão um período aproximado de tempo para a execução de uma tarefa, tornando assim mais fácil de programar um trabalho. Esta programação, isto é, este cronograma, poderá ser montado, através das simulações das tarefas, proporcionando uma estimativa de tempo necessário para a execução do trabalho a ser realizado, e assim melhor definir procedimentos e decisões que garantam a sobrevivência dos sistemas de missão crítica.

6 – CONCLUSÕES FINAIS

Em nossos estudos, diante das simulações que foram realizadas e o estudo de caso real, podemos concluir que a metodologia SNA e a ferramenta de simulação EASEL podem ser utilizadas em conjunto, pois ambas podem proporcionar varias modalidades de situações, isto é possibilitam a simulação de estudos de casos diversos. Estas situações podem ser de casos que tenham que ser implantados na realidade da Organização, mas que, por algum motivo, custo, tempo, possa impedir, ou mesmo que sejam inviáveis para a realidade.

A utilização da metodologia SNA em conjunto com a ferramenta de simulação EASEL, demonstrou que pode suprir os motivos acima, pois pode ajudar nas recomendações para a montagem das tabelas e as simulações das mesmas, sem a necessidade da implantação do caso na realidade, proporcionando à Organização um baixo custo em projetos e riscos que a Organização poderia vir a sofrer.

Recomendamos que em ambientes que contenham o fator humano, seja sempre dada atenção especial a este, devendo ser analisado e estudado para que algumas providências sejam tomadas, visando evitar no futuro, interferências no planejamento das tarefas que surgirão na montagem do mapa SNA.

Agora, disponibilizaremos algumas experiências e orientações para a aplicação da metodologia e da ferramenta de simulação EASEL:

Metodologia SNA

- Sempre que for fazer um planejamento para missão crítica de uma Organização, procure juntar os profissionais das gerências de risco e executiva e a área técnica, por mais difícil que seja. As informações passadas por eles são de fundamental importância para se montar o mapa SNA.

- Monte antes uma tabela do mapa SNA, mesmo que não seja a definitiva, após, comece a escrever os quatro passos da metodologia SNA. Esta montagem antecipada do mapa irá facilitar na dissertação dos passos e, posteriormente, já com a dissertação dos quatro passos em mãos, construa o mapa SNA através dela.

Ferramenta de Simulação EASEL

- Tivemos enormes problemas com a ferramenta de simulação EASEL. Esta ferramenta só funciona em sistema operacional macintosh OS X, ou maior. Este sistema operacional não é muito popular; para funcionar ele precisa de um hardware (micro) especifico, este hardware tem um valor bem maior do que o microcomputador de plataforma Intel. Por isto, não é fácil encontrar alguém que o tenha.

Como trabalhos futuros poderiam ser relacionados:

- Trabalho de coleta de dados do caso real, em períodos maiores.

- Montar um ciclo redundante, na montagem do mapa SNA. Isto é, fazer várias passagens pela Figura 4.1.

- Antes de montar uma estratégia para a missão crítica, fazer simulações com aspectos que possam influenciar na montagem da estratégia da missão crítica.

- Implantar a ferramenta EASEL para rodar na plataforma Intel com sistema operacional LINUX, podendo ser mais explorada em virtude da popularidade e gratuidade deste sistema operacional.

REFERÊNCIAS BIBLIOGRÁFICAS

ARBAUGH, W.A., FITHEN W.L., and McHUGH J. Windows of vulnerability: a case study analysis. IEEE Computer, Vol. 33, No. 12, Dec. 2000.

BRAND, Edson; Um guia para os iniciantes no mundo do FreeBSD - Cuidados básicos com a segurança de seu servidor FreeBSD; 2000; http://www.freebsd.ag.com.br/sessao10_1.html Acessado em 26/05/2006

CIDRAL, A.; Kemczinski, A. Proposta de Perfil do egresso do Bacharelado em Sistemas de Informação do Currículo de Referência 2000 da SBC. Porto Alegre: SBC, 2000. Disponível em http://www.inf.pucrs.br/~duncan/sbc_cr_si.html. Acessado em 26/05/2006

CRISTIE M. Alan; Network Survivability Analysis Using Easel. CMU/SEI-2002-TR-039, ESC-TR-2002-039. http://www.cert.org/archive/pdf/02tr039.pdf Acessado em 26/05/2006

EASEL, (Emergent Algorithm Simulation and Language); http://www.sei.cmu.edu/community/easel/ Acessado em 26/05/2006

ELLISON J. Robert, FISHER A. David, LINGER C. Richard, LIPSON F. Howard, LONGSTAFF A. Thomas, MEAD R. Nancy; Survivability: Protecting Your Critical Systems. Published in Internet Computing, November/December, IEEE, 1999.

FISHER A. David, LIPSON F. Howard, Emergent Algorithms: A New Method for Enhancing Survivability in Unbounded Systems. CERT Coordination Center Software Engineering Institute. Published in the Proceedings of the Hawaii International Conference On System Sciences, IEEE, 1999.

FISHER, A David; Design and Implementation of EASEL. A Language for Simulating Highly Distributed Systems.. Carnegie Mellon University Pittsburgh, PA. http://www.cert.org/archive/pdf/design-easel.pdf. Acessado em 26/05/2006

HOWARD F. Lipson & David A. Fisher. “Survivability — A New Technical and Business Perspective on Security; 2000.” http://www.cert.org/archive/pdf/busperspec.pdf Acessado em 26/05/2006

KUROSE, James F.; ROSS Keith W.; Redes de Computadores e a Internet Uma Nova Abordagem; Tradução Arlete Simille Marques; Revisão Técnica Wagner Luiz Zucchi; São Paulo 2003; Addison Wesley.

MEAD, R. Nancy; ELLISON Robert; LINGER C. Richard;LIPSON F. Howard; MCHUGH John; Life-Cycle Models for Survivable Systems. Published in the Proceedings of the Third Information Survivability Workshop (ISW-2000) October 24-26, 2000 Boston MA, USA.

MEC; SESu-MEC. Diretrizes Curriculares para cursos da área de Computação e Informática; Brasília - DF: MEC, 1998; http://www.mec.gov.br/sesu/ftp/curdiretriz/computacao/co_diretriz.rtf. Acessado em 26/05/2006

ROBERT, J. Ellison; Richard C. Linger; Thomas Longstaff and Nancy R. Mead.; “Survivable Network System Analysis: A Case Study”. http://www.cert.org/archive/pdf/network-analysis.pdf Acessado em 26/05/2006

SOARES, Luiz Fernando Gomes; LEMOS, Guido; Redes de Computadores/Das Lans, Mans e Wans as redes ATM; Rio de Janeiro 1995; Editora Campus.

TANENBAUM, Andrew S.; Redes de Computadores; 3ª edição; Rio de Janeiro 1997; Editora Campus.

BIBLIOGRAFIA

CERT; http://www.cert.org Acessado em 26/05/2006

CROSS E. Stephen; Cyber Security. March 1, 2000. http://www.cert.org/congressional_testimony/Cross_testimony_Mar2000.html Acessado em 26/05/2006

MOORE P. Andrew; Security Requirements Engineering through Iterative Intrusion-Aware Design. http://www.cert.org/archive/pdf/req_position.pdf Acessado em 26/05/2006

Nist, 1999, National Institute of Standards and Technology. "Announcing Draft Federal Information Processing Standard (FIPS) 46-3, Data Encryption Standard (DES), and Request for Comments" http://csrc.nist.gov/cryptval/des/fr990115.htm. Acessado em 26/05/2006

PETHIA, D. Richard; Computer Security. March 9, 2000. http://www.cert.org/congressional_testimony/Pethia_testimony_Mar9.html Acessado em 26/05/2006

RSA Key,1999, RSA Laboratories. "How large a key should be used in the RSA cryptosystem?" http://www.rsasecurity.com/rsalabs/faq/3-1-5.html. Acessado em 26/05/2006

SEI; www.sei.cmu.edu/community/easel/ Acessado em 26/05/2006

TEIXEIRA Jr., José Helvécio; SUAVE, Jacques Philippe; Redes de Computadores/Serviços, Administração e Segurança; São Paulo 1999; Editora Makron Books

 # Anexo 1
#
#
##
 # ########## Simulação
 # ########## Trabalho de Conscientização ###############

##

```
include "::libraries:physical.easel";



state_type: type is enum(normal, resistencia, reconhecimento, morto,
recuperacao);
epi: simulation type is
        vw:: view := ?;
        micros:: list := new list any;
        max_resistencia:: number := 10.0;
        max_reconhecimento:: number := 4.0;
        usuarios:: list := new list any;



micro(id: int, initial_state: state_type, num_users:int,um_micros:int):
actor type is
        state:: state_type := initial_state;
        simTime:: number := 0.0; # dias
        flag:: int := 0;
        dias_d:: number := 0.0;    # dias comprometido
        distMax:: number := 62.0;          # maxima distância reconhecimento
        infNeighbors:: int := 0;     # contador de vizinhos
        xHome:: number := ?;
        yHome:: number := ?;


        if initial_state = normal then
                xHome:= random(uniform, 10.0, 800.0);
                yHome := random(uniform, 10.0, 500.0);
        else
                xHome:= random(uniform, 10.0, 600.0);
                yHome := random(uniform, 10.0, 200.0);
                simTime := clock() + random(uniform, 0.0,
sim.max_reconhecimento); # configuração para o período de 0 - 4 dias

        xloc:: number := 1.1*xHome;
        yloc:: number := 1.1*yHome;
        myColor:: pattern := (green);
```

```
        if initial_state = normal then
                myColor := (yellow);
        if initial_state = reconhecimento then
                myColor := (red);

        depict(sim.vw, var offset_by(paint(circle(0.0, 0.0, 10.0), var myColor),
var xloc, var yloc));

        wait 1.0;

        for every true do
                wait 1.0;
                changePosition(self);
                if state = normal then
                        myColor := (yellow);
                        infNeighbors := 0;
                        for p: each sim.micros do
                                if p.state = reconhecimento then
                                        if get_distance(xloc, yloc, p.xloc,
p.yloc) < distMax then
                                                infNeighbors :=
infNeighbors + 1;

                        if infNeighbors > random(uniform, 0.0, 6.0) then
                                state := resistencia;
                                simTime := clock() + random(uniform, 0.0,
sim.max_resistencia);      # configuração período de resistencia é de 0 - 10 dias


                if state = resistencia then
                        myColor := (green);
                        if clock() >= simTime then
                                if random(uniform, 0.0, 10.0) > 2.0 then      #
                                        state := reconhecimento;
                                        simTime := clock() +
random(uniform, 0.0, sim.max_reconhecimento);
                                else
                                        state := recuperacao;

                if state = reconhecimento then
                        myColor := (red);
                        if clock() >= simTime then
                                if (random(uniform, 0.0 , 10.0) > 9.2) then
                                        state := morto;
                                else
                                        state := recuperacao;

                if state = recuperacao then
```

```
                    myColor := (white);

                if state = morto then
                    if flag = 0 then
                        dias_d := clock() + 3.0;
                        flag := 1;
                    myColor := (black);
                    wait 1.0;
                    if dias_d = clock() then
                        state := recuperacao;
                        myColor:= (white);


                if id =1 then
                    verificaComprometimento(num_users , num_micros);

# ######################################################
user(id: int, initial_state: state_type, num_users:int, num_micros:int):
actor type is
        state:: state_type := initial_state;
        simTime:: number := 0.0; # dias
        dias_d:: number := 0.0;    # dias comprometido
        flag:: int := 0;
        distMax:: number := 62.0;           # maxima distância de vizinhas
        infNeighbors:: int := 0;     # variável vizinhos
        xHome:: number := ?;
        yHome:: number := ?;

        if initial_state = normal then
                xHome:= random(uniform, 0.0, 800.0);
                yHome := random(uniform, 0.0, 500.0);
        else
                xHome:= random(uniform, 0.0, 600.0);
                yHome := random(uniform, 0.0, 200.0);
                simTime      :=      clock()      +      random(uniform,      0.0,
sim.max_reconhecimento);

        xloc:: number := 1.1*xHome;
        yloc:: number := 1.1*yHome;
        myColor:: pattern := (green);

        if initial_state = normal then
                myColor := (yellow);
        if initial_state = reconhecimento then
                myColor := (red);

        depict(sim.vw, var offset_by(paint(circle(5.0, 5.0, 5.0), var myColor),
var xloc, var yloc));
```

```
wait 1.0;

for every true do
        wait 1.0;
        changePositionUser(self);
        if state = normal then
                myColor := (yellow);
                infNeighbors := 0;
                for p: each sim.usuarios do
                        if p.state = reconhecimento then
                                if get_distance(xloc, yloc, p.xloc,
p.yloc) < distMax then
                                        infNeighbors :=
infNeighbors + 1;

                if infNeighbors > random(uniform, 0.0, 6.0) then
                        state := resistencia;
                        simTime := clock() + random(uniform, 0.0,
sim.max_resistencia);

        if state = resistencia then
                myColor := (green);
                if clock() >= simTime then
                        if random(uniform, 0.0, 10.0) > 2.0 then
                                state := reconhecimento;
                                simTime := clock() +
random(uniform, 0.0, sim.max_reconhecimento);
                        else
                                state := recuperacao;

        if state = reconhecimento then
                myColor := (red);
                if clock() >= simTime then
                        if (random(uniform, 0.0 , 10.0) > 9.2) then
                                state := morto;
                        else
                                state := recuperacao;

        if state = recuperacao then
                myColor := (white);

        if state =  morto then
                if flag = 0 then
                        dias_d := clock() + 3.0;
                        flag := 1;
                myColor := (black);
                wait 1.0;
                if dias_d = clock() then
                        state := recuperacao;
```

```
                    myColor:= (white);


# ##################################################

verificaComprometimento(num_usu: number, num_pc:number): action is
        num_user_compr:: int := 0.0;
        num_user_inc:: int := 0.0;
        num_user_inf:: int := 0.0;
        num_user_imune:: int := 0.0;

        num_pc_compr:: int := 0.0;
        num_pc_inc:: int := 0.0;
        num_pc_inf:: int := 0.0;
        num_pc_imune:: int := 0.0;

        percent_compro_user:: number := 0.0;
        percent_compro_pc:: number := 0.0;
        tot_compro:: number := 0.0;


        for p: each sim.usuarios do
            if p.state = morto then
                num_user_compr := num_user_compr + 1.0;
            if p.state = resistencia then
                num_user_inc := num_user_inc + 1.0;
            if p.state = reconhecimento then
                num_user_inf := num_user_inf + 1.0;
            if p.state = recuperacao then
                num_user_imune := num_user_imune + 1.0;

        for m: each sim.micros do
            if m.state = morto then
                num_pc_compr := num_pc_compr + 1.0;
            if m.state = resistencia then
                num_pc_inc := num_pc_inc + 1.0;
            if m.state = reconhecimento then
                num_pc_inf := num_pc_inf + 1.0;
            if m.state = recuperacao then
                num_pc_imune := num_pc_imune + 1.0;

        percent_compro_user := (100 * num_user_compr)/num_usu;
        percent_compro_pc := (100 * num_pc_compr)/num_pc;
        tot_compro := percent_compro_user + percent_compro_pc;

        output ("Dias: ", clock()-1,"\cr");
        output("Comprometimento da rede por usuarios: ",
percent_compro_user,
```

```
"%\cr");
                output("Comprometimento da rede por micros: ",
percent_compro_pc,
"%\cr\cr");

                output("USUARIOS","\cr");
                output("                - Comprometidos Parcial: ",
num_user_compr,"\cr");
                output("     - Reconhecimento: ",num_user_inf,"\cr");
                output("     - Resistencia: ",num_user_inc,"\cr");
                output("     - Recuperacao: ",num_user_imune,"\cr");

                output("MICROS","\cr");
                output("     - Comprometidos Parcial: ",num_pc_compr,"\cr");
                output("     - Reconhecimento: ", num_pc_inf,"\cr");
                output("     - Resistencia: ", num_pc_inc,   "\cr"      );
                output("                - Recuperacao:    ",num_pc_imune,
"\cr\cr\cr\cr\cr\cr");

                if tot_compro > 10 then
                        output("Rede comprometida por usuarios e micros");
                if num_user_compr >= (num_usu/10) then
                        output("Rede comprometida por usuarios");
                if num_pc_compr >= (num_pc/10) then
                        output("Rede comprometida por micros");


changePositionUser(u: user): action is
        u.xloc := u.xHome+random(uniform, ~20.0, 20.0);
        u.yloc := u.yHome+random(uniform, ~20.0, 20.0);

changePosition(p: micro): action is
        p.xloc := p.xHome+random(uniform, ~20.0, 20.0);
        p.yloc := p.yHome+random(uniform, ~20.0, 20.0);



simulate(): action is
        epidemic:: epi := new epi;
        epidemic.vw := new view(epidemic, "*** Simulacao - Comprometimento
de Rede ***", lightgray, nil);
        null make_window(epidemic.vw, 1);
        num_micros:: number := 563;
        num_reconhecimento:: number := 1.0;
        num_users:: number := 1200;
        num_user_reconhecimento:: number := 1.0;


        for i: each 1..num_micros do
```

```
                push(epidemic.micros, new(epidemic, micro(i,
normal,num_users,num_micros)));
        for i: each 1..num_reconhecimento do
                push(epidemic.micros, new(epidemic, micro(i,
reconhecimento,num_users,num_micros)));

        for i: each 1..num_users do
                push(epidemic.usuarios, new(epidemic, user(i,
normal,num_users,num_micros)));
        for i: each 1..num_user_reconhecimento do
                push(epidemic.usuarios, new(epidemic, user(i,
reconhecimento,num_users,num_micros)));
        wait epidemic;

simulate();
```

Anexo 2
#
#
##
Simulação
Dia a Dia

##

```
include "::libraries:physical.easel";


state_type: type is enum(naive, resistencia, reconhecimento, morto, recuperação);
epi: simulation type is
        vw:: view := ?;
        micros:: list := new list any;
        max_resistencia:: number := 10.0;
        max_reconhecimento:: number := 4.0;
        num_pc_imune:: int := 0.0;
        num_user_imune:: int := 0.0;
        usuarios:: list := new list any;



micro(id: int, initial_state: state_type, num_users:int, num_micros:int):
actor type is
        state:: state_type := initial_state;
        simTime:: number := 0.0; # dias
        flag:: int := 0;
        dias_d:: number := 0.0;    # dias comprometido
        distMax:: number := 62.0;          # maxima distancia entre vizinhos
        infNeighbors:: int := 0;     # variavel vizinhos
        xHome:: number := ?;
        yHome:: number := ?;


        if initial_state = naive then
                xHome:= random(uniform, 10.0, 800.0);
                yHome := random(uniform, 10.0, 500.0);
        else
                xHome:= random(uniform, 10.0, 600.0);
                yHome := random(uniform, 10.0, 200.0);
                simTime := clock() + random(uniform, 0.0, sim.max_reconhecimento); # configuração do período de reconhecimento de 0 – 4 dias.

        xloc:: number := 1.1*xHome;
```

```
        yloc:: number := 1.1*yHome;
        myColor:: pattern := (green);

        if initial_state = naive then
                myColor := (yellow);
        if initial_state = reconhecimento then
                myColor := (red);

        depict(sim.vw, var offset_by(paint(circle(0.0, 0.0, 10.0), var myColor),
var xloc, var yloc));

        wait 1.0;

        for every true do
                wait 1.0;
                changePosition(self);
                if state = naive then
                        myColor := (yellow);
                        infNeighbors := 0;
                        for p: each sim.micros do
                                if p.state = reconhecimento then
                                        if get_distance(xloc, yloc, p.xloc,
p.yloc) < distMax then
                                                infNeighbors :=
infNeighbors + 1;

                        if infNeighbors > random(uniform, 0.0, 6.0) then
                                state := resistencia;
                                simTime := clock() + random(uniform, 0.0,
sim.max_resistencia);     # Configuração do periodo de resistencia de 0 – 10
dias.

                        if random(uniform, 0.0, 1000)  < 2 then
                                        state := resistencia;
                                        simTime := clock() +
random(uniform, 0.0, sim.max_resistencia);

                if state = resistencia then
                        myColor := (green);
                        if clock() >= simTime then
                                if random(uniform, 0.0, 10.0) > 2.0 then
                                        state := reconhecimento;
                                        simTime := clock() +
random(uniform, 0.0, sim.max_reconhecimento);
                                else
                                        state := naive;
                                        sim.num_pc_imune :=
sim.num_pc_imune + 1;
```

```
                if state = reconhecimento then
                        myColor := (red);
                        if clock() >= simTime then
                                if (random(uniform, 0.0 , 10.0) > 9.2) then
                                        state := morto;
                                else
                                        state := naive;
                                        myColor := (yellow);
                                        sim.num_pc_imune                        :=
sim.num_pc_imune + 1;
                if state =  morto then
                        if flag = 0 then
                                dias_d := clock() + 3.0;
                                flag := 1;
                        myColor := (black);
                        wait 1.0;
                        if dias_d = clock() then
                                flag := 0;
                                state := naive;
                                myColor := (yellow);
                                sim.num_pc_imune := sim.num_pc_imune +
1;

                if id =1 then
                        verificaComprometimento(num_users , num_micros);

# ######################################################
user(id: int, initial_state: state_type, num_users:int, num_micros:int):
actor type is
        state:: state_type := initial_state;
        simTime:: number := 0.0; # dias
        dias_d:: number := 0.0;    # dias comprometido
        flag:: int := 0;
        distMax:: number := 62.0;            # maxima distancia entre vizinhos
        infNeighbors:: int := 0;     # variavel de vizinhos
        xHome:: number := ?;
        yHome:: number := ?;

        if initial_state = naive then
                xHome:= random(uniform, 0.0, 800.0);
                yHome := random(uniform, 0.0, 500.0);
        else
                xHome:= random(uniform, 0.0, 600.0);
                yHome := random(uniform, 0.0, 200.0);
                simTime      :=      clock()      +      random(uniform,      0.0,
sim.max_reconhecimento);

        xloc:: number := 1.1*xHome;
        yloc:: number := 1.1*yHome;
```

```
        myColor:: pattern := (green);

        if initial_state = naive then
                myColor := (yellow);
        if initial_state = reconhecimento then
                myColor := (red);

        depict(sim.vw, var offset_by(paint(circle(5.0, 5.0, 5.0), var myColor),
var xloc, var yloc));

        wait 1.0;

        for every true do
                wait 1.0;
                changePositionUser(self);
                if state = naive then
                        myColor := (yellow);
                        infNeighbors := 0;
                        for p: each sim.usuarios do
                                if p.state = reconhecimento then
                                        if  get_distance(xloc,  yloc,  p.xloc,
p.yloc) < distMax then
                                                infNeighbors          :=
infNeighbors + 1;

                        if infNeighbors > random(uniform, 0.0, 10.0) then
                                state := resistencia;
                                simTime  :=  clock()  +  random(uniform,  0.0,
sim.max_resistencia);

                        if random(uniform, 0.0, 1000) < 2 then
                                state := resistencia;
                                simTime  :=  clock()  +  random(uniform,  0.0,
sim.max_resistencia);

                if state = resistencia then
                        myColor := (green);
                        if clock() >= simTime then
                                if random(uniform, 0.0, 10.0) > 2.0 then
                                        state := reconhecimento;
                                        simTime       :=       clock()       +
random(uniform, 0.0, sim.max_reconhecimento);
                                else
                                        state := naive;
                                        myColor := (yellow);
                                        sim.num_user_imune          :=
sim.num_user_imune + 1;

                if state = reconhecimento then
```

```
                        myColor := (red);
                        if clock() >= simTime then
                                if (random(uniform, 0.0 , 10.0) > 9.2) then
                                        state := morto;
                                else
                                        state := naive;
                                        myColor := (yellow);
                                        sim.num_user_imune              :=
sim.num_user_imune + 1;

                if state =  morto then
                        if flag = 0 then
                                dias_d := clock() + 3.0;
                                flag := 1;
                        myColor := (black);
                        wait 1.0;
                        if dias_d = clock() then
                                flag := 0;
                                state := naive;
                                myColor:= (yellow);
                                sim.num_user_imune                      :=
sim.num_user_imune + 1;


# ######################################################

verificaComprometimento(num_usu: number, num_pc:number): action is
                num_user_compr:: int := 0.0;
                num_user_inc:: int := 0.0;
                num_user_inf:: int := 0.0;

                num_pc_compr:: int := 0.0;
                num_pc_inc:: int := 0.0;
                num_pc_inf:: int := 0.0;

                percent_compro_user:: number := 0.0;
                percent_compro_pc:: number := 0.0;
                tot_compro:: number := 0.0;


                for p: each sim.usuarios do
                        if p.state = morto then
                                num_user_compr := num_user_compr + 1.0;
                        if p.state = resistencia then
                                num_user_inc := num_user_inc + 1.0;
                        if p.state = reconhecimento then
                                num_user_inf := num_user_inf + 1.0;

                for m: each sim.micros do
```

```
                        if m.state = morto then
                                num_pc_compr := num_pc_compr + 1.0;
                        if m.state = resistencia then
                                num_pc_inc := num_pc_inc + 1.0;
                        if m.state = reconhecimento then
                                num_pc_inf := num_pc_inf + 1.0;

                percent_compro_user := (100 * num_user_compr)/num_usu;
                percent_compro_pc := (100 * num_pc_compr)/num_pc;
                tot_compro := percent_compro_user + percent_compro_pc;

                output ("Dias: ", clock()-1,"\cr");
                output("Comprometimento da rede por usuarios: ",
percent_compro_user,
"%\cr");
                output("Comprometimento da rede por micros: ",
percent_compro_pc,
"%\cr\cr");

                output("USUARIOS","\cr");
                output("                - Comprometidos Parcial: ",
num_user_compr,"\cr");
                output("     - Reconhecimento: ",num_user_inf,"\cr");
                output("     - Resistencia: ",num_user_inc,"\cr");
                output("     - Recuperacao: ",sim.num_user_imune,"\cr");

                output("MICROS","\cr");
                output("     - Comprometidos Parcial: ",num_pc_compr,"\cr");
                output("     - Reconhecimento: ", num_pc_inf,"\cr");
                output("     - Resistencia: ", num_pc_inc,    "\cr"       );
                output("                - Recuperacao:  ",sim.num_pc_imune,
"\cr\cr\cr\cr\cr\cr");

                if tot_compro > 10 then
                        output("Rede comprometida por usuarios e micros");
                if num_user_compr >= (num_usu/10) then
                        output("Rede comprometida por usuarios");
                if num_pc_compr >= (num_pc/10) then
                        output("Rede comprometida por micros");


changePositionUser(u: user): action is
        u.xloc := u.xHome+random(uniform, ~20.0, 20.0);
        u.yloc := u.yHome+random(uniform, ~20.0, 20.0);

changePosition(p: micro): action is
        p.xloc := p.xHome+random(uniform, ~20.0, 20.0);
        p.yloc := p.yHome+random(uniform, ~20.0, 20.0);
```

```
simulate(): action is
        epidemic:: epi := new epi;
        epidemic.vw := new view(epidemic, "*** Simulacao - Comprometimento
de
Rede ***", lightgray, nil);
        null make_window(epidemic.vw, 1);
        num_micros:: number := 563;
        num_reconhecimento:: number := 1.0;
        num_users:: number := 1200;
        num_user_reconhecimento:: number := 1.0;


        for i: each 1..num_micros do
                push(epidemic.micros,          new(epidemic,          micro(i,
naive,num_users,num_micros)));
        for i: each 1..num_reconhecimento do
                push(epidemic.micros, new(epidemic, micro(i,
reconhecimento,num_users,num_micros)));

        for i: each 1..num_users do
                push(epidemic.usuarios, new(epidemic, user(i,
naive,num_users,num_micros)));
        for i: each 1..num_user_reconhecimento do
                push(epidemic.usuarios, new(epidemic, user(i,
reconhecimento,num_users,num_micros)));
        wait epidemic;

simulate();
```