DÉBORA WINTER FERNANDES

SEGURANÇA NA INTERNET ?

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre em Engenharia de Produção da Universidade Federal de Santa Catarina, sob a orientação do Prof. Dr. Edgar Augusto Lanzer.

FLORIANÓPOLIS

2000

Segurança na Internet?

Dissertação submetida ao Colegiado do Curso de Mestrado em Engenharia de Produção do Centro de Engenharia de Produção em cumprimento parcial para a obtenção do título de Mestre, Curso de Pós-graduação em Engenharia de Produção da Universidade Federal de Santa Catarina e aprovada pela Comissão Examinadora formada pelos professores:

Prof. Dr. Edgar Augusto Lanzer
EPS/UFSC – Orientador

Prof. Dr. Carlos Becker Westphall
INE/UFSC

Prof. Dr. Bernardo Gonçalves Riso
INE/UFSC

Centro Tecnológico
Programa de Pós-Graduação em Eng. de Produção
Prof. Ricardo Miranda Barcia, Ph. D.
COORDENADOR

Florianópolis, 31 de março de 2000.

Ao meu amigo, colega de árduas jornadas de trabalho e também agora, meu companheiro....

À pequena Emily a quem já tive o prazer de dedicar meu primeiro trabalho

AGRADECIMENTOS

São tantos a agradecer que tenho receio de não ser justa com todos, por isso faço uma agradecimento especial:

aos meus pais, companheiros fiéis de vida, sempre dispostos a enfrentar juntos os desafios dos seus filhos e é deles que continuo aprendendo a cada dia à força de lutar pelos ideais;

à equipe do CPD do Porto de São Francisco do Sul, Rodrigo e Alexandre, pela paciência e apoio na parte final deste trabalho;

à Administração do Porto de São Francisco do Sul (APSFS) que além de servir como apoio de infraestrutura computacional, foi um grande aprendizado pessoal e profissional;

ao meu orientador que mesmo distante, tenho certeza, sempre acreditou no concepção do trabalho;

aos membros da banca pelas excelentes sugestões, que enriqueceram o texto e colaboraram para a sua conclusão final;

e aos amigos que de uma forma ou de outra souberam entender a importância do trabalho e colaborar para a sua conclusão.

A todos, agradeço.

SUMÁRIO

LISTA DE QUADROS E FIGURAS

LISTA DE ABREVIATURAS

API	Application Program Interface
ARP	Adress Resolution Protocol
ASN.1	Abstract Sintaxe Notation (1)
ATM	Asynchronous Transfer Mode
BGP	Border *Gateway* Protocol
BSD	Berkeley *Software* Distribution
CCITT	Comité Consultatif International Télégraphique et Téléphonique
CGI	Common *Gateway* Interface
DARPA	Defense Advanced Research Projects Agency
DDL	Linguagem de Definição de Dados
DES	Data Encryption Standard
DHCP	Dynamic *Host* Configuration Protocol
DML	Linguagem de Manipulação de Dados
DNS	Domain Name System
DoD	Departament of Defense
DSA	Digital Signature Algorithm
DSS	Digital Signature Standard
FDDI	Fiber Distributed Data Interface
FTP	File Transfer Protocol
HLSP	High Level Security Policy.
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IANA	Internet Assigned Numbers Authority
ICMP	Internet Control Message Protocol
ICQ	I Seek You
IDEA	International Data Encryption Algorithm

IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IIS	Internet Information Server
IMAP	Interactive Mail Access Protocol
IP	Internet Protocol
IPSec	IP Security Protocol
IPv6	IP versão 6
IPX	Internetwork Packet Exchange
IRC	Internet Relay Chat
ISDN	Integrated Services Digital Network
ISO	International Organization for Standardization
ISOC	Internet Society
ISP	Provedor de Serviços da Internet
KDBM	Kerberos Database Management System
KDM	Kerberos Database Manager
KDS	Key Distribution Services
LAN	Local Area Network
MAC	Control Access Media
MAN	Metropolitan Area Network
MD2/MD4/MD5	Message Digest Algorithm 2/4/5 (respetivamente)
MIB	Management Information Base
MIME	Multipurpose Internet Media Types
MOSPF	Multicast Open Shortest Path First
NAT	Network Address Translation
NetBios	Network Basic Input/Output System
NFS	Network File System
ODBC	Open Database Connectivity
OSI	Open System Interconnection
OSPF	Open Shortest Path First
PCT	Private Communications Technology
PGP	Pretty Good Protection

PPP	Point-to-point Protocol
PPTP	Point-to point Tunneling Protocol
Proxy ARP	Proxy Address Resolution Protocol
RACF	Resource Access Control Facility.
RADIUS	Remote Authentication Dial-In User Server
RARP	Reverse Address Resolution Protocol
RFC	Request for Comments
RIP	*Routing* Information Protocol
RSA	Rivest, Shamir and Adleman
RSCS	Remote Spooling Communications Subsystem
SHA	Secure Hash Algorithm
SHS	Secure Hash Standard):
SHTTP	Secure HTTP
SMTP	Simple Mail Transfer Protocol
SNA	System Network Architecture
SNMP	Simple Network Management Protocol
SQL	Structured Query Language
SSL	Secure *Socket* Layer
TACACS	Terminal Access Controler Access Control System
TCP	Transmission Control Protocol
TCSP	Technical Committee on Security and Privacy
TGS	Ticket Granting Service
TTL	Time To Live
UDP	User Datagram Protocol
URI	Universal Resource Identifier
URL	Uniform Resource Locator
VPN	Network Private Virtual
WAN	Wide Area Network
Web	abreviatura de World Wide Web
WWW	World Wide Web

RESUMO

No início dos anos 70, foi criada a Internet pelo Departamento de Defesa Americano (DoD). Foi desenvolvido uma arquitetura e softwares (protocolos) que permitiam a comunicação entre os computadores. O objetivo desta estrutura inicial era facilitar ao máximo o tráfego das informações entre os órgãos envolvidos no projeto. Foi um sucesso e rapidamente novos computadores se conectavam a esta estrutura de "rede". Conjuntamente com a Internet surgiram novos conceitos e entre eles a "segurança das informações" que transitam livremente pela rede. Matéria esta abordada desde 1970, em algumas universidades americanas, que com uma visão futurista já percebiam a importância da confiabilidade e integridade dos dados. Atualmente este é um tema polêmico, muito se houve falar e muitos trabalhos são realizados em todas as partes do mundo. Mas, o que existe de real, de possível, sem os exageros da mídia e sem a grandiosidade pessoal? Existem muitos mecanismos que tratam de forma eficiente e confiável o tema "segurança". E é sobre eles que este trabalho se refere. Mostrando inicialmente a arquitetura básica de redes, os protocolos mais utilizados por esta estrutura, técnicas de firewalls para controle de fluxo de acesso na rede, os protocolos "seguros", a autenticação e certificação e finalmente das VPNs. "Segurança na Internet?" procura ser uma fonte de consulta para administradores e gerentes de redes que, preocupados com as informações coorporativas, gerenciam os acessos, as transações e as autenticações de suas redes locaise WANs, usuários e os acessos à Internet.

ABSTRACT

In the early of 1970s, the U.S. Departament of Defense (DOD) created the Internet. It was developed an architecture and softwares that allowed the communication among computers. The objective of this initial structure was to facilitate the traffic of the information among the organs involved in the project.
It was a success and quickly new computers were connected to this net structure. Jointly with the Internet new concepts appeared and among them the security of the information that go through freely for the net. So, it was approached since 1970 in the americans universities that with a futurist vision, noticed the importance of the reliability and integrity of the data. Actually, this is a polemic subject, and many peoples talk about that around the world. But, what is real or possible, without the exaggerations of the communication medium and without the personal greatness? Today, there are many efficient and reliable mechanisms about this subject,. and this dissertation is about them. We start showing the network basic architeture, the most used protocols by this structure, firewall techniques for access control in the net, the secure protocols, the authentication and certification techniques and finally the VPN's. "Security in the Internet" is a consultation source for net administrators and managers, that worried with the corporate informations, manage the accesses, the transactions, and the authentications of your nets.

INTRODUÇÃO

Um antigo truísmo sobre segurança é que o custo de se proteger contra uma ameaça deve ser menor que o custo de se recuperar se a ameaça o atingir. Devemos lembrar que o custo nesse contexto deve incluir perdas expressas em moeda corrente, reputação, confiabilidade e outras medidas menos óbvias.

"The site security handbook", RFC 2196

Ano 2000, chegamos ao novo milênio, passamos pelo século mais importante na história tecnológica do mundo. Um século de descobrimentos, revelações fantásticas, o impossível era rapidamente possível.

No começo do século a comunicação a distância era feita principalmente através da invenção de Grambell, por telégrafo (Samuel F. B. Morse, 1838) e a cavalo(correio), sem levar em consideração o tempo até chegar ao seu destinatário.

A educação era restrita a poucos que tinham condições de pagar escolas que dispunham da informação (livros, enciclopédias) para o bom ensino de seus alunos. Mesmo mais recentemente, no meio do século XX, as bibliotecas eram o centro de informações, para qualquer trabalho acadêmico, pois o preço dos livros era inacessível à grande maioria de alunos.

E hoje, qual é nossa realidade?

Podemos nos comunicar rapidamente utilizando telefone convencional ou celular, telegramas ou se desejarmos uma comunicação com baixo custo a qualquer um dos continentes, usamos a Internet.

Um aluno de qualquer ano escolar, senta-se em frente de seu computador, faz sua consulta em um site de busca e encontra informações a respeito do assunto que precisa para seu trabalho.

Listas de discussão, BBS, FAQ, IRC, ICQ, e-mail... são tantos os meios e modos de comunicação na Internet que as vezes é difícil se encontrar neste ciberespaço. Desde formas de comunicação mais informativas e de conhecimento técnico até bate-papos informais.

Com um grande número de "internautas navegando", a Internet despertou o interesse de empresas que desejavam oferecer seus produtos e serviços, surge o marketing virtual.

Neste universo sem fronteiras e com pouca legislação surgiu a preocupação com as informações que navegavam de um nó a outro. No início da década de 70 algumas universidades americanas já possuíam em seus currículos dos cursos da área tecnológica, matérias sobre segurança de dados. É claro que de uma forma bem diferente da segurança que necessitamos na Internet de hoje, pois nesta década de 70 os sistemas eram basicamente de grande porte, sistemas distribuídos.

Entenda-se por sistema distribuído: "...em um sistema distribuído a existência de computadores autônomos é transparente para o usuário" (TANENBAUM, p. 2, 1994). Em outras palavras, o usuário não percebe a existência de múltiplos processadores, tudo parece como um monoprocessador virtual. Em uma rede de computadores: "...um usuário precisa abrir *explicitamente* uma sessão em uma máquina, submeter *explicitamente* uma tarefa remota, mover *explicitamente* os arquivos de lá para cá e, de maneira geral, lidar pessoalmente com a gestão da rede". (Ibid, p.3)

E é exatamente nesta "segurança" das informações que nos últimos anos as empresas tem se voltado, pois a confidencialidade e integridade dos dados é vital para o funcionamento de muitas empresas que utilizam a Internet como caminho entre uma fábrica e outra. E para nós que a utilizamos para troca de mensagens, compras, consulta a *home banking* etc, tornou-se vital manter nossos dados seguros.

Nesta era digital surgiu um novo papel, o *Hacker*, que pela definição seria uma pessoa muito especializada em determinada área, mas que virou o bandido do ciberespaço. Teoricamente uma pessoa que tem "poderes" para invadir um computador e realizar suas "maldades" ou se apropriar de informações que não lhe são devidas.

E então, realmente estamos seguros? Como saber se estamos seguros? Como fazer para estarmos seguros? O que é verdade, o que é especulação?

Foi pensando em como encontrar as respostas dessas e outras perguntas que este trabalho foi elaborado. Para fornecer material de pesquisa para administradores e gerentes que estão preocupados com a implementação de segurança voltada as "portas abertas" da Internet.

A INTERNET

O início da Internet foi na década de 70, quando o departamento de defesa americano (DoD) interligou alguns de seus computadores a outras redes de computadores dos campus de pesquisas de centros de tecnologia e instituições de ensino superior, para trocar informações, criando assim uma Inter-rede. (STARLIN, 1999)

A Internet marcou profundamente a mutação intelectual e política das últimas décadas. No Brasil do final dos anos 80, por exemplo, vivia-se a euforia pela emergência das redes de computadores e pela esperança de que a abertura de mercado, popularizaria a tecnologia em pequenas empresas e à nível pessoal. A década seguinte, foi a vitória da sociedade, pela concretização da abertura de mercado e pelo decréscimo do preço dos equipamentos necessários a difusão da tecnologia de rede. Nesta mesma década, o Brasil fazia parte da grande rede, com várias empresas e universidades, enviando e recebendo informações. Com o crescente aumento, desenvolveu-se várias linguagens, aplicativos e principalmente protocolos que servem de base deste fluxo.

O TCP/IP que foi desenvolvido por um conjunto de pesquisadores é na realidade um conjunto de serviços que padroniza um sistema de comunicação combinando aplicações e protocolos e foi o responsável pelo início da Internet. Na década de 80 com a entrada de muitas empresas que passaram a vender e comprar

serviços e produtos, foi desenvolvido em 1986 um sistema de interfaceamento padrão de saída de dados chamado HTML, que conhecemos hoje como Web ou WWW, que facilitou o uso da Internet pelos usuários.

Os números da Internet no Brasil são rapidamente crescentes. O Comitê Gestor da Internet no Brasil, calculou, em abril de 1996, o número de usuários no país em 300 mil. Dados da Embratel informavam que existiam em 1997 cerça de 800 mil pessoas conectadas à rede. Em 1998, nova pesquisa do IBOPE, ofereceu números mais precisos sobre a rede brasileira: 1,8milhão de usuários nas principais regiões metropolitanas (São Paulo, Rio de Janeiro, Belo Horizonte, Recife, Fortaleza, Salvador e Distrito Federal). (OLIVO, 1998)

Como se verifica na informação acima, o número de consumidores não só no Brasil, mas em todos os continentes é muito grande e por esta razão é que a Internet é o caminho mais curto e rápido entre o usuário final e sua busca, seja ela por informação, serviços ou produtos. E a cada acesso o usuário entra num mundo desconhecido de *links* onde muitas vezes é necessário a identificação através da digitação de números de senha, da carteira de identidade, CPF e cartão de crédito. Estas informações são altamente sigilosas e não devem ser interceptadas ou lidas por pessoal não autorizado.

E este, é entre muitos dos itens a serem avaliados num *site* ou numa rede, o que merece destaque e um dos que mais preocupam qualquer desenvolvedor de aplicativos Web ou administradores de redes: a segurança das informações que nela navegam.

SEGURANÇA NA INTERNET

Segurança é definida como uma "corrente" composta por diversos elos, que seriam os pontos vulneráveis merecedores de atenção. Neste conceito de corrente, existe a preocupação em manter o mesmo nível de segurança para cada elo, pois a força desta corrente é medida pelo seu elo mais fraco (vulnerável).

É um tópico especialmente importante para projetistas de redes corporativas, devido ao número crescente de conexões da Internet e extranets, ao aumento do comércio eletrônico na Internet e a um número maior de pessoas que trabalham em casa e usuários em trânsito que tem acesso a redes corporativas a partir de *sites* remotos. (OPPENHEIMER, 1999)

Ameaças como os atuais "vírus" ou "cavalos de Tróia" potencializam este tipo de vulnerabilidade, permitindo ao invasor adquirir domínio total da estação e capturar toda e qualquer informação armazenada, inclusive a senha de acesso aos servidores da mesma rede. A privacidade e proteção na utilização do correio eletrônico, requer sistemas específicos e soluções de identidade digital que permitem a troca de mensagens criptografadas – garantido o sigilo e integridade no trajeto e na armazenagem.

A segurança é uma matéria complexa, são muitos parâmetros e variáveis envolvidas no processo. De nada adianta, toda uma estrutura de defesa no acesso da rede à internet, se os funcionários (rede interna) fazem acesso direto via *modem* a seus provedores e escancarem uma porta pela qual pode entrar ou sair qualquer tipo de informação ou código hostil.

Como garantir a segurança no ciberespaço? *Firewalls*, criptografia, autenticação, certificação, protocolos seguros, VPN, *firewall* distribuído, são tecnologias disponibilizadas continuamente, mas qual utilizar para garantir a segurança necessária? Após responder estas perguntas e definir o projeto da rede e do *site*, podemos definir as tecnologias necessárias à segurança em função de seu custo e valor do patrimônio(informação) que será protegido. Como pessoas e tecnologias sozinhas não são uma união das mais estáveis, processos precisam ser formalizados, uma visão estratégica da segurança precisa ser endossada pela alta administração e regras claras precisam ser criadas. Precisamos ainda definir a Política de Segurança de Informações baseada em regras estratégicas, táticas e operacionais, apoiada no desenvolvimento de cultura de segurança entre usuários, uso de ferramentas tecnológicas e monitoração constante.

Os termos segurança e contingência são tão antigos na área da tecnologia quanto os mainframes que funcionavam a válvulas. Entre desconsiderar o perigo e levá-lo a paranóia, devemos buscar uma postura madura em relação ao nível de segurança necessário aos sistemas definindo concretamente as necessidades reais dentro da empresa ou num *site*.

Segurança é projeto, é processo dinâmico e é experiência vivenciada ao mesmo tempo que tem alguém criando dificuldades ou implementando proteção, outros tantos estão estudando como burlar ou quebrar o modelo. Existe a necessidade de alcançar o estratégico da organização e conseguir emanar ações preventivas e pró-ativas durante a elaboração do projeto, pensando globalmente e agindo localmente. Definir quais informações serão colocadas e recebidas na Internet e que impactos trariam se perdessem confidencialidade, disponibilidade e integridade.

Utilizando planos de segurança (como evitar problemas – descobrir os pontos vulneráveis, avaliar os riscos, tomar as providências adequadas e investir o necessário para ter uma segurança homogênea e suficiente) e plano contingência (o que fazer em caso de problemas) se reduz em grande parte os problemas de segurança. Não existe "sistema totalmente seguro", mas existem formas de reduzir e dificultar processos indevidos e precauções em casos de invasões, perdas de dados, catástrofes ou problemas ambientais.

Num projeto de segurança é vital desenvolver a visão de como os relacionamentos e as trocas de informações ocorrem, identificando quem são os fornecedores, clientes e parceiros. Conhecidos os processos críticos, é preciso identificar as aplicações e a infra-estrutura tecnológica que os sustentam, estabelecendo os perímetros e os mecanismos de segurança para a proteção das informações do negócio.

Nos casos de contingência, os backup's fazem parte de uma política completa e adequada, contemplando estratégias de rodízio incremental – utilizando-se de diversas unidades de fita e, um esquema de armazenamento de acesso controlado. Mesmo que o servidor seja bem configurado, o e-mail seguro, o lixo

informático fragmentado, e os backup's realizados com exatidão, é necessário ainda uma política de segurança que sirva como bússola, definindo diretrizes, responsabilidades, normas e procedimentos para a melhor utilização do ambiente informatizado.

Normalmente após realizar a instalação de um *firewall*, roteador, instalação dos últimos *patches* dos aplicativos, tem-se a sensação de "segurança". Esta falsa sensação de segurança, de achar que está seguro e não precisar fazer mais nada, é muito comum. Mas o problema, é que para se estar seguro, devemos estar sempre atentos aos acontecimentos. A velocidade do surgimento de novos problemas é muito grande. Se houver problema no sistema, como um novo *bug*, as correções devem ser efetuadas. A dinâmica é muito grande e os administradores de redes e *sites* devem estar sempre atentos a novas e concretas ameaças.

Como este trabalho não é sobre o desenvolvimento de projetos de segurança, se o leitor desejar mais informações sobre desenvolvimento destes projetos (metas, etapas etc) consulte OPPENHEIMER, RFC 2196, PARKER e CARUSO; STEFFEN.

ORGANIZAÇÃO E METODOLOGIA

A exposição deste trabalho foi organizada em cinco capítulos:

O primeiro capítulo apresenta os principais protocolos da Internet, necessários a comunicação nas camadas da arquitetura TCP/IP.

O segundo capítulo sobre segurança no acesso: refere-se a habilidade da empresa em proteger os computadores, memória, discos, impressoras e qualquer outro equipamento de uso não autorizado. Este tipo de controle permite gerenciar o acesso de usuários a arquivos e diretórios, incluir testes de invasão nos servidores Internet, analisar a logística de segurança da aplicação Web e da arquitetura da rede interna.

Na sequência, o terceiro capítulo define a segurança das transações: implementação de criptografia, garantindo a transmissão segura dos dados (protocolos seguros) e da transação eletrônica via Internet (SET).

O quarto capítulo se refere aos meios de autenticação (cliente do sistema): requer que uma pessoa ou programa prove sua identidade, usualmente pelo uso de senhas, centros de certificação, autentificação da identidade dos remetentes e destinatários de mensagens (pacotes).

E finalmente, o quinto capítulo mostra a excelente combinação de protocolos, sistemas de autenticação e criptografia utilizando pelas redes virtuais privadas, as VPNs.

A metodologia de trabalho foi definida visando a dinâmica do assunto, inicialmente utilizou-se literatura base para iniciar a pesquisa como Tanenbaum, Soares; Lemos e Colcher, que proporcionaram ampla base sobre a estrutura de redes e protocolos. Em seguida procurou-se fontes sobre projetos e implementação de segurança em redes corporativas e *sites*, onde além da literatura base foi utilizado documentos de trabalhos recentes e também de organizações que padronizam a comunicação na Internet. Muita literatura foi extraída da Internet, de *sites* como IANA, IEEE, TCSP, IETF, W3C e de universidades. Como é necessário conhecer os "dois lados da moeda" foram utilizados *sites* de pessoas que se autodenominam "*hackers*", para busca de informações a respeito de quebras de segurança, pois acredito que para ser um administrador de redes e desenvolvedor de *sites e-comerce* é necessário também saber como são as ações e pensamentos, de um *hacker*.

PRIMEIRO CAPÍTULO
PROTOCOLOS DA INTERNET

A Internet é uma selva. Você quer ser caçador ou caçado?.

Autor desconhecido

Não podemos iniciar este capítulo ou este trabalho sem antes apresentarmos alguns fundamentos sobre estrutura de redes, o modelo de referência OSI e a arquitetura utilizada na Internet – TCP/IP, que são a base da arquitetura das modernas redes de computadores e onde encontramos a fundamentação para o início deste trabalho.

1.1 REDES DE COMPUTADORES

Durante as primeiras décadas da indústria da computação, as redes eram altamente centralizadas, com acesso restrito a poucos que ficavam dentro de uma sala. A idéia de que em vinte anos computadores pudessem ser altamente especializados e realizar funções poderosas em muito pouco espaço de tempo, era um sonho. O velho modelo de um único computador servindo a todas as necessidades foi sendo substituído pelo modelo com grande número de computadores separados, mas interconectados, que executam a tarefa. (TANENBAUM, 1994)

"Uma rede de computadores é formada por um conjunto de módulos (MPs)[1] capazes de trocar informações e compartilhar recursos, interligados por um sistema de comunicação" (SOARES; LEMOS; COLCHER, p. 10, 1995), conforme Figura 1.

[1] A definição de módulos processadores se refere a qualquer dispositivos capaz de se comunicar através do sistema de comunicação por troca de mensagens. Por ex.: um microcomputador, uma máquina copiadora, um computador de grande porte, um terminal, uma impressora etc.

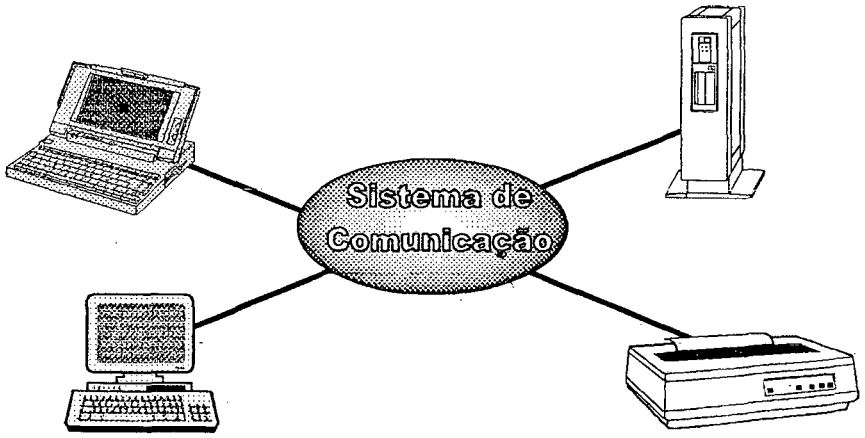

FIGURA 1. Rede de Computadores

O sistema de comunicação é formado por um arranjo topológico interligando os vários módulos processadores através de enlaces físicos e de um conjunto de regras que organizam a comunicação (protocolos). Redes locais (LANs) atuam em distâncias de vão de alguns poucos metros até alguns poucos quilômetros. Permitem a comunicação de dados numa pequena região. Como a definição é vaga, e cada autor define distâncias diferentes, vamos considerar o padrão utilizado por Tanenbaum que as define dentro de um perímetro de 10m até 10km, muito embora as limitações técnicas utilizadas em redes locais não imponham limites a essas distâncias. Possuem taxas de transmissão em torno de 10 a 100Mbps e baixas taxas de erros.

As redes metropolitanas (MANs), que surgiram com o aparecimento do padrão IEEE 802.6, apresentam características semelhantes às das redes locais, sendo que em geral, cobrem distâncias superiores do que as LANs e operando em velocidades maiores.

As redes geograficamente distribuídas (WANs)surgiram da necessidade de se compartilhar recursos dispersos em áreas superiores as MANs. com são redes que utilizam recursos tecnológicos de custo elevado, estas redes são na sua maioria públicas, sendo gerenciadas e mantidas por grandes operadoras (públicas e privadas). São definidas em até 1000km.

Tanenbaum define ainda a interconexão de redes de longa distância, que ultrapassam qualquer distância que seja necessário vencer, pois realizam a conexão de outras redes existentes.

O Modelo de Referência OSI

Para diminuir a complexidade e fornecer uma base comum de desenvolvimento de projetos e padrões de aplicações foi formalizado um modelo de referência para interconexão de sistemas abertos denominado OSI (sistemas que são abertos à comunicação com outros sistemas), que se baseia em uma proposta desenvolvida pela ISO.

O modelo OSI possui sete camadas. Os princípios utilizados para chegar-se às setes camadas foram, de acordo com Tanenbaum:

1. Uma camada deve ser criada onde é necessário um nível de abstração diferente.
2. Cada camada deve desempenhar uma função bem definida.
3. A função de cada camada deve ser definida tendo em vista a definição de protocolos de padrão internacional.
4. As fronteiras entre as camadas devem ser escolhidas de forma a minimizar o fluxo de informações através das interfaces.
5. número de camadas deve ser grande o suficiente para que não seja preciso agrupar funções em uma mesma camada por necessidade, e pequeno o suficiente para que a arquitetura fique manejável.

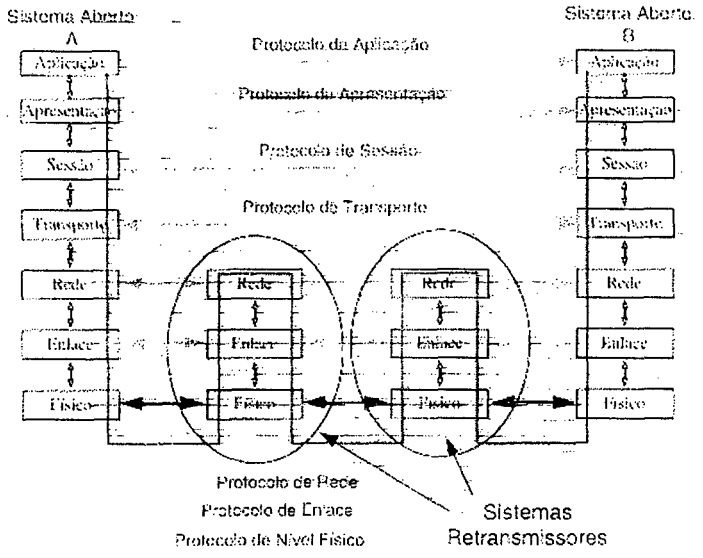

FIGURA 2. Modelo de Referência OSI
Fonte: (SOARES, LEMOS, COLCHER, 1995)

Camada física (1)

Define as especificações elétricas, mecânicas, procedurais e funcionais para ativar, manter e desativar o *link* físico entre sistemas finais. As questões de projeto desta camada fazem parte do domínio de engenheiros elétricos, se referem mais a transmissão pura de *bits* através de um canal de comunicação. (OPPENHEIMER, 1999)

Camada de enlace de dados (2)

O objetivo desta camada é detectar e opcionalmente corrigir erros que ocorram do meio físico. Oferece assim trânsito de dados confiáveis através de um link físico. A técnica utilizada para isto é a fragmentação do dados de entrada em quadros, cada um contendo alguma forma de redundância para detecção de erros. Outro objetivo é utilizar um controle de fluxo que não permita que um transmissor rápido afogue um receptor lento com dados. Esta função também é realizada nas maioria das camadas superiores.

Camada de rede (3)

Se preocupa com o roteamento dos pacotes, como eles vão da origem até o destino. Responsável pelo controle (conectividade) da operação da sub-rede, deve resolver os problemas entre diferentes protocolos, redes heterogêneas, endereçamento, tamanho do pacote etc.

Camada de transporte (4)

No nível de transporte a comunicação é fim a fim, significa que ela é a responsável pela comunicação confiável entre nós finais. Estabelece mecanismos para estabelecer, manter e encerrar circuitos virtuais; detecção e recuperação de falhas de transporte e controle de fluxo de informações. Dois controles importantes são a *multiplexação* (várias conexões de transporte partilhando a mesma conexão de rede) e o *splitting* (uma conexão de transporte ligada a várias conexões de rede).

Camada de sessão (5)

Fornece mecanismos que permitem administrar e encerrar sessões entre aplicativos. Os principais serviço são: gerenciamento de *Token*, controle de diálogo e gerenciamento de atividades.

Camada de apresentação (6)

Os serviços oferecidos por este nível são: transformação de dados, formatação de dados, seleção de sintaxes e estabelecimento e manutenção de conexões de apresentação. Esta camada assegura legibilidade das informações enviadas pela camada de aplicação de um sistema à camada de aplicação de outro.

Camada de aplicação (7)

Fornece serviços e processos de aplicativos. Atua como interface, através do qual o usuário tem acesso aos serviços fornecidos pela aplicação do sistema aberto.

Transmissão de dados no Modelo OSI

A Figura 3 ilustra como acontece a transmissão de dados utilizando o modelo de referência OSI.

O sistema A envia dados (mensagem) ao sistema B.

O processo inicia com a entrega dos dados a serem transmitidos pelo usuário a uma entidade do nível de aplicação no sistema A. Os dados do usuário recebem o denominação de Unidade de Dados do Serviço – SDU, sendo eles neste caso a SDU do nível de aplicação. A entidade da camada de aplicação junta aos dados do usuário um cabeçalho denominado Informação de Controle de Protocolo –

PCI. O objeto resultante dessa junção é chamado Unidade de Dados do Protocolo – PDU. A PDU é a unidade de informação trocada pelas entidades pares, ao executar o protocolo de uma camada, para fornecer o serviço que cabe à camada em questão. A PDU do nível de aplicação (cabeçalho + dados) é passada para o nível de apresentação.

A entidade do nível de apresentação trata a unidade que recebe da mesma forma que o nível de aplicação trata os dados do usuário (a PDU do nível de aplicação é uma SDU no nível de apresentação), e acrescenta seu cabeçalho compondo assim, a PDU do nível de apresentação. Este processo continua até o nível de enlace, que geralmente acrescenta um cabeçalho e um fecho que contém uma *Frame Check Sequence* – FCS para detecção de erros. A PDU do nível de enlace, que é denominada quadro (*frame*), é transmitida pelo nível físico através do meio de transmissão, depois de agregar ao quadro seu cabeçalho e seu fecho. Quando o quadro é recebido pelo destinatário, o processo inverso ocorre. A medida que a unidade de dados vai sendo passada para as camadas superiores, cada camada retira o cabeçalho e o fecho que foi acrescentado por sua entidade par na origem, executa as operações do protocolo de acordo com a informação contida no cabeçalho, e passa a unidade de dados para à camada superior. O processo se encerra com o usuário no sistema remoto B recebendo os dados enviados pelo usuário no sistema A. (SOARES, LEMOS, COLCHER, 1995)

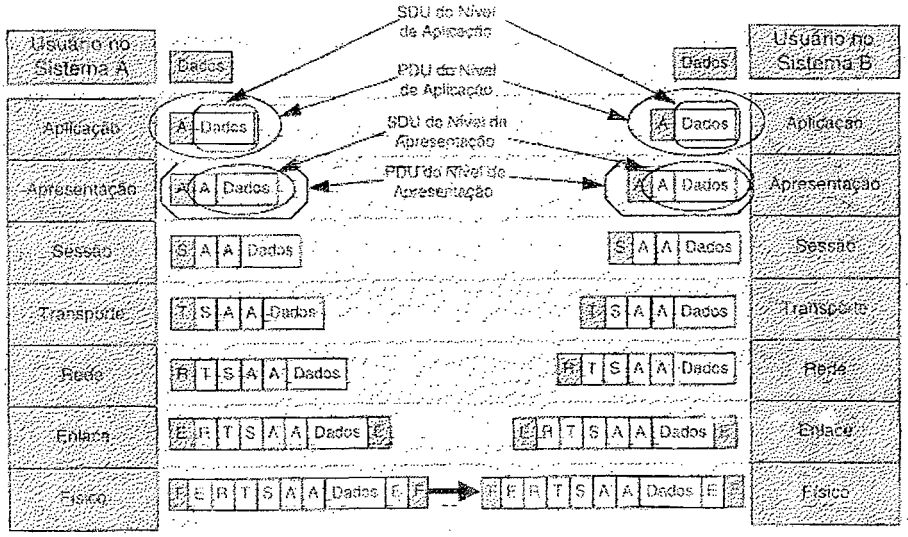

FIGURA 3. Transmissão de dados no modelo OSI
Fonte: (SOARES, LEMOS, COLCHER, 1995)

1.2 TCP/IP (RFC1155)

TCP/IP (*Transmission Control Protocol/Internet Protocol*) se refere ao conjunto de protocolos utilizados na Internet. Na verdade o termo genérico TCP/IP significa "qualquer coisa e tudo relacionado aos protocolos TCP e IP. Isto inclui outros protocolos (UDP, ARP, ICMP) e aplicações (TELNET, FTP)". Inclui também uma série de padrões que especificam como os computadores vão se comunicar e cria convenções para interconectar a rede e para o roteamento através dessas conexões. Os protocolos da Internet (IP) são o resultado de um projeto da DARPA (*Defense Advanced Research Projects Agency*) sobre conectividade entre redes no final dos anos 60. Ele foi utilizado em todas as redes de longa distância do sistema de defesa dos EUA em 1983, mas não foi amplamente aceito até ser incorporado ao BSD (*Berkeley Software Distribution*).

A popularidade do TCP/IP é baseada principalmente em:

- estrutura cliente/servidor robusta – O TCP/IP é uma excelente plataforma cliente/servidor, especialmente em redes WAN (*wide-area Network*);
- compartilhamento de informações – Permite que milhares de organizações militares, educacionais, científicas e comerciais compartilhem dados, correio eletrônico (*e-mail*) e outros serviços na Internet.
- Ampla disponibilidade – Implementações do TCP/IP estão disponíveis em praticamente todos os sistemas operacionais populares. Seu código fonte é amplamente disponível em várias implementações. Fabricantes de *bridges*, roteadores e analisadores de redes oferecem suporte para o TCP/IP em seus produtos.

O desenvolvimento, arquitetura e futuro do TCP/IP estão diretamente ligados aos avanços e administração da Internet. Embora nenhuma organização seja dona da Internet ou de suas tecnologias, muitas são responsáveis por sua direção. Dentre elas destacamos as ligadas à utilização do TCP/IP na Internet:

- *Internet Society* (ISOC) – É uma organização global, criada em 1992, com o propósito de promover o desenvolvimento de novas tecnologias e aplicações para Internet, bem como definir os padrões de protocolo que permitam o funcionamento da Rede;
- *Internet Architecture Board* (IAB) - É um grupo técnico da *Internet Society* responsável por publicar RFCs (*Request for Comments*) e supervisionar os padrões estabelecidos para a Internet. O IAB é responsável pelo IETF (*Internet Engineering Task Force*), IANA (*Internet Assigned Number Autority*) e o IRTF (*Internet Research Task Force*);

O IETF é responsável pelo desenvolvimento de soluções para problemas técnicos na Internet, bem como o desenvolvimento de padrões e protocolos a serem utilizados.

O IANA supervisiona e coordena a atribuição dos identificadores de protocolos utilizados na Internet.

O IRTF é responsável por coordenar todos os projetos de pesquisa relacionados ao TCP/IP.

Request for Comments (RFCs) – Os padrões definidos para o TCP/IP são publicados em uma série de documentos chamados RFCs, que descrevem o funcionamento interno da Internet. Embora todos os padrões para o TCP/IP sejam publicados através de RFCs, nem todas elas especificam padrões. Padrões para o TCP/IP não são desenvolvidos por um comitê, mas são resultado de um consenso, uma vez que qualquer membro da *Internet Society* pode submeter um documento para publicação como uma RFC. O documento apresentado é avaliado por um grupo técnico e recebe uma classificação, que especificará se este deve ou não ser considerado com um padrão.

Existem 5 classificações para as RFCs:

- Requeridas – tem necessariamente que ser implementadas por todos os *hosts* e *gateways* baseados em TCP/IP;
- Recomendadas – conforme o próprio nome especifica, é recomendado embora não obrigatório que todos os *hosts* e *gateways* baseados em TCP/IP implementem essas RFCs. Normalmente elas são implementadas;
- Eletivas – a implementação dessas RFCs é opcional.
- Uso Limitado – não é disponibilizada para uso geral;
- Não recomendadas – sua implementação não é recomendada.

Quando um documento é publicado, a ele é atribuído um número de RFC. A RFC original nunca é atualizada. Se alguma mudança se fizer necessária uma nova RFC é publicada com um novo número. Sendo assim, é importante verificar se estamos utilizando a mais recente RFC de um determinado tópico. O IAB publica periodicamente o IAB *Official Protocol Standard*, que deve ser utilizado para determinar qual a RFC corrente para cada protocolo.

A arquitetura TCP/IP

A arquitetura TCP/IP baseia-se principalmente na idéia de que não existe nenhuma tecnologia de rede que atenda a toda a comunidade de usuários. Alguns usuários precisam de redes de alta velocidade que geralmente cobrem uma área geográfica restrita, enquanto outros, utilizam redes de baixa velocidade que conectam equipamentos a milhares de quilômetros de distância. Sendo assim, a única forma de permitir que esses usuários troquem informações é conectar todas essas redes, formando uma inter-rede. (SOARES, LEMOS, COLCHER, 1995)

Para que duas redes possam ser interligadas, há necessidade de se conectar uma máquina a cada uma delas, que deverá ficar responsável pela tarefa de transmissão de dados entre as redes. Essa máquina é chamada de internet *gateway* ou internet *router*. Para que os *gateways* possam efetuar o roteamento das mensagens eles precisam saber como as diversas redes estão interconectadas.

A arquitetura TCP/IP se baseia em um modelo com quatro camadas, onde cada uma executa um conjunto bem definido de funções de comunicação. No modelo em camadas TCP/IP, não existe uma estruturação formal para cada camada, conforme ocorre no modelo OSI. Procura-se definir um protocolo próprio para cada camada, assim como a interface de comunicação entre duas camadas adjacentes.

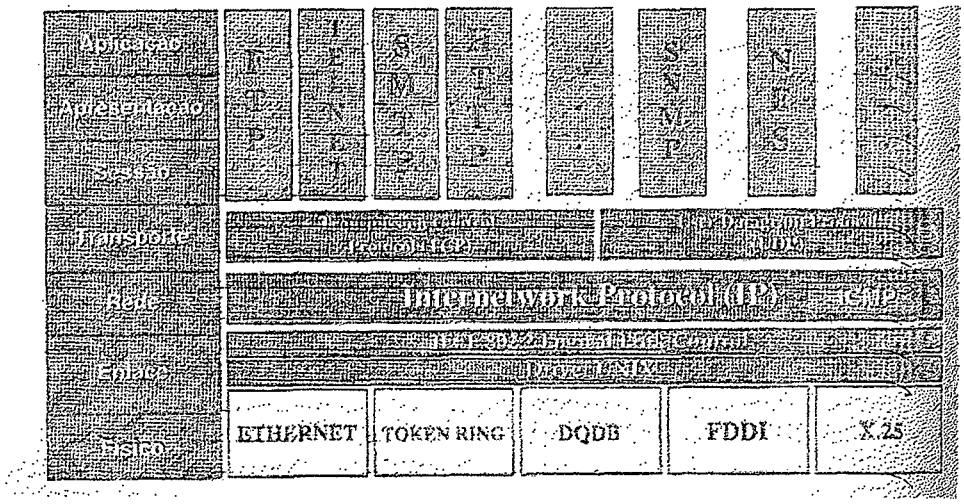

FIGURA 4. Modelo da Arquitetura TCP/IP
Fonte: (SOARES, LEMOS, COLCHER, 1995)

A camada interface de rede - Na camada de rede de comunicação do TCP/IP, não existe um padrão para a sub-rede de acesso, possibilitando a conexão de qualquer tipo de rede, desde que haja uma interface que compatibilize a tecnologia da rede com o protocolo IP. Desta forma, um número muito grande de tecnologias pode ser utilizado na sub-rede de acesso, como Ethernet, Token Ring, FDDI, X.25, Frame Relay, ATM, etc.

Para que todas estas tecnologias possam ser "vistas" pela rede inter-rede, existe a necessidade de uma conversão de endereçamentos do formato utilizado pela sub-rede e o formato IP. Esta conversão é realizada pelos *gateways*, que tornam a interconexão das redes transparente para o usuário. Além das conversões de protocolos, os *gateways* são responsáveis pela função de roteamento das informações entre as sub-redes.

A camada de inter-rede - também chamada de Internet, é equivalente à camada de rede do modelo OSI. Nela são especificados vários protocolos, dentre os quais se destaca o IP (*Internet Protocol*).

O IP é um protocolo não orientado à conexão, cuja função é transferir blocos de dados denominados datagramas da origem até o destino, podendo passar inclusive por várias sub-redes (a origem e o destino são *hosts* identificados por endereços IP). A operação no modo datagrama é uma comunicação não confiável, não sendo usado nenhum reconhecimento fim a fim ou entre nós intermediários, nem qualquer tipo de controle de fluxo. Nenhum mecanismo de controle de erro de dados é utilizado, apenas um controle de verificação do cabeçalho, para garantir que os *gateways* encaminhem as mensagens corretamente.

A camada de transporte - A camada de transporte tem o objetivo de prover uma comunicação confiável entre dois processos, estando eles ocorrendo dentro da mesma rede ou não. Ela deve garantir que os dados sejam entregues livres de erros, em sequência e sem perdas ou duplicação.

A Arquitetura TCP/IP especifica dois tipos de protocolos na camada de transporte: o UDP (*User Datagram Protocol*) e o TCP (*Transmission Control Protocol*). O UDP é um protocolo não orientado à conexão que pode ser considerado como uma extensão do protocolo IP, e não oferece nenhuma garantia em relação à entrega dos dados ao destino.

Já o protocolo TCP oferece aos seus usuários um serviço de transferência confiável de dados, através da implementação de mecanismos de recuperação de dados perdidos, danificados ou recebidos fora de sequência, minimizando o atraso na sua transmissão.

A camada de Aplicação - As aplicações na arquitetura TCP/IP, ao contrário do que ocorre com as OSI, são implementadas de uma maneira isolada, ou seja, não existe um padrão que defina como deve ser estruturada uma aplicação. As aplicações interagem com o nível de transporte para enviar e receber dados e podem usar serviços orientados à conexão oferecidos pelo TCP, ou não orientados à conexão fornecidos pelo UDP. Fazem parte desta camada aplicações como : FTP, HTTP, TELNET, DNS, NFS, SMTP, SNMP, etc.

Comparação entre as arquiteturas OSI e Internet TCP/IP

A diferença mais observável é o número de camadas. Na arquitetura OSI e TCP/IP, temos sete e quatro, respectivamente. Alguns dos serviços definidos para o modelo OSI são opcionais, como o nível de enlace, rede e transporte que podem oferecer serviços orientados à conexão(circuito virtual) ou não-orientados à conexão (datagrama). Isto é consequência do modelo procurar tratar todos os aspectos de interconexão aberta de sistemas. Já a arquitetura TCP/IP foi desenvolvida para solucionar um problema prático: interligar redes com tecnologias distintas. E para tal foram desenvolvidos protocolos específicos. Os níveis físico, de enlace, e o nível de rede do modelo OSI, relativos à transmissão de dados, não são abordados na arquitetura TCP/IP, que os agrupa na camada de inter-rede, se limita a definir uma interface entre o nível intra-rede e o nível inter-rede. (SOARES, LEMOS, COLCHER, 1995).

Os serviços relativos à interconexão de redes do modelo OSI são tratados pelo protocolo IP na arquitetura TCP/IP.

No nível de transporte a arquitetura TCP/IP oferece duas opções : o TCP (circuito virtual) e o UDP (datagramas). Esses protocolos são equivalentes aos protocolos orientados e não-orientados à conexão do modelo OSI no nível de transporte.

A camada de aplicação da arquitetura TCP/IP é um aglomerado das camadas de sessão, apresentação e aplicação do modelo OSI, não permitindo assim uma maior reutilização de esforços durante o desenvolvimento de aplicações distribuídas. (Idem)

Os protocolos da arquitetura TCP/IP implementam uma solução simples e funcional, para o problema ao qual propuseram solucionar. Como seus protocolos não são soluções proprietárias, essa arquitetura se tornou um padrão **de fato**.

1.3 TCP (TRANSMISSION CONTROL PROTOCOL)

O TCP é um protocolo da camada de transporte orientado à conexão que fornece um serviço confiável de transferência de dados fim a fim. O TCP foi projetado para funcionar com base em um serviço de rede sem conexão e sem confirmação.

O TCP interage de um lado com processos das aplicações e do outro com o protocolo da camada inter-rede da arquitetura Internet. A interface entre os processos de aplicação e o TCP consiste em um conjunto de chamadas semelhantes às que os sistemas operacionais fornecem aos processos para abrir e fechar conexões e para enviar e receber dados em conexões previamente estabelecidas. As aplicações podem usar o serviço orientado à conexão, fornecido pelo TCP (serviço de circuito virtual), ou o serviço não orientado à conexão, fornecido pelo UDP (*User Datagram Protocol*).

A interface entre o TCP e à camada inferior define um mecanismo através do qual as duas camadas trocam informações assincronamente. O nível inter-rede é responsável pela transferência de dados através de dados da inter-rede, desde a máquina origem até a de destino. O protocolo utilizado por este nível é o protocolo IP. O protocolo IP (camada de Inter-rede) , e os protocolos de nível físico (CSMA/CD, *token ring*, FDDI etc) caracterizam-se por fornecer apenas uma comunicação insegura entre as máquinas, onde os pacotes (ou datagramas) podem

ser perdidos ou duplicados por erros na transmissão. Nenhuma verificação é feita. Todavia, a maioria dos programas aplicativos precisa enviar grandes quantidades de dados, e deixar a cargo da aplicação quaisquer verificação é problemático.

O TCP é um protocolo de propósito geral, que ajuda a isolar os programas aplicativos dos detalhes da transmissão em rede, fornecendo uma interface uniforme para as aplicações.

O pacote TCP

O protocolo TCP vê os dados como uma sequência de *bytes*, que são divididos em segmentos para transmissão. O segmento é a unidade de transferência do TCP.

Cada segmento é divido em duas partes: o cabeçalho e os dados. Os campos da porta origem e destino contém o número da porta na máquina origem e na destino, respectivamente. O número de sequência identifica a posição do segmento na cadeia de dados enviada pelo servidor. O número de reconhecimento (ACK) identifica a posição do maior segmento recebido. Tanto o número de reconhecimento quanto o de sequência são utilizados pelo protocolo para fornecer segurança na transmissão. O campo de deslocamento (OFF) contém um inteiro que especifica o deslocamento da área de dados do segmento. Ele é necessário devido a um outro campo, o campo de opções (OPTIONS), que pode fazer com que o tamanho do segmento varie. Em outras palavras, o tamanho do segmento TCP pode variar.

O campo de código (CODE) do segmento TCP é utilizado para informar qual o objetivo e o conteúdo do segmento: estabelecer ou fechar uma conexão, enviar dados, etc. Este campo vai determinar como interpretar outros campos, de acordo com o valor de seus 6 *bits*, mostrados na tabela abaixo:

BIT	SIGNIFICADO
URG	Campo de ponteiro de urgência
ACK	Campo de ACK
PSH	Este segmento requer ou não um PUSH
RST	Reseta a conexão
SYN	Sincroniza os números de sequência
FIN	Transmissor chegou ao fim desta cadeia de *bytes*

QUADRO 1. Valores dos *bits* dos cabeçalhos de segmentos

O TCP permite que o transmissor indique que determinado segmento é urgente, ou seja, precisa ser entregue o mais rápido possível. O mecanismo usado para marcar dados como urgente consiste no bit de código URG e no campo de ponteiro urgente (URGENT POINTER). Quando o bit URG está setado para 1, o campo de ponteiro urgente especificará a posição na cadeia de *bytes* onde os dados urgentes terminam. Este tráfego de dados urgentes também pode ser referenciado como tráfico *Out-of-Band.*

O campo WINDOW indica o tamanho do *buffer*, ou seja, quantos dados ele poderá receber de cada vez.

O campo de OPÇÕES é utilizado para que à camada TCP do transmissor e receptor possam se comunicar, de modo a poderem definir o tamanho de segmento máximo. A definição deste tamanho máximo viabiliza a comunicação entre uma máquina com uma pequena capacidade de *buffer*, e outra com uma grande capacidade. A escolha de um tamanho de segmento muito grande ou muito pequeno pode prejudicar o desempenho da rede. A utilização de segmentos muito pequenos tendem a fazer a utilização da rede cair. Já segmentos muito grandes resultam, em níveis mais baixos de protocolo, em datagramas muito grandes, que são fragmentados para transmissão, mas ao contrário dos datagramas propriamente ditos, estes fragmentos deverão todos chegar ao destino, caso contrário, todos deverão ser retransmitidos. Aumentar o tamanho do segmento vai, além disso, diminuir o *throughput* da rede.

Stream orientation e conexão full-duplex

O TCP é capaz de transferir uma cadeia (*stream*) contínua de octetos (*bytes*) nas duas direções (*Full-Duplex*), entre seus usuários. O serviço entrega ao receptor exatamente a mesma cadeia (sequência de octetos), na mesma sequência, que foi enviada pelo transmissor.

Transmissão utilizando buffer

O protocolo TCP armazena dados a serem transmitidos/recebidos em *buffers*, e age no sentido de fornecer segurança nas transmissões de dados, dividindo grandes quantidades de dados em pequenas porções, reagrupadas no receptor, e fazendo verificação de erros durante a transmissão. Normalmente o TCP decide o momento de parar de agrupar os octetos de, consequentemente, transmitir o segmento formado por esse agrupamento. Porém, caso deseje, os usuários do TCP podem fazer uso da função *push* que faz com que o TCP transmita imediatamente os octetos que estão nos seus *buffers* aguardando transmissão.

Estabelecimento de conexão (orientado à conexão)

Antes de iniciar a transferência, ambas as máquinas origem e destino interagem, enviando mensagens pela rede, para verificar se a transferência está autorizada e se ambos os lados estão preparados para transmitir. Decididos os detalhes, é dito que uma conexão foi estabelecida entre os pontos. E mesmo depois de estabelecida a conexão, durante a transmissão propriamente dita, os pontos permanecem trocando informações de controle a fim de verificar a consistência dos dados transmitidos.

Segurança na transmissão (garantia de entrega dos dados)

O TCP não exige um serviço de rede confiável para operar, logo, responsabiliza-se pela recuperação de dados corrompidos, perdidos, duplicados, ou entregues fora de ordem pelo protocolo inter-rede. O TCP garante a entrega correta dos segmentos de dados transferidos. Ele faz isto através do estabelecimento de uma conexão, e através da troca de informações de controle. Conceitualmente, cada octeto transmitido é associado a um número de sequência. O número de sequência do primeiro octeto dos dados contidos em um segmento é transmitido junto com o segmento e é denominado número de sequência do segmento. Os segmentos carregam de carona (*piggybacking*) um reconhecimento. O reconhecimento constitui-se do número de sequência do próximo octeto que a entidade TCP transmissora espera receber do TCP receptor na direção oposta da conexão. Este mecanismo também é conhecido como *Positive Acknowledgement with Retransmission*. Outra técnica utilizada pelo TCP para garantir a entrega dos segmentos é o uso de *Time Out*. Quando uma entidade TCP transmite um segmento contendo dados, ele coloca uma cópia do segmento em uma fila de retransmissão e dispara um temporizador. Quando o reconhecimento da recepção dos dados é recebido, o segmento é retirado da fila. Se, por outro lado, o reconhecimento não for recebido antes do temporizador esperar, o segmento é retransmitido. A quantia de tempo que a entidade TCP espera antes de retransmitir o segmento é chamada de *Time Out*.

Controle de fluxo

O TCP provê meios para que o receptor possa determinar o volume de dados que o transmissor pode lhe enviar, ou seja, para controlar o fluxo dos dados. O mecanismo de controle de fluxo baseia-se no envio, junto como o reconhecimento, do número de octetos que o receptor tem condições de receber

(tamanho da janela de recepção), contados a partir do último octeto da cadeia de dados recebido com sucesso. Com base nessa informação o transmissor analisa sua janela de transmissão, ou seja, calcula o número de octetos que pode enviar antes de receber outra liberação

Como o TCP sabe que um segmento não foi danificado?

No TCP receptor, os número de sequência são usados para ordenar os segmentos que porventura tenham sido recebidos fora de ordem, e para eliminar segmentos duplicados. Para tratar os segmentos corrompidos é adicionado um *checksum* a cada segmento transmitido. No receptor é feita uma verificação e os segmentos danificados são descartados.

Dentro do cabeçalho de segmentos TCP existe um campo chamado *Checksum*. Este campo possui um inteiro de 16 *bits*, que é usado para verificar a integridade de todo o segmento. Para calculá-lo, o TCP do transmissor prepara um *pseudo-header*, ou pseudo-cabeçalho, para o segmento, e acrescenta *bytes* com valor zero suficientes para tornar o segmento múltiplo de 16 *bits*, e calcula o *checksum* sobre o resultado. O TCP não conta os zeros adicionados no tamanho do segmento, nem os transmite. Ele também assume que o campo de *checksum* é zero, a fim de realizar o seu cálculo. No lado do receptor, o TCP realiza o mesmo cálculo para verificar se o segmento chegou intacto.

O uso de pseudo-cabeçalho permite verificar se o segmento chegou ao seu destino certo, incluindo o endereço Internet e o número da porta. (SOARES, LEMOS, COLCHER, 1995)

No pseudo-cabeçalho o campo PROTO especifica o tipo TCP, e o campo TCP *Lenght* especifica o tamanho total do segmento TCP. Na recepção, a informação usada no pseudo-cabeçalho é extraída do pacote IP, que transmitiu o segmento.

Three-way Handshake

O estabelecimento de uma conexão de transporte seria simples se a rede não perdesse, armazenasse ou duplicasse pacotes. Para resolver estes tipos de problemas estabeleceu-se um tempo de vida máximo na rede para um pacote, isto é, passando esse tempo ele é destruído. O tempo de vida de um pacote (T) estaria relacionado a um tempo de nascimento (*timestamp*) com a hora de sua criação, e pode ser limitado usando-se um contador de saltos em cada pacote, incrementado cada vez que passa por um nó intermediário da rede.

A idéia básica dos métodos propostos para solução deste problema é que duas unidades de dados do protocolo de transporte (segmentos, ou, também chamados T-PDU) com numeração idêntica nunca estejam pendentes ao mesmo tempo. Para tanto, as numerações dos segmentos não se repetem dentro do período T definido anteriormente. Toda vez que é feito um pedido de conexão por uma entidade de transporte E1, é enviado um número inicial de sequência para os segmentos. O destino (entidade de transporte E2) ao receber esse segmento deve responder confirmando o número e estabelecendo o número inicial de sequência a partir do qual seus segmentos serão numerados. A entidade E1 deve confirmar esse número através de um reconhecimento, que pode ser transportado em seu primeiro pacote de dados.

O primeiro segmento do estabelecimento da conexão pode ser identificado através do bit SYN, no campo de código (CODE) do cabeçalho TCP. Na segunda mensagem os *bits* SYN e ACK foram setados, indicando a confirmação do primeiro segmento, bem como continuando o estabelecimento da conexão. A mensagem final é somente uma confirmação, e é usado apenas para informar que ambas as partes estão de acordo com a conexão que está sendo estabelecida.

A técnica apresentada para resolver o problema da duplicação de pacotes é conhecida como *three-way handshake* e funciona da seguinte maneira: suponha que a entidade E1 faça um pedido de conexão escolhendo "*a*" como o número inicial da sequência de seus segmentos. Como a resposta ao pedido atrasou, E1 envia novo pedido de conexão, agora propondo o número de sequência "*b*". Note que tanto "*a*"

quanto "*b*" não podem ser repetidos dentro de um período T. Ao receber o pedido de conexão E2 o aceita, confirmando o número de sequência "*b*" e estabelecendo o número de sequência "*c*" para seus segmentos. Ao receber a confirmação da conexão, E1 envia seu primeiro segmento, confirmando o número de sequência "*c*". Suponha que a resposta a esse segmento demore tanto que E1 o retransmite. Quando E2 recebe esse segmento a conexão é estabelecida e os dados são trocados até que a conexão seja desfeita.

Suponha que após a conexão ser desfeita chegue o primeiro pedido de conexão atrasado enviado por E1 (número de sequência *"a"*). A entidade E2 o aceita, confirmando o número de sequência "*a*" e estabelecendo o número de sequência "*k*" para seus segmentos. Em seguida, a entidade E2 recebe o pacote de dados atrasado enviado por E1, carregando a confirmação de sua sequência "*c*". O fato de "*c*" ter sido confirmado ao invés de "*k*" faz com que E2 descarte o segmento recebido. Por sua vez, ao receber uma confirmação de conexão de E2 confirmando o número de sequência "*a*", E1 a rejeita, avisando a E2 que se trata de uma duplicata.

Como as conexões do TCP são *full-duplex*, para encerrar uma conexão, ambos os lados devem desejá-lo. Se apenas um dos lados enviar o pedido de fim de conexão, feito através de um segmento cujo *bit* FIN do campo de código está setado, apenas um sentido do fluxo de dados é terminado. Supondo a conexão entre duas entidades TCP E1 e E2, se E1 enviar um segmento contendo o bit FIN setado, apenas o fluxo no sentido E1 - E2 estará encerrado. A comunicação no sentido E2 - E1 permanece. Somente quando ambas as direções tiverem sido encerradas (quando E2 enviar um segmento com FIN setado para E1), a conexão será apagada. Algumas anormalidades na rede podem forçar um programa a quebrar a conexão. O TCP fornece uma operação de *reset* para tais casos. Um lado da conexão inicia um *reset* enviando um segmento com o *bit* RST do campo de código igual a 1. O outro lado responde a este segmento imediatamente, abortando a conexão. Este também informa ao programa que um *reset* ocorreu. Abortar a conexão significa cessar a transferência em ambas as direções imediatamente, e liberar todos os recursos alocados nos *buffers*.

Acknowledgement e time out

Cada segmento enviado pelo transmissor é identificado por um número de sequência, que indica a sua posição na cadeia de dados a ser enviada. Ao receber um segmento, o receptor testa o campo de *checksum*, a fim de comprovar sua integridade. Feito isso, o receptor envia para o transmissor uma confirmação positiva de que o segmento foi recebido com sucesso. Esta se dá através do envio do número de sequência recebido no campo de ACK do segmento de confirmação.

Supõe-se uma entidade de transporte E1 que esteja com uma conexão estabelecida com uma entidade E2 de mesmo nível. Quando E1 envia um segmento para E2, é disparado um contador de tempo. A entidade E1 fica esperando pela confirmação (ACK) do segmento enviado até que o contador chegue ao limite de tempo (*time out*). Caso o reconhecimento chegue neste período de tempo, outro segmento poderá ser enviado, e a comunicação continua normalmente. Caso o *time out* seja alcançado, o segmento enviado é retransmitido.

Vários fatores podem causar a retransmissão de um segmento. Entre eles, a perda de um segmento durante a transmissão, devido a algum erro na rede, ou a demora na entrega do segmento ou da confirmação. A demora na entrega do segmento pode ser causada por uma rede muito lenta, ou sobrecarregada. Neste caso, outro problema surge: a duplicação de pacotes. Este problema consiste na recepção de dois ou mais segmentos com o mesmo número de sequência, ou seja, repetidos. Durante a transmissão, apenas um destes segmentos duplicados é aceito, sendo que os demais são descartados. Já durante o estabelecimento da conexão, este problema pode receber um tratamento um pouco diferente.

Uma vantagem do reconhecimento é que ambos (número de sequência e ACK) são fáceis de gerar e não-ambíguos.

Janelas deslizantes (*sliding windows*)

O TCP utiliza o protocolo de *bit* alternado, ou seja, o transmissor descobre que ocorreu um erro no segmento por ele enviado quando seu reconhecimento não chega antes do *time out* finalizar. Porém, para aumentar a eficiência da utilização dos canais de comunicação, foram elaborados protocolos que permitem que o transmissor envie vários segmentos mesmo sem ter recebido o reconhecimento do segmentos anteriormente enviados. O número máximo de segmentos que podem ser enviados, sem que tenha chegado um reconhecimento, definem a largura da janela de transmissão, sendo que este tamanho pode ser 1 ou maior que 1.

O TCP utiliza a técnica de *sliding window* para resolver dois problemas: eficiência de transmissão (tornando possível enviar múltiplos segmentos antes da chegada do ACK) e controle do fluxo dos dados. As *sliding windows* do TCP operam a nível de *byte* e não a nível de segmento. Os *bytes* da cadeia de dados são numerados sequencialmente e são associados três ponteiros à janela deslizante. O primeiro, a esquerda da janela, separa *bytes* que foram enviados e reconhecidos dos que estão sendo enviados. O segundo, a direita da janela, determina o *byte* da sequência dos que ainda podem ser enviados antes de receber o reconhecimento. O terceiro marca o limite entre os *bytes* que serão enviados dos que não poderão ser enviados enquanto a janela não deslizar.

Uma consequência do uso de janelas deslizantes é o aumento no *throughput* total da rede.

Portas TCP

Para permitir que vários processos em um único *host* possam simultaneamente transmitir cadeias de dados, ou seja, possam simultaneamente usar seus serviços, o TCP utiliza o conceito de porta. As portas são identificadas por um número inteiro de 16*bits*. Cada um dos usuários (processos de aplicação) que o TCP está atendendo em um dado momento é identificado por uma porta diferente. Como

os identificadores de portas são selecionados isoladamente por cada entidade TCP, eles podem não ser únicos na inter-rede. Para obter um endereço que identifique univocamente um usuário TCP, o identificador da porta é concatenado ao endereço IP onde a entidade TCP está sendo executada, definindo um *socket*. Um *socket* identifica univocamente um usuário TCP em toda a inter-rede.

A associação de portas a processos é tratada independentemente por cada entidade TCP. Entretanto, processos servidores que são muito usados (FTP, Telnet etc.) são associados a portas fixas e reservadas, e são então divulgadas para os usuários. Algumas destas portas são listadas abaixo:

PORTA	**APLICAÇÃO**
20	Dados do FTP
21	FTP - FILE TRANSFER PROTOCOL
23	TELNET
25	SMTP - SIMPLE MAIL TRANSFER PROTOCOL
53	DNS - DOMAIN NAME SERVER
79	FINGER
80	HTTP - HYPER TEXT TRANSFER PROTOCOL
111	PORTMAPPER
113	IDENTIFICATION
119	NNTP _ NETWORK NEWS
512	REXEC _ REMOTE EXEC
513	RLOGIN _ REMOTE LOGIN
514	RSH _ REMOTE SHELL

QUADRO 2. Portas de Comunicação TCP

Sockets

Um *socket* identifica univocamente um usuário TCP em toda a rede. Uma conexão entre dois pontos é identificada pelo par de *sockets* de suas extremidades. Um *socket* local pode participar de várias conexões diferentes com *sockets* remotos.

Os mecanismos utilizados nas funções de controle de erros e de fluxo exigem que o TCP inicie e mantenha informações de estado para cada conexão estabelecida. O conjunto dessas informações - os *sockets*, os números de sequência, o tamanho das janelas etc - define a conexão.

Quando dois processos desejam comunicar-se, as instâncias do TCP às quais eles estão associados devem estabelecer uma conexão. Esta conexão é estabelecida através da conexão entre os *sockets* de cada extremidade.

Os processos de aplicação transmitem seus dados fazendo chamadas ao TCP, passando como parâmetros os *buffers* onde estão os dados. Estas chamadas são feitas através dos *sockets* envolvidos na conexão. O TCP empacota os dados armazenados nos *buffers* em segmentos e chama o módulo IP para transmitir os segmentos para o TCP destino. O TCP receptor coloca os dados recebidos em segmentos nos *buffers* do usuário destinatário e notifica-o da entrega, tudo isso também através de *sockets*.

O TCP assume que opera como um módulo do sistema operacional. O TCP assume também que a interface com a rede é controlada por um *device driver*. O TCP **não** acessa o *device driver* da rede diretamente, mas interage com o módulo IP que é o responsável pelo acesso à rede feito através desse *device driver*. A interface oferecida aos usuários de seus serviços baseia-se em chamadas para abrir (*open*) ou fechar (*close*) uma conexão, para enviar (*send*) ou receber (*receive*) dados, ou para obter informações sobre o estado (*status*) de uma conexão. Essa chamadas são semelhantes às que os processos de aplicação fazem ao sistema operacional, por exemplo, para abrir (*open*), ler (*read*) e fechar (*close*) um arquivo. No caso de acesso a arquivos, utiliza-se descritores, que, no caso das conexões de rede, são os *sockets*. Estes fazem o papel de descritores de uma conexão.

1.4 IP (INTERNET PROTOCOL)

O IP foi projetado para permitir a interconexão de redes de computadores que utilizam a tecnologia de comutação de pacotes. Sua função é transferir pacotes ou blocos de dados denominados datagramas da origem para o destino, que são *hosts* identificados por endereço IP. Possui também, um serviço de fragmentação e remontagem de datagramas longos, para transmissão através de redes onde o tamanho máximo permitido para o pacote é pequeno. (GORKI, 1999)

Cada datagrama é independente, ou seja, não possui relacionamento com qualquer outro datagrama, já que o serviço oferecido pelo IP não orientado à conexão. Nenhum mecanismo de controle de fluxo é empregado, como também nenhum controle de erro é efetuado com relação aos dados transmitidos, exceto um *checksum* do cabeçalho, que garante que as informações nele contidas, que são usadas pelos *gateways* para encaminhar os datagramas, estão corretas.

Segundo Soares algumas das principais características desse protocolo são:

1. Serviço de datagrama não confiável.
2. Endereçamento hierárquivo.
3. Facilidade fragmentação e remontagem de pacotes.
4. Identificação da importância do datagrama e do nível de confiabilidade exigido.
5. Identificação da urgência de entrega e da ocorrência futura ou não de pacotes na mesma direção (pré-alocação, controle de congestionamento).
6. Campo especial indicando qual protocolo a ser usado no nível superior.
7. Roteamento adaptativo distribuído nos *gateways*.
8. Descarte e controle de tempo de vida dos pacotes inter-rede no *gateway*.

Endereços IP

O endereço IP corresponde a um número de 4 *bytes* separados por três pontos que representam uma conexão na inter-rede e não uma máquina individual, ou seja, um *gateway* que conecta várias redes deverá ter vários IP's , um para cada conexão. A primeira parte desse número identifica uma rede específica e a segunda um *host* dentro dessa rede, por exemplo 128.0.0.0.

O IP utiliza 5 classes diferentes para representar os endereços. A utilização de classes deve-se ao tamanho das redes que variam desde uns poucos computadores em redes locais até milhares como no caso de grandes redes públicas.

Classe A

O primeiro *byte* representa a rede (o bit mais significativo tem sempre o valor 0) e os três restantes o endereço local. Essa classe de endereços é utilizada para redes de grande porte, e o valor do *byte* que a identifica deve estar entre 1 e 126. Cada uma dessas 126 redes pode ter até 16 milhões de *hosts*. O DARPA é um exemplo de redes com endereço classe A.

Classe B

Essa classe utiliza dois *bytes* para representar a rede e os outros dois para endereçamento de *hosts*. Os endereços da classe B variam de 128.1 a 191.254 (os números 0 e 255 no segundo *byte* e 127 no primeiro são reservados para funções especiais). Isto permite 16.384 redes e aproximadamente 65.000 por rede.

Classe C

Essa classe de endereços utiliza os três primeiros *bytes* para representar a rede e o último para representar os *host*. Normalmente é utilizada para pequenas rede locais (*LANs*). Permite aproximadamente 2 milhões de redes com 254 *hosts* em cada uma.

Classe D

A classe D representa os endereços IP cujo número é superior a 224 e está reservado para criar agrupamentos de computadores para o uso de *Multi Cast*. O sistema *Multi Cast* permite que um grupo de computadores utilize um ou mais endereços para enviar dados somente para os computadores que estejam configurados para receber por este endereço. Não podem ser utilizado para endereçar os computadores de usuários na rede TCP/IP.

Classe E

A classe E é um endereço reservado e utilizado para testes e novas implementações e controles do TCP/IP. São endereços IP com valores iniciais acima de 240.0.0.0. Não podem ser utilizado para endereçar os computadores de usuários na rede TCP/IP.

ARP e ICMP

Dois outros protocolos com importantes funções, embora estas não estejam diretamente relacionadas com a transmissão de dados mantém a estrutura do IP e usualmente são invisíveis aos usuários e às aplicações. São eles o ARP (*Address Resolution Protocol*) e o ICMP (*Internet Control Message Protocol*).

Os cabeçalhos do IP contém tanto o endereço IP de origem quanto o de destino, mas o endereço do hardware também tem de ser conhecido. O IP obtém um endereço de *hardware* de um determinado sistema, difundindo pela rede um pacote especial de requisição (um pacote ARP de requisição) contendo o endereço IP do sistema com o qual está tentando se comunicar. Todos os nós da rede local que tiverem o ARP habilitado detectam essa difusão, e o sistema que tem o número de IP em questão envia um pacote (do tipo ARP *Reply*) contendo o seu endereço de hardware para o computador que fez a solicitação. O endereço de hardware e o endereço de IP desse computador são então armazenados na cache para uso futuro.

Como a resposta ARP também é feita na forma de difusão, é normal que outros nós usem essa informação para atualizar seus caches ARP.

O protocolo ICMP permite que 2 nós em uma rede IP compartilhem o *status* do IP (protocolo) e informação de erros. Esta informação pode ser utilizada por protocolos de alto nível para tratar problemas de transmissão ou para administradores de rede para detectar problemas na rede. Embora esteja encapsulado em pacotes IP, o ICMP não é considerado um protocolo de alto nível. O utilitário *ping* faz uso do ICMP para determinar se um certo endereço IP na rede está operacional ou não. É também utilizado por roteadores para fazer redirecionamento de pacotes. Quando uma rede possui um roteador *default* e não é encontrada uma rota específica para um datagrama, ele é enviado para esse roteador. Esse roteador informará ao módulo IP se ele não é a melhor escolha para alcançar uma determinada rede. Essa mensagem, chamada de *redirect* no ICMP, carrega como parâmetro o endereço do roteador que é a escolha correta. O módulo IP ao receber uma mensagem *redirect*, adiciona uma entrada em sua tabela de roteamento associando a rede de destino ao endereço do roteador recebido na mensagem. A partir de então, todas as referências à rede em questão serão corretamente encaminhadas ao novo roteador associado a ela.

Funcionamento do IP

O protocolo IP funciona de maneira relativamente simples usando mensagens divididas em datagramas de no máximo 64 K*byte*s cada um. Cada datagrama é transmitido (possivelmente fragmentado em unidades menores), e sendo reagrupadas quando chegam ao destino. Este processo de fragmentar e remontar as mensagens também fica a cargo da camada de transporte (TCP).

Cada datagrama IP consiste em uma parte de cabeçalho (composta por diversos campos, explicados abaixo) e outra de texto. O cabeçalho tem uma parte fixa de 20 *byte*s e uma parte opcional de comprimento variável.

O Campo TIPO DE SERVIÇO permite que o *host* informe a sub-rede que tipo de serviço deseja. São possíveis diversas combinações de confiabilidade e velocidade. Para voz digitalizada, a entrega mais rápida é muito mais importante do que a correção de erros de transmissão. Para transferência de arquivos, a transmissão precisa é muito mais importante do que a entrega rápida. Também são possíveis diversas outras combinações, desde tráfego de rotina até transmissões instantâneas.

O COMPRIMENTO total inclui tudo no datagrama tanto o cabeçalho quanto os dados. O comprimento máximo é de 65.536 *bytes*.

O campo IDENTIFICAÇÃO é necessário para permitir ao *host* de destino determinar a que datagrama pertence um fragmento recém-chegado. Todos os fragmentos de um datagrama contêm o mesmo valor em IDENTIFICAÇÃO.

Em seguida vem um bit não utilizado e depois dois campos de 1*bit*. DF quer dizer DON'T FRAGMENT (não fragmentar). É uma ordem para que os *gateways* não fragmentem o datagrama, pelo fato de que o destino é incapaz de juntar novamente as partes. Por exemplo, um datagrama que esteja sendo transferido para execução em um pequeno micro poderia ser marcado com DF porque a ROM do micro espera o programa inteiro em um datagrama. Se o datagrama não puder ser transmitido através de uma rede, ele tem de ser roteado de forma a contornar a rede ou ser descartado.

MF significa MORE FRAGMENTS (mais fragmentos). Todos os fragmentos exceto o último apresentam esse bit ativado. Ele é utilizado como uma dupla verificação do campo COMPRIMENTO TOTAL, a fim de garantir que nenhum fragmento seja omitido e que todo o datagrama seja remontado.

O DESLOCAMENTO DO FRAGMENTO informa a que posição no datagrama atual pertence o fragmento. Todos os fragmentos com exceção do último em um datagrama devem ser múltiplos de 8 *bytes*, a unidade elementar de fragmento. Como são fornecidos 13 *bits*, há um máximo de 8192 fragmentos por datagrama, o que dá um comprimento máximo de datagrama igual a 65.536 *bytes*, de acordo com o campo COMPRIMENTO TOTAL.

O campo TEMPO PARA PERMANÊNCIA é um contador, usado para limitar tempos de vida de pacotes. Quando ele se torna zero, o pacote é destruído. A unidade de tempo é o segundo, sendo permitido um tempo máximo de permanência de 255 segundos.

Quando termina a montagem de um datagrama completo, à camada de rede precisa saber o que fazer com ele. O campo PROTOCOLO informa qual, dentre os diversos processos da camada de transporte, pertence o datagrama. O TCP é certamente uma possibilidade, mas podem haver outras.

A soma de verificação do cabeçalho confere apenas o cabeçalho. Essa soma de verificação tem sua utilidade justificada pelo fato de que o cabeçalho pode se alterar em um *gateway* (p.ex., pode ocorrer uma fragmentação).

O endereço de origem e o endereço de destino indicam o número da rede e o número do *host*. São usados quatro formatos diferentes, Os quatro esquemas permitem até 128 redes com 16 milhões de *hosts* cada uma, 16.384 redes com até 64k *hosts*, dois milhões de redes, presumivelmente *LANs*, tendo cada uma até 256 *hosts*, e ainda o *multicast*, no qual um datagrama é dirigido a um grupo de *hosts*. Aproximadamente 1000 redes fazem parte atualmente da internet ARPA, em locais de pesquisa, DoD e outros pontos governamentais e comerciais. Os endereços a partir de 1111 estão reservados para uso futuro.

O campo OPÇÕES é usado para informações de segurança, roteamento na origem, relatório de erros, depuração, fixação da hora e outras. Esse campo proporciona basicamente uma saída para que as versões subsequentes do protocolo incluam informações não presentes no projeto original, para que os pesquisadores experimentem novas idéias e para evitar a alocação de *bits* do cabeçalho a informações raramente necessárias.

A operação da rede é monitorada por IMPs e *gateways*. Quando ocorre alguma coisa suspeita, o evento é relatado pelo ICMP (*Internet Control Message Protocol*). Existem em torno de uma dezena de mensagens definidas. Cada mensagem é enviada em um pacote IP. Entre elas:

- *DESTINATION UNREACHABLE* - destino não localizado / entrega impossível
- *TIME EXCEDED* - pacote abandonado por exceder o número de tentativas
- *PARAMETER PROBLEM* - cabeçalho com algum campo inválido
- *SOURCE QUENCH* -desacelera a velocidade de transmissão dos *hosts*
- *REDIRECT* - erro de roteamento na mensagem

1.5 IP VERSÃO 6 – IP NEXT GENERATION

O problema de exaustão da capacidade de endereçamento previsto em agosto de 1990 no *meeting* da IETF levou à criação, em novembro de 1991, de um grupo de trabalho denominado ROAD (*Routing and Addressing Group*), que analisou e procurou soluções para este problema. O trabalho deste grupo, apresentado em março de 1992, apontou essencialmente em duas direções:

- a curto prazo: desenvolver e aplicar políticas de atribuição de endereços que permitam atribuir várias classes C a organizações que necessitem de mais do que 256 endereços mas para as quais uma classe B seja excessiva; utilizar agregações de redes de classe C de forma a evitar o anúncio de todas as redes diminuindo assim o crescimento das tabelas de *routing*.
- a longo prazo: desenvolvimento de um protocolo de rede que além de estender o espaço de endereçamento supere as limitações do IPv4 nomeadamente nos aspectos de:
    1. Encaminhamento de pacotes segundo políticas administrativas (*policy-based routing*);
    2. Controle de Fluxo;
    3. Garantias de serviço (Qualidade de Serviço Forte);
    4. Contabilização.

A solução de curto prazo encontrada foi denominada de CIDR (*Classless Inter-Domain Routing*) e é mais um balão de ar do que propriamente uma solução, pois permite apenas adiar, por mais algum tempo, a data de exaustão do espaço de endereçamento que se prevê, atualmente, que seja para meados do ano 2010. O método CIDR consiste, essencialmente, na perda do valor semântico das classes de endereços passando a informação de *routing* a ser considerada em relação ao par (rede, máscara) ou (rede, n.º de *bits* significativos) em contraste com a situação original em que a máscara era obtida implicitamente por classe de endereçamento. A atribuição de endereços passa a ser feita de uma forma hierárquica por região ou *provider*.

São reservados blocos de endereçamento para:

a) Organizações multi-regionais;
b) Europa (sob administração do RIPE);
   - América do Norte;
   - América Central e do Sul;
   - Pacífico.

Subsequentemente, parte do espaço de endereçamento regional é atribuído a *providers* que operem dentro dessa zona geográfica.

Paralelamente foram desenvolvidos protocolos de *routing* inter-domínio e intra-domínio com capacidade para utilizar agregação de informação de *routing* segundo a norma CIDR.

Em meados de 1992 a *Internet Authority Board* (IAB) publica um documento intitulado *IP version* 7 onde preconiza um esforço imediato por parte do IETF de forma a preparar um plano para a utilização futura do protocolo CLNP como a base para a versão 7 do protocolo IP. O IETF decide não aceitar esta recomendação e emitir um pedido de propostas conforme o preconizado pelo grupo de trabalho ROAD. Em resposta a este pedido formaram-se vários grupos de trabalho com vista ao estudo de possíveis sucessores do IP.

No *meeting* do IETF de novembro de 1992 decorre uma sessão denominada *Selection Criteria BOF* com o objetivo de obter consenso em relação ao critério aplicado na escolha do sucessor do protocolo IPv4. Dos dois documentos preparatórios em discussão e dos pontos de vista expressos na reunião saiu um novo documento contendo os critérios de seleção tidos como consensuais.

Dispondo da sugestão do *Internet Engineering Steering Group* (IESG) em relação ao critério técnico a tomar na escolha do próximo protocolo IP delineada em RFC 1380 e dos resultados da discussão dentro do próprio IETF anteriormente referida, o IETF considera necessário um debate alargado sobre este tópico e emite um *call for white papers.* Em resposta a esta solicitação foram recebidos vinte e três artigos contendo os pontos de vista dos setores industrial, tecnológico e comercial.

Estes documentos são os resultados da sessão *Next Generation Requirements* durante o encontro do IETF de março de 1994 em *Seattle* e as discussões tidas na lista de *mail big-internet* foram utilizados por *Frank Kastenholz* e *Craig Partridge* para rever o seu artigo inicial e produzir um documento intitulado *Technical Criteria for Choosing IP the Next Generation (IPng).* Este documento propõe uma lista de objetivos e um conjunto de critérios a serem atingidos pelo protocolo IPng. Os objetivos enunciados, a nível geral, são os seguintes:

- Simplicidade de arquitetura.
- Um protocolo comum a todos os sistemas capaz de garantir a conectividade global.
- Longevidade.
- Aumento de funcionalidade em relação ao IPv4.
- Modelo cooperativo em termos de *Internetworking.*

A seguinte lista contem critérios específicos de avaliação recomendados, sendo a ordem irrelevante, mas definindo o documento o tempo de disponibilização necessário para o item em questão:

a) Escalabilidade (suporte para um mínimo de 1012 sistemas finais).
b) Flexibilidade de topologias.
c) Performance.
d) Robustez.
e) Estratégia de transição.
f) Independência em relação ao meio físico.
g) Serviço orientado a datagrama sem garantia de entrega.
h) Configuração, administração e operação.
i) Segurança.
j) Identificação única de um nó.
k) Acessibilidade de especificações técnicas e algoritmos.
l) Suporte de *Multicast*.
m) Facilidades de extensão.
n) Distinção do serviço oferecido pela rede (qualidade de serviço, reserva de recursos, etc.).
o) Suporte de mobilidade.
p) Protocolo de controle (funcionalidades de *debug*).

A 25 de julho de 1994, é proposta uma recomendação do IPng no *meeting* da IETF em Toronto que é documentada no TFC 1752, sendo uma parte significativa do protocolo base proveniente do grupo de trabalho SIPP. Esta recomendação é aprovada a 17 de novembro de 1994 e proposta como padrão. O conjunto base de protocolos do IPv6 é aprovado e proposto para padrão em 18 de setembro de 1995.

Entretanto a idéia de criar uma rede de testes à semelhança da já existente para testes *multicast* (*MBone*) é posta em prática. Em junho de 1996 concretiza-se esta idéia com a construção da rede *6Bone*.

O protocolo IPng foi desenhado como uma evolução do protocolo IPv4. As características do IPv4 que se consideram estar na base do sucesso do protocolo foram mantidas no IPv6. Funcionalidades que não tem um bom desempenho ou que são usadas de maneira não muito frequente foram removidas ou tornadas opcionais. Algumas novas capacidades que se consideram necessárias foram adicionadas sem, no entanto, alterar os conceitos base do IPv4.

As características mais importantes do IPv6 são:

a) Extensão das capacidades de endereçamento e *routing*

O tamanho de endereços passa de 32 *bits* na versão 4 para 128 *bits* no IPv6, o que permite o suporte de um número muito superior de nós finais, uma melhor hierarquização do espaço de endereçamento essencial à escalabilidade do *routing* e uma maior facilidade em termos de auto-configuração visto que permite a utilização de endereços IEEE-802 embebidos em endereços IPv6. A escalabilidade dos endereços *multicast* é também aumentada através da utilização de um campo de defíne o âmbito de alcance do datagrama. Os endereços podem ser *unicast* (globais, locais, *link* e de IPv4-compatíveis), *multicast* (*one-to-many*), de *cluster* (*anycast*: *one-to-nearest*) ou reservados.

Como curiosidade a capacidade total de endereçamento do novo protocolo é de 340.282.366.920.938.463.374.607.431.768.211.456 endereços, o que dá 665.570.793.348.866.943.898.599 endereços por m^2 do planeta Terra. No entanto, e tendo em conta as políticas de atribuição de endereços que possam vigorar, a visão mais pessimista prevê que venham a existir "apenas" 1564 endereços por m^2.

b) Simplificação do cabeçalho

Alguns campos do cabeçalho IPv4 foram retirados ou passaram a ser opcionais de forma a simplificar o tratamento de um pacote comum. Embora os endereços IPv6 sejam 4 vezes maiores aos endereços IPv4 o cabeçalho é apenas duas vezes maior.

c) Suporte para cabeçalho de extensão e de opções

As opções são codificadas, no IPv6, em cabeçalhos separados que se localizam entre o cabeçalho IPv6 e o cabeçalho de transporte. Visto que a maioria das opções apenas são examinadas e processadas por nós finais, esta codificação permite que a utilização de opções e extensões ao protocolo não interfira com a capacidade de encaminhamento de pacotes nos *router*s. Este suporte permite também que outras opções futuras possam ser incorporadas dando assim maior flexibilidade. Ao contrário do IPv4 onde o comprimento máximo da parte opcional do cabeçalho é de 40 *byte*s, o que se torna uma severa limitação à utilização de certas opções, as opções em IPv6 podem ser de comprimento arbitrário.

d) Suporte para autenticação e privacidade

O protocolo IPv6 inclui as definições de extensões que permitem a autenticação e confidencialidade de comunicações ao nível de rede.

e) Suporte de auto-configuração

A nova versão do protocolo possui mecanismos destinados a facilitar a gestão e configuração de ambientes IP através da utilização de mecanismos de auto-configuração. São definidos mecanismos de auto-configuração com manutenção de estado (dependentes de uma entidade que realiza a atribuição de endereços) e sem manutenção de estado. Esta funcionalidade é bastante útil para o estabelecimento de ligações móveis.

f) Suporte para seleção de rota pelo originador

O IPv6 inclui uma extensão que permite a especificação de rota pelo originador desenhada para se integrar com a utilização do protocolo *Source Demand Routing Protocol* (SDRP). Este protocolo tem por objetivo a seleção de rotas pelo originador de forma a completar o encaminhamento de pacotes com base na informação fornecida pelos protocolos de *routing* intra e inter-domínio correntes. Esta opção permite não só, controlar o tráfego na rede, como também aumentar a segurança na transmissão da informação.

g) Transição simples e flexível

Uma das características chave do protocolo IPv6 é um plano de transição simples que permita a instalação incremental de nós IPv6 no ambiente atual. Este plano contempla a instalação de nós IPv6 sem exigir qualquer dependência em relação a outros nós e permitindo o endereçamento de nós IPv6 com base nos endereços IPv4 já atribuídos.

h) Suporte para tráfego com garantia de qualidade de serviço

O cabeçalho IPv6 contem um campo de fluxo destinado a ser utilizado em conjunto com um protocolo de reserva de recursos, de forma a permitir a utilização de qualidade de serviço garantida.

i) Suporte para *Jumbograms*

Possibilidade de enviar pacotes com dimensão superior a 64Kb. O limite de um pacote *Jumbogram* é de 4Gb (tamanho registado nos primeiros 32 *bits* do *payload*) sendo colocado o valor 0 no campo *Payload Length* do cabeçalho, indicando assim um *Jumbogram*. Esta propriedade é útil para as redes com grande largura de banda.

Cabeçalhos de Extensão

A informação respeitante a opções é codificada, no IPv6, em cabeçalhos separados que podem ser colocados entre o cabeçalho IPv6 e o cabeçalho do protocolo de transporte.

Vários cabeçalhos de extensão podem ser encadeados dado que cada opção é identificada por um valor distinto atribuído pela IANA.

À exceção de um cabeçalho de opção denominado *hop-by-hop*, os cabeçalhos de extensão não são examinados ou processados por nenhum nó intermédiário até o pacote ser entregue a interface identificado pelo endereço de destino. O cabeçalho *hop-by-hop*, quando presente, segue imediatamente o cabeçalho IPv6.

Os cabeçalhos são processados sequencialmente pelo destinatário. Ao encontrar um tipo de cabeçalho desconhecido, este deverá descartar o pacote e enviar uma mensagem de erro ao originador por ICMP - *Internet Control Message Protocol*.

Uma implementação completa de IPv6 inclui a implementação dos seguintes cabeçalhos de extensão:

a) Opções nó-a-nó (*Hop-by-Hop Options*)
b) *Routing*
c) Fragmentação
d) Opções de Destino
e) Autenticação
f) Privacidade (*Encapsulating Security Payload*)

O comprimento de um cabeçalho de extensão é múltiplo de 8 *bytes* de forma a manter o alinhamento de 8 *bytes* para os cabeçalho subsequentes.

Segurança em IPv6

As especificações do IPv6 definiram dois mecanismos de segurança: a autenticação de cabeçalho (*authentication header*, (RFC1826)) ou autenticação IP, e a segurança do encapsulamento IP (*encrypted security payload*, (RFC1827)).

A autenticação de cabeçalho assegura ao destinatário que os dados IP são realmente do remetente indicado no endereço de origem, e que o conteúdo foi entregue sem modificações. A autenticação utiliza um algoritmo chamado MD5 (*Message Digest* 5), especificado em (RFC1828). A segurança do encapsulamento IP permite a autenticação dos dados encapsulados no pacote IP, através do algoritmo de criptografia DES (*Data Encryption Standard*) com chaves de 56 *bits*, definida em (RFC1829). Os algoritmos de autenticação e criptografia citados acima utilizam o conceito de associação de segurança entre o transmissor e o receptor. Assim, o transmissor e o receptor devem concordar com uma chave secreta e com outros parâmetros relacionados à segurança, conhecidos apenas pelos membros da associação. Para gerenciar as chaves provavelmente será utilizado o IKMP (*Internet Key Management Protocol*), desenvolvido pelo grupo de trabalho em Seguranca IP.

1.6 IP SECUTIRY PROTOCOL

O *IPsec* (*IP Security Protocol*) é um conjunto de padrões abertos que oferece um caráter confidencial aos dados, integridade de dados e autenticação entre participantes não hierárquico na camada IP. À medida que o IPsec ganhar a aceitação do mercado, os clientes exigirão suporte para ele nos produtos de interligação de redes que comprarem. Embora o IPsec seja relativamente novo, muitos fabricantes de roteadores e servidores oferecem suporte para ele. o IPsec está documentado nas RFCs 1825-1829.

O IPsec permite a um sistema selecionar protocolos e algoritmos de segurança, além de estabelecer chaves de criptografia. O protocolo IKE (*Internet Key Exchange*) fornece autenticação de pares IPsec. O IKE usa as seguintes tecnologias:

- DES – criptografa/descriptografa pacotes de dados.
- Diffie-Hellman – Estabelece uma chave de seção compartilhada e secreta.
- MD5 (Message Digest 5) – Um algoritmo de hash que autentica dados de pacotes.
- SHA (*Secure Hash Algorith*) - Um algoritmo de hash que autentica dados de pacotes.
- Recursos de criptografia do RSA – Fornece repúdio (recurso de segurança que impede a uma terceira pessoa de provar que ocorreu uma comunicação entre duas outras. Esse recurso é desejável se você não quer que sua comunicação possa ser rastreada).
- Assinatura RSA – Fornece não-repúdio (é o oposto do repúdio: uma terceira pessoa pode provar que ocorreu uma comunicação entre duas outras pessoas. O não repúdio é desejável se você quer ser capaz de rastrear suas comunicações e provar que elas aconteceram). (OPPENHIMMER 1999)

Implementando *IPSec*

O IPSec pode ser implementado nos *hosts*, nos *gateways*/roteadores ou em ambos, dependendo do nível de seguridade desejado pelos usuários. A implementação em *hosts* é utilizada quando a segurança desejada é "fim a fim". No entanto quando se deseja oferecer segurança a apenas uma parte da rede a implementação em roteadores é preferível, como é o caso das VPNs e *Intranets*.

Implementação no *Host*

A definição de *host* neste contexto está relacionada ao dispositivo onde o pacote se origina. Tem como vantagens:

- Fornece seguridade "fim a fim".
- Habilidade para implementar de todos os modos de seguridade do IPSec.
- Habilidade para manter o contexto do usuário para autenticação no estabelecimento de conexões IPSec.

Implementações utilizando o *host* podem ser efetivadas de duas maneiras:

a) Integrada com o sistema operacional – Como o IPSec é um protocolo da camada de rede, ele pode ser implementado como parte desta camada. O IPSec utiliza os serviços dessa camada para construir o cabeçalho IP. Este modelo é idêntico ao da implementação de qualquer outro protocolo da camada de redes, como o ICMP por exemplo. São muitas as vantagens obtidas quando se instala o IPSec integrado ao Sistema Operacional, dentre as quais destacamos:
   - Como o IPSec é fortemente integrado com à camada de redes, ele permite uma utilização mais eficiente de serviços de rede como a fragmentação e o uso de *Sockets;*
   - Todos os modos de implementação IPSec são suportados

b) *Bump in the Stack* – Para empresas que fornecem soluções utilizando VPNs ou Intranets, a solução integrada com o sistema operacional apresenta um sério problema. Elas têm que trabalhar com as características de cada sistema operacional, o que pode limitar a implementação de soluções avançadas. *Bump in the Stack* requer a implementação duplicada de muitos serviços da camada de rede como a fragmentação e tabela de roteadores, o que torna mas difícil o gerenciamento desses serviços. Sua principal vantagem é a capacidade de em uma única implementação fornecer uma solução completa. Normalmente, empresas que fornecem solução integrada como *Firewalls*, preferem ter seus próprios fornecedores de sistemas operacionais e nesse caso podem não ter todas as características necessárias para fornecer uma solução completa.

Implementação com *Gateways*

A utilização desse tipo de implementação oferece garantias de segurança para os pacotes transmitidos por uma parte da rede. Como exemplo podemos citar uma empresa que tenha implementado uma VPN ou Intranet e utilize a Internet e não sua rede privada para transferência de informações. A implementação do IPSec neste caso, pode oferecer segurança para os dados que trafegam na Internet.

O implementação com *gateways* tem as seguintes vantagens:

- Garantir a segurança dos pacotes que trafegam entre duas redes e utilizam a estrutura de uma rede pública como a Internet.
- Capacidade para autenticar usuários que efetuam *logon* na rede privada. Esta é uma característica que muitas empresas usam quando constróem VPNs ou Intrantes utilizando a Internet. Anteriormente isso era possível somente utilizando acesso *dial-up*.

Implementações utilizando o *host* podem ser efetivadas de duas maneiras:

- *Native Implementing* – Nesse caso o IPSec é integrado ao *Software* do *Gateway*, de maneira análoga à integração de *hosts* com o sistema operacional.
- *Bump in the Wire* – A implementação é análoga ao *bump in the stack*. Nesse caso, o IPSec é implementado em um dispositivo conectado a interface física do roteador. Esse dispositivo não processa qualquer algoritmo de roteamento, sendo utilizado somente para garantir a segurança dos pacotes.

A implementação de IPSec utilizando *gateways* tem muitas implicações na capacidade de remessa de pacotes. Espera-se que os roteadores transmitam os pacotes tão rápido quanto possível. Na verdade, hoje já temos roteadores capazes de transmitir até 30 milhões de pacotes por segundo. O IPSec não deveria afetar os pacotes que não requerem implementação de segurança. Esses pacotes deveriam ser transmitidos a taxas normais e não é isso que acontece. Muitas implementações fazem uso de algum *hardware* especial para ajudar a executar operações com chaves públicas, criptação e decriptação para obterem uma melhor performance. Outra característica a levar em consideração quando da implementação de IPSec em *gateways* é a quantidade de memória, que ainda é escassa nos roteadores, embora isso esteja mudando rapidamente. Os roteadores tem que armazenar tabelas enormes e normalmente não possuem grandes discos para utilizarem como memória virtual.

Modos de IPSec

Há quatro possíveis combinações de modos e protocolo utilizados pelo IPSec: AH em modo de transporte, AH em modo de túnel, ESP em modo de transporte, e ESP em modo de túnel. Na prática, AH em modo de túnel não é usado porque protege os mesmos dados que AH em modo de transporte.

O Modo de Transporte

Em modo de transporte, os protocolos AH e ESP protegem o cabeçalho de transporte. Neste modo, AH e ESP interceptam os pacotes que trafegam da camada de transporte para à camada de rede e têm segurança configurada.

Quando a segurança não é habilitada, pacotes da camada de transporte como TCP e UDP fluem para à camada de rede IP que acrescenta o cabeçalho de IP que é utilizado pela camada inter rede. Quando a segurança na camada de transporte é habilitada, os pacotes fluem no componente de IPSec. O componente de IPSec é implementado como parte da camada de rede (quando integrado com OS). O componente de IPSec soma o AH, ESP, ou ambos os cabeçalhos, e invoca à parte da camada de rede que soma o cabeçalho da camada de rede.

O modo de transporte de IPSec só pode ser usado quando é desejada segurança: "fim a fim".

Modo de túnel

IPSec em modo de túnel é normalmente usado quando o último destino do pacote é diferente do ponto de terminação de segurança. Usado em casos quando a segurança é provida por um dispositivo que não originou pacotes - como no caso de VPNs - ou quando o pacote precisa ter segurança garantida até um destino que é diferente do destino atual. No caso de modo de túnel, IPSec encapsula um pacote de IP com cabeçalhos de IPSec e soma um cabeçalho IP exterior.

Associações de segurança

As Associações de Segurança, ou SAs como normalmente são referenciados para em terminologia de IPSec, formam a base para o IPSec. O SAs são o contrato entre a comunicação de duas entidades. Elas determinam os protocolos de IPSec usados para garantir a seguridade dos pacotes, as chaves, e a duração para a qual as chaves são válidas. Qualquer implementação de IPSec

sempre constrói um banco de dados para um SA (SADB) isso mantém o SAs que os protocolos de IPSec usam para garantir a integridade dos pacotes.

Há outro componente na arquitetura de IPSec chamado banco de dados de política de segurança (SPD). O SPD trabalha junto com o SADB processando pacotes. A política é um componente extremamente importante de arquitetura de IPSec. A política define as características de comunicações seguras entre duas entidades. Define que protocolos deve usar em que modos e como os pacotes de IP são tratados.

Política de segurança

A política de segurança determina os serviços de segurança disponíveis para um pacote. Toda implementação de IPSec armazena a política em um banco de dados chamado SPD. O banco de dados é indexado através de seletores e contém as informações sobre os serviços de segurança oferecidos a um pacote de IP.

A política de segurança é consultada para ambos os processos, de envio e recebimento de pacotes de IP, para determinar quais os serviços se aplicam aquele pacote. Um SPD separado pode ser mantido para pacotes recebidos e os pacotes a serem remetidos podem utilizar uma política assimétrica. Isto é, utilizar uma política de segurança diferenciada para pacotes que trafegam entre dois *hosts*. Porém, o protocolo de administração de chaves sempre negocia SAs bidirectionais. A política de segurança requer uma política de gerenciamento para acrescentar, excluir, e modificar critérios estabelecidos. O SPD é armazenado no núcleo e a implementação do IPSec deveria prover uma interface para manipulá-lo.

1.7 UDP (USER DATAGRAM PROTOCOL) (RFC768)

O protocolo UDP (*User Datagram Protocol*) é um protocolo de transporte pertencente à família de protocolos TPC/IP.

O UDP fornece um serviço mínimo de transporte, em redes que usam o protocolo IP, permitindo que as aplicações tenham acesso direto aos serviços da camada de rede. O tipo de serviço fornecido pelo UDP tem as seguintes características :

- não é fiável ;
- não orientado à conexão;

O fato de ser não orientado à conexão significa que um datagrama pode ser enviado a qualquer momento, sem qualquer tipo de aviso, negociação ou preparação com o *host* destino. É esperado que o *host* de destino esteja preparado para receber e processar os datagramas.

O fato do protocolo não ser fiável significa que :

- não existe nenhuma garantia que os datagramas sejam entregues no destino;
- não existe registo dos datagramas enviados;
- os datagramas podem chegar fora de ordem;
- os datagramas podem chegar duplicados;
- não existe controle de fluxo;
- não existe controle de congestionamento da rede;

Deste modo, compete às aplicações que usam o UDP, implementar mecanismos de detecção e correção de erros, de modo a garantir que os datagramas cheguem corretamente ao destino.

Embora as características do serviço prestado pelo UDP sejam as mesmas que as da camada IP, este vem acrescentar dois serviços que à camada IP não disponibiliza :

a) a capacidade de distinguir um, dentre os vários, processos que estejam a usar os serviços da camada de rede IP, num mesmo *host* (multiplexagem/desmultiplexagem);

b) a capacidade de verificação da exatidão dos dados recebidos;

O mecanismo que permite distinguir um entre múltiplos destinos independentes dentro de um mesmo *host*, é implementado através da criação de um conjunto de pontos de destino abstratos chamados *protocol* ports.

Os *protocol* ports podem ser vistos como um ponto de ligação para conexões de rede. Se uma aplicação pretende oferecer um determinado serviço, esta associa-se a um *protocol* ports e espera que os clientes solicitem os seus serviços. Um cliente que pretenda utilizar o serviço, solicita um *protocol* port ao seu sistema operacional, e liga-se ao *protocol* port do servidor remoto. Cada port é identificado através de um número inteiro positivo único que pode ir de 0 a 64K-1. Para comunicar com uma aplicação que se encontra a correr num outro *host* é, deste modo, necessário especificar, além do número IP do *host* de destino, o número do *protocol* port que identifica a aplicação destino.

O mecanismo que permite verificar se os dados chegaram ao destino intactos, é implementado através de um *checksum* sobre o conjunto (cabeçalhos, dados) da mensagem UDP. Como o *checksum* existente no cabeçalho do protocolo IP não é calculado sobre à parte de dados do datagrama IP, o *checksum* existente no cabeçalho de uma mensagem UDP é o único meio de garantir que os dados chegaram intactos e podem ser usados.

Formato das mensagens UDP

Às mensagens UDP dá-se o nome de datagrama. Conceitualmente, um datagrama UDP é constituído por duas partes :

1) O *Header* - Constituído por quatro campos de 16-*bits* :
   a) *Udp Source* Port: usado para identificar o processo, dentro do *host* origem, que enviou a mensagem. O preenchimento deste campo é opcional, devendo nesse caso ter todos os *bits* com valor zero.
   b) *Udp Destination* Port: usado para identificar o processo, dentro do *host* destino, ao qual é dirigida a mensagem.
   c) *Udp Message Length*: contém o número total de octetos da mensagem UDP (cabeçalho + dados).
   d) *Udp Checksum*: *checksum* da mensagem UDP (cabeçalho + dados).O preenchimento deste campo é opcional, devendo nesse caso ter todos os *bits* com valor zero.

2) A Zona de Dados – Constituída pelos dados que estão sendo transmitidos.

Cálculo do checksum UDP

O *checksum* UDP é calculado levando em conta mais informação do que a existente no datagrama UDP. Para calcular o *checksum*, o *software* UDP junta um pseudo-cabeçalho ao datagrama e calcula o *checksum* do conjunto {pseudo-cabeçalho ,datagrama}. O *pseudo-header* tem a seguinte constituição :

   a) *Source IP address* : endereço IP do *host* origem, usado no envio do datagrama.
   b) *Destination IP Address* : endereço IP do *host* destino, ao qual se destina o datagrama.

c) *Zero :* um octeto com todos os *bits* com zero, usado para que o comprimento total do pseudo-cabeçalho seja um múltiplo exato de 16*bits*.

d) *Proto :* contem o código do protocolo, UDP=17.

e) *Udp Length :* comprimento total do datagrama UDP em octetos (mesmo valor que o campo UDP MESSAGE LENGTH do cabeçalho UDP). Nota: não inclui o pseudo-cabeçalho.

O uso do pseudo-cabeçalho no cálculo do *checksum*, permite verificar se o datagrama UDP chegou ao destino correto, isto é, na aplicação certa do *host* certo. É necessário o par (endereço IP, *protocol* port) para identificar completamente uma aplicação existente num determinado *host*. No destino, o *software* UDP calcula o *checksum* do mesmo modo, usando o endereço IP de destino contido no datagrama IP. Se o valor calculado coincidir com o valor contido no cabeçalho UDP, então o datagrama deve ter chegado ao destino pretendido, bem como ao *protocol* port correto.

No cálculo do *checksum*, o *software* coloca os *bits* do campo *checksum* a zero e só depois calcula o *checksum* do conjunto {pseudo-cabeçalho, datagrama}. O cálculo é efetuado exatamente do mesmo modo que no caso do *checksum* IP : o conjunto {pseudo-cabeçalho, datagrama} é dividido em elementos de 16-*bits*, e depois é calculado o complemento-para-um da soma, em complemento-para-um, dos vários elementos de 16-*bits*.

É necessário notar que o pseudo-cabeçalho não é transmitido. Apenas utilizado para o cálculo do valor do *checksum*.

Como vimos, o cálculo do *checksum* UDP é opcional, devendo o campo *checksum* do *header* ter o valor zero, caso não se pretenda usar esse serviço. O fato do cálculo poder resultar num valor nulo não constitui problema, uma vez que o zero tem duas representações em complemento-para-um, todos os *bits* a zero ou todos os *bits* a um. Assim, quando o cálculo resultar num valor nulo, usa-se a representação do zero com todos os *bits* a um, eliminando assim qualquer tipo de ambigüidade.

Encapsulamento da mensagem UDP

Tendo em conta o modelo conceitual de estratificação de protocolos, o protocolo UDP encontra-se situado numa camada superior à do protocolo IP. Deste modo, as mensagens UDP são encapsuladas num datagrama IP para serem enviadas através da rede, e os datagramas IP são, por sua vez, encapsulados numa *frame*, a qual depende do tipo de rede física a que o *host* destino se encontra ligado. Podemos ver que o UDP respeita o princípio da estratificação, um datagrama UDP recebido da camada IP no *host* destino, é idêntico ao datagrama que o UDP passou para à camada IP no *host* origem. Do mesmo modo, os dados que o UDP entrega na aplicação destino são exatamente iguais aos dados que a aplicação na origem enviou para o UDP. Temos assim, uma divisão de tarefas entre as diferentes camadas de protocolos: à camada IP é responsável apenas pela transferência dos dados entre dois *hosts*, enquanto à camada UDP é responsável pela diferenciação das múltiplas aplicações origem ou destino, dentro de um mesmo *host*.

No entanto, o fato de à camada UDP necessitar do endereço IP do seu *host* quando envia uma mensagem, e do endereço IP de origem quando processa uma mensagem recebida, para o cálculo do *checksum*, constitui uma violação ao principio da estratificação.

Multiplexagem, Desmultiplexagem e portas UDP

O UDP aceita datagramas de várias aplicações e envia-os, num único fluxo de informação, para à camada IP de transmissão. A esta ação dá-se o nome de multiplexagem.

O UDP aceita datagramas vindos da camada IP e passa-os para a aplicação apropriada. A esta ação dá-se o nome de desmultiplexagem.

Cada aplicação antes de poder enviar um datagrama UDP, tem de negociar com o sistema operacional para obter um *protocol port*. Assim que *protocol port*

tenha sido atribuído, qualquer datagrama que a aplicação envie através desse port especificado no campo UDP SOURCE PORT.

Na recepção de datagramas vindos da camada IP, o UDP faz a desmultiplexagem em termos do *protocol port* destino .

Os *protocol ports* são implementados através de filas internas e, à medida que as mensagens vão chegando o sistema operacional vai colocando-as na fila correspondente ao DESTINATION PORT indicado no cabeçalho da mensagem. Se a fila estiver cheia, ocorre um erro e o UDP descarta o datagrama.

Quando o UDP recebe um datagrama, verifica se o port destino condiz com um dos ports correntemente em uso. Se a verificação resultar verdadeira, o UDP coloca a mensagem na fila correspondente. Caso contrário, o UDP envia uma mensagem de erro ICMP do tipo port *unreachable* e descarta o pacote.

Ports reservados e Ports disponíveis

Antes de dois *hosts* se comunicarem, devem saber o numero de port com o qual querem comunicar. Para atribuição de *port*, um dos 2 métodos abaixo é utilizado:

- *Central Authority* : Há uma autoridade que estabelece o numero dos ports. Estes números são designados *Well-Known port assigments*. Este método também é conhecido por *Universal Assigments*.
- *Dynamic binding* : Sempre que um programa necessitar de um port, o sistema operacional do *host* com o qual quer comunicar lhe atribui um.

No caso do UDP optou-se por uma solução híbrida em que alguns ports são atribuídos à prior, deixando a maior parte disponível para uso geral.

Em ambiente exclusivamente *Windows* por exemplo temos:

Port	Keyword	Descrição
137	NETBIOS-NS	NetBIOS Name Service
138	NETBIOS-DGM	NetBIOS DataGrams
139	NETBIOS-SSN	NetBIOS Session

QUADRO 3. Portas TCP/UDP no Ambiente Windows

Aplicações que utilizam UDP

O UDP é usado nas seguintes situações :

- Quando as aplicações não necessitam do nível de serviço que o TCP fornece; o uso do UDP diminui a complexidade no processamento dos datagramas;
- Quando os dados de uma aplicação não são essenciais, como por exemplo um *query DNS*; se a resposta não chegou volta-se a perguntar;
- Quando a qualidade e a eficiência da rede é elevada;

Exemplo de aplicações que usam UDP :

- RIP (*Routing Information* Protocol)
- DNS (*Domain Name* Service)
- TFTP (*Trivial File Transfer Protocol*)
- NFS (*Network File System*)
- SNMP (*Simple Network Management* Protocol)
- NetBIOS

1.8 TELNET (RFC 416 E RFC 764)

O TCP/IP inclui um protocolo de aplicação padrão para *login* remoto conhecido como Telnet. O protocolo Telnet permite que usuário estabeleça uma sessão interativa com outra máquina da rede. Uma vez estabelecida a sessão, a máquina do usuário passa a atuar como cliente e máquina contactada como servidor. O módulo de cliente Telnet estabelece uma conexão TCP utilizando o nome ou o endereço IP da máquina de destino, captura todas as teclas digitadas e as envia ao servidor através dessa conexão.

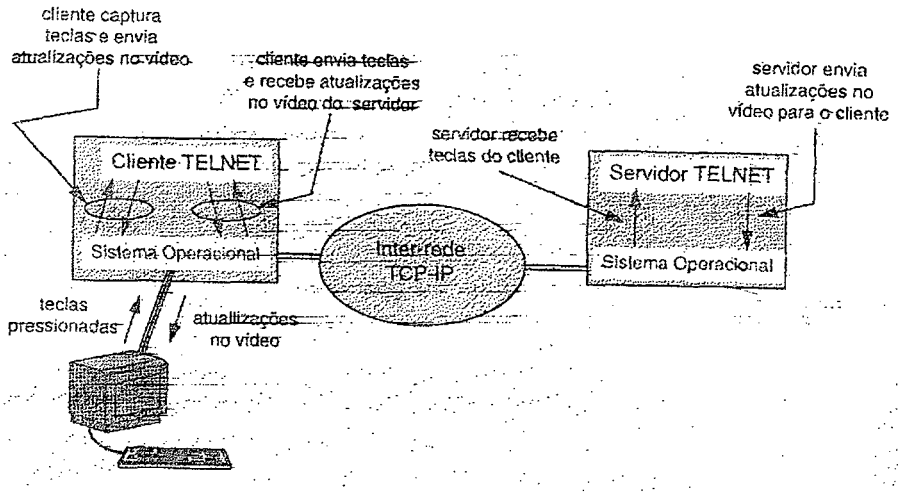

FIGURA 5. Conexão Telnet
Fonte: (SOARES, LOPES, COLCHER, 1995)

O protocolo Telnet define o formato de dados que uma aplicação tem que enviar para uma máquina remota para logar-se naquele sistema, bem como o formato das mensagens que a máquina remota deve retornar. (COMER, STEVENS, v. III, 1994)

Após estabelecida a conexão no entanto, o usuário atuando como cliente deve informar seu *login* e senha no servidor para que possa acessar os serviços, ou seja, o usuário tem que ter uma conta na máquina remota, para poder continuar com a sessão aberta.

O servidor processa as teclas digitadas e encaminha para o cliente os caracteres que devem ser mostrados no vídeo do terminal.

O protocolo Telnet foi construído tendo como base as seguintes idéias: o conceito de terminal virtual de rede, o princípio da negociação de opções e o tratamento equivalente de terminais e processos.

Quando uma conexão é estabelecida, assume-se que ambas as extremidades estão ligadas a um terminal virtual de rede, que é um dispositivo lógico que fornece uma representação padronizada de um terminal, e elimina a necessidade dos participantes conhecerem um os detalhes da máquina do outro. Cliente e servidor mapeam as características de seus terminais locais para as do terminal virtual e assumem que seu parceiro está fazendo o mesmo.

Uma vez conectados os participantes poderão também negociar as opções de comportamento do terminal virtual, tais como, o formato de representação dos caracteres utilizados e o modo de operação (*half-duplex* ou *full duplex*), que corresponde ao princípio da negociação de parâmetros. (SOARES, LEMOS, COLCHER, 1995)

O princípio do tratamento equivalente de terminais e processos faz com que o protocolo Telnet trate os terminais e os processos remotos de forma simétrica, ou seja, o cliente não precisa necessariamente ser um terminal, podendo ser um processo de aplicação qualquer.

Um fato a ser considerado é que dada à simetria da conexão, ambas as extremidades podem tomar a iniciativa da negociação de opções, o que não seria compatível com a arquitetura cliente/servidor, onde normalmente o servidor é passivo, atuando somente em respostas às solicitações feitas pelo cliente.

Autenticação em Telnet

A adaptação do Telnet para implementar métodos de autenticação foi feita através da inclusão de uma opção chamada *Authentication Option* e alguns comandos para negociar o tipo de autenticação, como o método e o direcionamento da operação. A negociação é efetuada através do uso das seguintes mensagens:

IAC WILL AUTHENTICATION

Este comando é enviado pelo cliente indicando que ele deseja utilizar uma técnica de autenticação segura.

IAC DO AUTHENTICATION

Este comando é enviado pelo servidor indicando que ele deseja que o cliente utilize uma técnica de autenticação segura.

IAC WONT AUTHENTICATION

É uma resposta negativa do cliente a um DO AUTHENTICATION indicando que ele não entende esta opção.

IAC DONT AUTHENTICATION

É uma resposta negativa do servidor a um WILL AUTHENTICATION indicando que ele não entende esta opção.

Por restrição de projeto, a negociação só pode ser feita do servidor para o cliente. O servidor deve mandar um *DO* e o cliente um *WILL* ou *WONT*. Em qualquer caso, se o servidor receber um *DO*, ele deve responder com um *WONT*, e se o cliente receber um *WILL*, ele deve responder com um *DONT*.

Após estabelecido que os dois lados entendem a opção, os seguintes comandos (ou *sub-options*) são usados para caracterizar o tipo de autenticação e o direcionamento: IAC SB AUTHENTICATION SEND *authentication-type-pair-list IAC SE*. Somente o servidor pode enviar este comando. Neste comando o servidor envia uma lista dos tipos de autenticação suportados por ele para o cliente, em ordem de preferência, o primeiro tem a maior preferencia e o último, a menor. Esta lista está em *authentication-type-pair-list*.

O grande problema de segurança do Telnet é que para a identificação do usuário na máquina remota, trafegam pela rede o *username* e a senha em claro, sem qualquer método de criptografia, podendo no entanto ser utilizados algum dos métodos de autenticação citados a seguir:

- KERBEROS_V4
- KERBEROS_V5
- SPX
- RSA
- LOKI

É importante notar que o esquema de opções do Telnet permitem manter a compatibilidade entre sistemas que suportam autenticação e outros mais antigos que não suportam, pois estes últimos simplesmente recusam a negociação de qualquer opção que desconheçam, mandando um *DONT* ou *WONT* para o outro lado da conexão. O lado que recebe a negativa passa a trabalhar do modo original, sem a opção a qual tentou negociar, e sem a geração de qualquer tipo de erro.

1.9 FTP (FILE TRANSFER PROTOCOL)

O FTP é um protocolo que permite que um usuário em computador cliente transfira, renomeie ou remova arquivos remotos ou ainda crie, remova e modifique diretórios remotos. O FTP permite a transferência somente de arquivos completos. Antes que possa efetuar qualquer operação o usuário precisa logar-se no servidor FTP, informando seu *login* e senha.

O FTP não se preocupa em definir um sistema de arquivos virtual e sim em definir uma interface com os sistemas de arquivos nativos.

A operação do FTP baseia-se em duas conexões entre o cliente (quem solicita a conexão) e o servidor (computador onde estão localizados os arquivo desejados): uma conexão denominada de controle (permanece aberta enquanto durar a conexão FTP) é utilizada para a transferência de comandos e outra denominada conexão de transferência de dados, é utilizada para a transferência dos dados entre o cliente e o servidor. (SOARES, LEMOS, COLCHER, 1995)

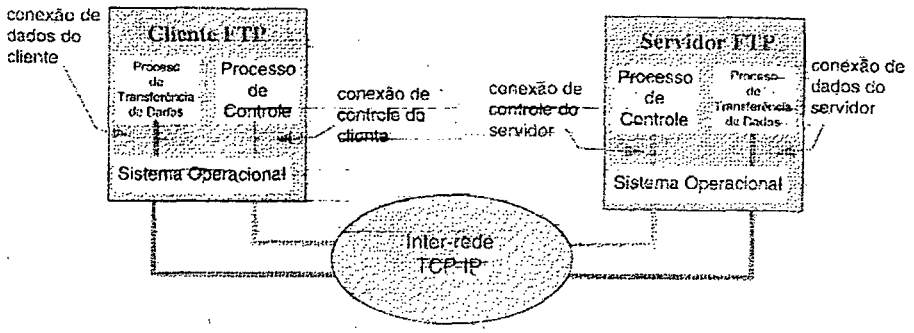

FIGURA 6. Conexões FTP de controle e de transferência de dados.
Fonte: (SOARES, LEMOS, COLCHER, 1995)

A conexão executada no cliente (chamada de Cliente-FTP) pode ser dividida em três módulos que interagem por algum mecanismo interno. Esses módulos são:

1. Interface do Usuário;
2. Interpretador de Protocolo do Cliente (Cliente-PI);
3. Processo de Transferência de dados (Cliente-DTP).

A conexão executada no servidor (chamada de Servidor-FTP) é dividida em dois módulos com funções análogas aos seus equivalentes no cliente. Esses módulos são:

1. Servidor-PI;
2. Servidor-DTP.

A conexão de controle, usada na transferência de comandos FTP e suas respostas, é realizada diretamente entre o Cliente-PI e o Servidor-PI, e a conexão de dados é estabelecida entre o Cliente-DTP e o Servidor- DTP

Durante uma sessão FTP, podem ser transferidos vários arquivos, cada um por uma conexão de transferência de dados aberta especificamente para tal.

Cada sistema de arquivo tem suas maneiras de armazenar e acessar seus arquivos; definir regras de proteção e de manipulação de maneira diversificada. Dessa forma, torna-se difícil encontrar uma definição comum a todos os sistemas de arquivos existentes.

É interessante que o FTP saiba como os dados estão armazenados nos arquivos - como eles estão representados, para permitir uma melhor adequação dos mesmos no momento de sua transferência.

O FTP, na sua implementação mais completa, suporta quatro (4) tipos diferentes de dados:

- **ASCII** – para transferência de arquivos que contém somente texto.
- **EBCDIC** - para facilitar a transferência de dados entre dois sistemas que utilizam o código EBCDIC como código nativo, o código EBCDIC é representado por caracteres de 8*bits*. o código de caracteres é a única diferença entre este código e o ASCII. mesmo não sendo comum organizar arquivos em linha em EBCDIC, pode-se utilizar o caracter <nl> para indicar o fim de uma linha.
- **Imagem** - esse tipo de dados é considerado como um conjunto contínuo de *bits* (agrupados em pacotes de 8 *bits* para efeitos de transferência). Se os dois sistemas de arquivos são do mesmo tipo, deve ser garantido que a cópia do arquivo seja idêntica ao original. isso faz com que esse tipo de dados seja o ideal para transferência de programas executáveis, bibliotecas e outros arquivos que tenham sido compilados. Este tipo de dados foi criado para facilitar a transferência de dados binários.
- **Local** - o FTP transfere os dados do tipo local em *bytes* lógicos com tamanho especificado por um segundo parâmetro (agrupados em pacotes de 8 *bits* para efeitos de transferência). O valor deste parâmetro é um número inteiro decimal (não é definido um valor pré-estabelecido). O

destino da transferência possui uma definição interna de qual é o seu tamanho lógico de caracter e faz a conversão para o melhor valor possível de armazenamento. Esse tipo foi criado para facilitar a transferência de dados binários que dependem do tamanho da unidade de dados - como inteiros binários ou reais de ponto flutuante.

Para transferir dados deve existir uma conexão de dados entre portas apropriadas e deve ser feita uma escolha de parâmetros de transferência. Os processos Cliente-DTP e Servidor-DTP possuem portas com valores *default* que devem ser suportadas por todas as versões de FTP. Entretanto, o cliente pode alterar o valor de tais portas.

Logo que inicia a transferência de dados, o gerenciamento da conexão de transferência de dados passa a ser responsabilidade do servidor; salvo uma transferência sem erros e em que os dados estão indo do cliente para o servidor. Nesse caso, em vez de enviar um *end of file*, torna-se responsabilidade do cliente fechar a conexão para indicar o fim de arquivo.

O FTP não se preocupa com a perda ou a adulteração de *bits* durante a transferência, pois é atribuição do TCP - protocolo do nível de transporte, que provê mecanismos para um eventual reinicio da transferência quando ela for interrompida por problemas externos ao sistema (como uma falha na alimentação elétrica).

Este procedimento de reinicio só está disponível nos modos de transferência que permitem inserir controles no meio do fluxo de dados

O grande problema deste serviço é que para a identificação do usuário na máquina remota, trafegam pela rede o *username* e a senha em claro, sem qualquer método de criptografia.

O FTP pode ser configurado para trabalhar também com acesso anônimo, ou seja, qualquer pessoa, mesmo que não tenha conta naquela máquina pode depositar e copiar arquivos dela. O FTP anônimo é muito utilizado para a distribuição de documentos e *softwares* através da rede. Deve ser tomado muito cuidado na configuração de um servidor de FTP anônimo para que o acesso seja

restrito aos arquivos determinados incluindo restrições de acesso. Outra atenção é que o servidor de FTP não fique sendo depósito de documentos indesejados como arquivos de intrusos etc, por este motivo é aconselhável criar um diretório separado para a colocação de arquivos com espaço limitado.

A arquitetura TCP/IP define adicionalmente um outro protocolo que fornece um serviço simplificado de transferência de arquivos, o TFTP (*Trivial File Transfer Protocol*), que restringe sua operação simplesmente a transferência de arquivos, não implementando mecanismos de autenticação e operando em uma única conexão. O TFTP utiliza o UDP para o transporte de blocos de dados de tamanho fixo. Como o serviço fornecido pelo UDP não garante a entrega dos blocos ao destinatário, o TFTP utilizado o protocolo de *bit* alternado para transmitir seus blocos.

1.10 O PROTOCOLO SNMP

O protocolo *Simple Network Management Protocol* é a solução adotada na Internet para permitir que gerentes de redes possam localizar e corrigir problemas. Geralmente, é utilizado um processo na máquina do administrador chamado de cliente (uma *workstation* ou um *gateway*, por exemplo) que se conecta a um ou mais servidores SNMP localizados em máquinas remotas, para executar operações sobre os objetos gerenciados (por exemplo, para obter informações sobre estes objetos). O SNMP utiliza o protocolo UDP na comunicação entre cliente e servidor. Para o cliente da rede, o SNMP executa as operações sobre os objetos de forma transparente, o que permite a interface do *software* de gerenciamento da rede criar comandos imperativos para executar operações sobre os objetos gerenciados. Esta é à grande diferença entre gerenciar uma rede usando o protocolo SNMP e gerenciar a mesma rede usando outros protocolos.

No protocolo SNMP são definidas tanto a sintaxe (forma e a representação dos nomes e do valores) como o significado das mensagens trocadas entre os clientes e os servidores. O SNMP também define as relações administrativas entre os vários *gateways* que estão sendo gerenciados, determinando a autenticação necessária para os clientes acessarem os objetos gerenciados.

Ao contrário dos outros protocolos de gerenciamento que apresentam muitos comandos (operações), o SNMP apresenta somente um conjunto limitado de comandos, baseado num simples mecanismo de busca/alteração. Portanto, é muito mais simplés de ser implementado do que um protocolo com muitas operações, em que cada operação sobre um objeto necessita de um comando diferente para implementá-la.

O cliente SNMP cria em memória um banco de dados de informações denominado MIB (Management Information Base), que retém todas as estatísticas que poder ser consultadas a partir do servidor SNMP. Nessas estatísticas podem ser encontradas, por exemplo, informações como a quantidade de pacotes IP, TCP e UDP que o computador enviou pela placa de comunicação com sucesso ou falha. Quando o equipamento gerenciado é um roteador TCP/IP é possível configurar, habilitar e desabilitar o roteamento de pacotes.

O mecanismo de **busca/alteração** conceitualmente só apresenta duas operações: uma que permite ao cliente alterar atributos de um objeto (**SET**), e outra para obter os valores dos atributos de um objeto (**GET**). Somente estão disponíveis estas operações (e suas variações) para o gerenciamento da rede, que serão aplicadas sobre os objetos.

Podemos, resumidamente, dizer que os principais objetivos do protocolo SNMP, devido ao protocolo desejar ser flexível e simples, são:

- Reduzir o custo da construção de um agente que suporte o protocolo;
- Reduzir o tráfego de mensagens de gerenciamento pela rede necessárias para gerenciar os recursos da rede;

- Reduzir o número de restrições impostas as ferramentas de gerenciamento da rede, devido ao uso de operações complexas e pouco flexíveis;
- Apresentar operações simples de serem entendidas, sendo facilmente usadas pelos desenvolvedores de ferramentas de gerenciamento;
- Permitir facilmente a introdução de novas características e novos objetos não previstos ao se definir o protocolo;
- Construir uma arquitetura que seja independente de detalhes relevantes à somente a algumas implementações particulares.

O SNMP tem como base a técnica "*fetch-store*", ou seja, todas as suas operações previstas são derivadas de operações básicas de busca e armazenamento. Estas operações são:

Operação	Função
get-request	Leitura de uma variável
get-next-request	Leitura da próxima variável
get response	Resposta a uma operação de leitura
set request	Gravação de um campo variável
trap	Notificação da ocorrência de um evento

QUADRO 4. Operações básicas de busca e armazenamento em SNMP

Um gerente interage com um agente de acordo com as regras estabelecidas pelo *framework* de gerenciamento. Em geral, o gerenciamento da rede impõe *overheads* significativos, pois cada nó apenas produz algumas variáveis que serão lidas e usadas para sua monitoração.

Duas operações básicas no protocolo SNMP, são:

1. A **operação SET** é usada por um cliente para alterar um ou mais atributos de um objeto gerenciado (*set-request*);
2. A **operação GET** é usada por um cliente para obter o(s) valor(es) de um ou mais atributos de um objeto gerenciado (*get-request* para o pedido e *get-response* para obter o retorno deste pedido).

Uma operação GET ou SET somente se refere a uma **única instância** de um objeto representada através de seu nome. No protocolo SNMP, as operações são **atômicas**, isto é, todas as operações de um pedido devem ser executadas. Não existem execuções parciais de um pedido (no caso, operações aplicadas a múltiplos objetos). Se ocorrer algum erro durante a execução de uma operação, os resultados produzidos por esta operação devem ser ignorados.

Antes de executar um pedido, o servidor deve mapear apropriadamente os nomes dos objetos codificados nos objetos internos que armazenam as características das entidades da rede (através dos atributos do objeto).

Além das operações padrões, existem mais outras duas operações:

- Numa **operação GET-NEXT** o nome do objeto não só especifica o objeto a acessar (para obter seus atributos, como na operação GET normal), como também é usado para descobrir qual o próximo objeto na sequência léxica. Como retorno, a operação informa o nome do próximo objeto na hierarquia, e os valores dos seus atributos (obtidos através da execução de uma operação GET normal sobre o objeto).
- Uma **TRAP** que é usada para informar a ocorrência de eventos, permitindo aos servidores SNMP enviarem informações aos clientes sempre que ocorrer algum evento que sinalize a ocorrência de alterações nos objetos (no protocolo, foram definidas somente algumas *traps*).

A operação GET-NEXT é útil para obter os atributos dos objetos de uma tabela de tamanho desconhecido, pois um cliente pode enviar continuamente requisições GET-NEXT a um servidor que se encarregará de enviar os atributos do objeto e o nome do próximo objeto. Cada novo pedido deve especificar o nome do objeto retornado pelo pedido anterior, o que permite varrermos a tabela sem saber qual o próximo objeto desta tabela. Este processo é chamado de caminhamento na tabela. A implementação de uma estrutura de dados que suporte o comando GET-NEXT pode ser complicada devido a esta operação poder pular o próximo objeto simples (na ordem lexicográfica) devido a existência de objetos vazios. Como consequência, não pode-se usar simplesmente a ordem lexicográfica presente na árvore para determinar quais objetos satisfazem a um comando GET-NEXT, devendo também existir um programa que examine os objetos, pule aqueles objetos que estejam vazios e descubra o primeiro objeto simples pertencente a um objeto não vazio.

As entradas em uma tabela apontam para outras tabelas que não contém o identificador completo do objeto, mas somente o prefixo deste identificador, porque o identificador completo do objeto para um item da tabela é formado pelo prefixo que identifica a tabela, mais um sulfixo que identifica uma entrada particular na tabela em que o objeto está armazenado.

Mensagens no protocolo SNMP

Ao contrário de muitos outros protocolos TCP/IP, as mensagens no protocolo SNMP não apresentam campos fixos (tanto a mensagem de pedido, como a de resposta), o que dificulta o entendimento e a decodificação das mensagens.

As partes mais importantes de uma mensagem são: as operações (GET, SET e GET-NEXT) e a identificação, dos objetos em que as operações devem ser aplicadas.

Deve existir um cabeçalho que informe o tamanho da mensagem, que só será conhecido após a representação de cada campo ter sido computada. Na verdade, o tamanho da mensagem depende do tamanho de sua parte remanescente

(que contém os dados), portanto o tamanho só poderá ser computado após a construção da mensagem. Uma maneira de evitar este problema é construir a mensagem de trás para frente.

Uma mensagem SNMP deve definir o servidor do qual obtemos ou alteramos os atributos dos objetos, e que será responsável por converter as operações requisitadas em operações sobre as estruturas de dados locais. Após verificar os campos de uma mensagem, o servidor deve usar as estruturas internas disponíveis para interpretar a mensagem e enviar a resposta da operação ao cliente que requisitou o pedido.

Servidores e Clientes SNMP

Um servidor SNMP deve ser capaz de aceitar pedidos de operações sobre os objetos gerenciados, executá-los e retornar o resultado das operações após sua execução.

Numa operação de busca, as informações sobre o(s) objetos(s) são retornados na mensagem SNMP de resposta ao pedido, que depois de ser convertida para o formato de uma mensagem SNMP, será enviada ao cliente que solicitou a operação. Um servidor SNMP deve ter um eficiente mapeamento de nomes, pois quando um nome de um objeto chegar ao servidor num pedido, o servidor deverá ser capaz de reconhecer o nome, para chamar o procedimento correto para executar a operação solicitada no pedido.

Ao invés de manter todas as informações necessárias para atender ao pedido, podemos chamar um procedimento que irá mapear o nome do objeto para a sua representação interna correspondente. A maioria destes procedimentos são rápidos e diretos, pois simplesmente convertem o formato de uma mensagem SNMP para a representação interna, mas se não existir uma representação para algum objeto no servidor, os pedidos que executem operações sobre este objeto irão requerer mais processamento por parte do servidor, e não somente o processamento

necessário para uma simples tradução de um nome de um objeto para a estrutura de dados local usada para armazenar os dados.

Após a conversão dos campos da mensagem para a forma interna usada pelo servidor, o pedido será armazenado numa estrutura descritora que contem um ponteiro para uma lista ligada com os nomes de todos os objetos sobre os quais a operação deve ser aplicada. Após serem geradas as respostas, estas devem ser convertidas para que possam ser adicionadas na mensagem de resposta, que será enviada ao cliente que solicitou o pedido. As operações presentes nos pedidos geralmente são executadas por funções no processo servidor.

Um cliente SNMP deve construir e enviar o seu pedido ao servidor, esperar pela resposta de seu pedido, e verificar se a resposta concorda com a resposta do que foi pedido. Devido ao protocolo UDP não garantir a entrega dos pacotes, o cliente deve implementar estratégias para *time out* e retransmissão das mensagens que contém os pedidos.

Um cliente só pode obter ou alterar os atributos de um objeto gerenciado somente se tiver permissão para acessar o objeto. Esta permissão é definida através de uma política de acesso. Esta política usa o mecanismo de comunidades (*community*), em que definimos para cada comunidade, um grupo de objetos e de operações que podem ser realizadas sobre estes objetos. As comunidades são classificadas como *Monitor, Control* e *Trap*. Em cada comunidade é possível declarar uma senha igual ou diferente. A classe comunidade *Monitor* declara que o Servidor SNMP poderá consultar informações da MIB, a comunidade *Control* indica que o Servidor SNMP poderá alterar alguns campos da MIB do cliente e classe de comunidade *Trap* permite que o servidor SNMP receba mensagem de ocorrências do TCP/IP detectadas nos clientes SNMP.

Se um cliente não pertencer a comunidade autorizada para acessar o objeto, ou se não tiver autoridade para executar a operação sobre o objeto presente em seu pedido, o pedido será recusado, e será retornada uma mensagem de erro ao cliente, informando que ele não tem direito de acesso ao objeto, ou que ele não pode executar a operação pedida sobre os atributos do objeto. Este mecanismo permite a

definição de relações administrativas entre os servidores e os clientes SNMP de uma rede.

Muitas das recomendações para proteção de conexões da Internet também se aplicam à proteção de redes corporativas internas. Os serviços internos de rede podem fazer uso da autenticação e da autorização, de filtros de pacotes, *logs* de auditoria, segurança física, criptografia e assim por diante. Logo, a limitação do SNMP deve ser considerada em redes corporativas para as quais as metas de segurança superam a facilidade de gerenciamento. Um dos assuntos principais com SNMP é a operação SET que permite à uma estação remota mudar dados de gerenciamento e configuração. Uma nova versão do SNMP (SNMPv3) está em desenvolvimento e oferecerá suporte à autenticação para uso com a operação SET e outras operações no SNMP. Para clientes com numerosos roteadores e *switches*, um protocolo como o TACACS (*Terminal Controler Access Control System*) pode ser utilizado para administrar grandes números de IDs de usuários e senhas de roteadores e *switches* em um banco de dados centralizado. O TACACS também oferece características de auditoria. (OPPENHEIMER, 1999)

1.11 SMTP

O SMTP (*Simple Mail Transfer Protocol*) é um protocolo TCP/IP que geralmente opera na porta 25, usado para enviar e receber *e-mails*. É sempre usado em conjunto com um dos dois protocolos, POP3 e IMAP, que permitem ao usuário salvar mensagens na caixa de correio em um servidor de mensagens, bem como baixá-las periodicamente. É o protocolo usado no sistema de correio eletrônico na arquitetura Internet TCP/IP.

Um usuário, ao desejar enviar uma mensagem, utiliza o módulo interface com o usuário para compor a mensagem e solicita ao sistema de correio eletrônico que a entregue ao destinatário. Quando recebe a mensagem do usuário, o sistema de correio eletrônico armazena uma copia da mensagem em seu *spool* (área do

dispositivo de armazenamento), junto com o horário do armazenamento e a identificação do remetente e do destinatário. A transferência da mensagem e executada por um processo em *background*, permitindo que o usuário remetente, após entregar a mensagem ao sistema de correio eletrônico, possa executar outras aplicações. O processo de transferência de mensagens, executando em *background*, mapea o nome da máquina de destino em seu endereço IP, e tenta estabelecer uma conexão TCP com o servidor de correio eletrônico da máquina de destino. Note que o processo de transferência atua como cliente do servidor do correio eletrônico. Se a conexão for estabelecida, o cliente envia uma cópia da mensagem para o servidor, que a armazena em seu *spool*. Caso a mensagem seja transferida com sucesso, o servidor avisa ao cliente que recebeu e armazenou uma cópia da mensagem. Quando recebe a confirmação do recebimento e armazenamento, o cliente retira a cópia da mensagem que mantinha em seu *spool* local. Se a mensagem, por algum motivo, não for transmitida com sucesso, o cliente anota o horário da tentativa e suspende sua execução. Periodicamente o cliente acorda e verifica se existem mensagens a serem enviadas na área de *spool* e tenta transmiti-las.

Se uma mensagem não for enviada por um período, por exemplo de dois dias, o serviço de correio eletrônico devolve a mensagem ao remetente, informando que não conseguiu transmití-la. Em geral, quando um usuário se conecta ao sistema de correio eletrônico ativa a verificação de mensagens na caixa postal do usuário. Se existirem, o sistema de correio eletrônico emite um aviso para o usuário que, quando achar conveniente, ativa o módulo de interface para receber as correspondências.

Uma mensagem SMTP divide-se em duas partes: cabeçalho e corpo, separados por uma linha em branco. No cabeçalho são especificadas as informações necessárias para a transferência da mensagem. O cabeçalho é composto por linhas, que possuem uma palavra-chave seguida de um valor. Por exemplo, identificação do remetente (palavra-chave "para:" seguida do seu endereço), identificação do destinatário, assunto da mensagem, etc... No corpo são transportadas as informações da mensagem propriamente dita. O formato do texto é livre e as mensagens são

transferidas no formato texto. Os usuários do sistema de correio eletrônico são localizados através de um par de identificadores. Um deles especifica o nome da máquina de destino e o outro identifica a caixa postal do usuário. Um remetente pode enviar simultaneamente várias cópias de uma mensagem, para diferentes destinatários utilizando o conceito de lista de distribuição (um nome que identifica um grupo de usuários). O formato dos endereços SMTP é o seguinte: nome_local@nome_do_domínio onde o nome_do_domínio identifica o domínio ao qual a máquina de destino pertence (esse endereço deve identificar um grupo de máquinas gerenciado por um servidor de correio eletrônico). O nome_local identifica a caixa postal do destinatário. O SMTP especifica como o sistema de correio eletrônico transfere mensagens de uma máquina para outra. O módulo interface com usuário e a forma como as mensagens são armazenadas não são definidos pelo SMTP. O sistema de correio eletrônico pode também ser utilizado por processos de aplicação para transmitir mensagens contendo textos.

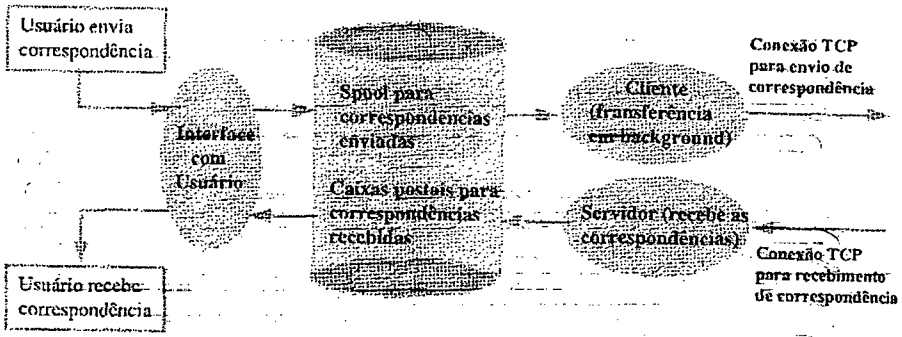

FIGURA 7. Arquitetura do Correio Eletrônico SMTP
Fonte: (SOARES, LOPES, COLCHER, 1995)

1.12 HTTP (HYPERTEXT TRANSFER PROTOCOL) (RFC1945)

O HTTP é um protocolo da camada de aplicação da arquitetura Internet com a leveza e velocidade necessárias para um sistema de informação, baseado na hipermídia, colaborativo e distribuído.

Trabalha principalmente com documentos HTML, mas, na verdade, está aberto para suportar um ilimitado e extensível conjunto de formatos.

O protocolo HTTP é também usado como um protocolo genérico para comunicação entre programas clientes e *"proxies/gateways"* para outros protocolos da camada de aplicação da arquitetura Internet, como SMTP, NNTP, FTP, Gopher, e WAIS, permitindo acesso a recursos disponíveis de diversas aplicações.

Modelo Cliente Servidor

A principal idéia por trás da arquitetura cliente-servidor é estruturar o sistema como um conjunto de processos cooperativos (servidores) que oferecem serviços aos processos de usuários (clientes). No modelo cliente-servidor a comunicação entre processos se dá, exclusivamente, através da troca de mensagens. Isto torna o modelo adequado à implementação de sistemas distribuídos, uma vez que os mecanismos de comunicação podem ser desenhados para permitir a troca de mensagens entre processos executados em locais distintos.

Geralmente, o modelo cliente-servidor faz uso de protocolos de comunicação simples do tipo requisição/resposta. A fim de obter um serviço, um cliente envia uma requisição ao servidor. Este, por sua vez, executa as operações associadas ao serviço e envia uma resposta ao cliente, contendo dados ou um código de erro caso o serviço não possa ser executado.

Esquema de endereçamento

O protocolo HTTP usa o conceito de referência providenciado por uma URI (*Universal Resource Identifier*). As URIs são conhecidas por vários nomes: endereço WWW, Identificador Universal de Documentos, e principalmente, URL (*Uniform Resource Locator*). Uma URI é simplesmente uma string que identifica um recurso disponível na rede.

Formato de uma URL HTTP

http://*host* < : port > <caminho> <arquivo>

onde:

- ***host*** é o nome da máquina registrada em um domínio internet, ou seu endereço IP no formato decimal pontuado;
- **port** é o número da porta TCP, se este número não é informado é assumido o número 80;
- **caminho/arquivo** é o diretório onde o arquivo se encontra. Caso não sejam fornecidos, o servidor HTTP usa o valor padrão definido na implementação.
- **Protocolo não orientado a conexão (*connectionless*)**

Embora o protocolo HTTP faça uso do protocolo TCP, da camada de transporte, que é um protocolo orientado a conexão, sendo portanto, estabelecida uma conexão entre o cliente e o servidor, o HTTP é dito não orientado a conexão: uma vez que uma requisição seja satisfeita a conexão é desfeita.

Protocolo stateless

Após o servidor ter respondido a requisição do cliente, a conexão entre o cliente e o servidor é desfeita e esquecida. Não existe memória entre as conexões dos clientes. Um servidor HTTP trata toda requisição como se ela fosse a primeira vez, isto é, sem contexto.

Ou ainda, não é guardada nenhuma informação referente a estados de conexão.

Extensível conjunto de formatos

O protocolo HTTP usa o *Multipurpose Internet Media Types* (MIME) para prover aberto e extensível tipos de dados e negociação de tipos. Com o protocolo HTTP, onde o transmissor e o receptor podem se comunicar diretamente, as aplicações são permitidas mais livremente a usar tipos não registrados.

Quando o cliente inicia uma transação com o servidor, cabeçalhos são enviados de acordo com a especificação padrão *e-mail* (RFC822). Quando o servidor HTTP transmite informações em resposta a uma requisição, ele inclui um cabeçalho para informar ao cliente o tipo de dado seguindo o cabeçalho. A transação então depende de o cliente possuir o utilitário apropriado (visualizador de imagem, visualizador de imagem em movimento, ...) correspondente aquele tipo de dado.

Tipos de Mensagens

As mensagens HTTP consistem de requisições do cliente (*requests*) para o servidor e respostas do servidor (*response*) para o cliente. O formato das mensagens é definido na RFC 822. Ambas as mensagens podem incluir opcionalmente campos de cabeçalhos. O corpo da mensagem é separado do cabeçalho por uma linha nula.

Requests

Uma mensagem de requisição do cliente para o servidor inclui, na primeira linha da mensagem, o método a ser aplicado no recurso, o identificador do recurso, e a versão do protocolo em uso.

Response

Após receber e interpretar uma mensagem, o servidor responde na forma de uma mensagem de resposta HTTP

A primeira linha de uma mensagem de resposta é a linha de *status* (*Status-Line*), consistindo da versão do protocolo seguido por um código numérico de *status* e uma frase textual associada a este código, com cada elemento separado por um caractere de espaço "SP" e encerrada pela sequência "CRLF". Após esta linha seguem-se linhas de cabeçalhos.

Cabeçalhos das Mensagens

Uma transação HTTP consiste de um cabeçalho seguido opcionalmente por uma linha em branco e algum dado. O cabeçalho é uma informação genérica que pode, entre outras coisas, especificar a ação requerida no servidor, o tipo de dado que está sendo transferido ou um código de *status*. O uso dos campos de cabeçalho dá ao protocolo HTTP uma enorme flexibilidade. Estes campos permitem o uso de informações descritivas para serem enviadas na transação, fornecendo possibilidades para autenticação, encriptação e/ou identificação do usuário. O cabeçalho é referenciado como Meta Informação, por se tratar da informação da informação.

Content-Type

Este campo de cabeçalho indica o tipo de mídia do dado enviado para o recipiente ou, no caso do método HEAD, o tipo que deverá ser enviado quando for uma requisição usando o método GET. Este campo é usado pelos clientes para identificar o tratamento que deverá ser dado aos dados. Um exemplo:

- *Content-Type: text/html*

Os tipos de mídia são registrados com o IANA. O uso de mídias não registradas é desencorajado.

Content-Length

O campo de cabeçalho *Content-Length* indica, em um número decimal, a quantidade de octetos do objeto, enviado para o recepiente, ou no caso do método HEAD, a quantidade de octetos que seriam enviados no caso do método GET. Um exemplo:

- *Content-Length*: 1234

Date

O campo de cabeçalho *date* indica a data e a hora em que a mensagem foi originada. Um exemplo:

- Date: Sun, 05 Jan 1997 08:15:40 GMT

Expires

Este campo informa a data após a qual a informação deixará de ser válida.

From

O campo From, se fornecido, deverá conter um endereço eletrônico Internet (*e-mail address*) do usuário que controla o cliente. Um exemplo:

- *manoel@jerimum.crn.inpe.br*

Este campo pode ser usado com o propósito de "login" e é um meio de identificar a origem de requisições invalidas.

If-Modified-Since

O campo de cabeçalho If-Modified-Since é usado com o método GET para torná-lo condicional: se o recurso requisitado não foi modificado deste a data especificada neste campo, uma cópia do recurso não deverá ser retornada do servidor; e sim, um código de resposta 304 - veja quadro 5 e 6 com os códigos numéricos de resposta.

O método GET condicional requer que o recurso seja transferido somente se ele foi modificado após da data dada no campo *If-Modified-Since*.

Last-Modified

O campo de cabeçalho *Last-Modified* indica a data e a hora na qual o servidor julga que o recurso foi modificado pela última vez.

O exato significado deste campo depende da implementação do servidor e da natureza do recurso. Para arquivos, ele pode ser a data da última modificação do arquivo.

Location

O campo de cabeçalho *Location* retorna a exata localização de um recurso identificado em uma requisição. Se o seu valor for uma URL completa, o servidor retorna um *redirect* para o cliente resgatar o objeto especificado diretamente. Um exemplo:

- *Location*: http://www.crn.inpe.br/~manoel/http.html

Referer

O campo de cabeçalho *Refer* permite ao cliente especificar, para uso do servidor, o endereço (URI) do recurso do qual a requisição URI foi obtida. Isto permite ao servidor gerar uma lista de links reversos para recursos de interesses. Também permite links errados ou obsoletos serem descobertos.

Server

O campo de cabeçalho de resposta *Server* contém informação a respeito do *software* usado pelo servidor de origem. Este campo pode conter nomes do produto e subprodutos significantes. Exemplo:

- *Server*: CERN/3.0 libwww/2.17

User-Agent

O campo de cabeçalho *User-Agent* contém informação do cliente originador da requisição. Este campo pode ser usado com finalidades estatísticas, para rastrear violações do protocolo, e automatizar o reconhecimento de clientes com o propósito de enviar respostas sob medida para evitar possíveis limitações de clientes.

WWW-Autenticate

O campo de cabeçalho *WWW-Autenticate* deve ser incluído em uma resposta 401 (*unauthorized*). O valor do campo consiste de pelo menos um pedido de senha que indica o esquema de autenticação e os parâmetros aplicáveis.

Authorization

Um cliente que deseja autenticar-se (usualmente, mas não necessariamente, após receber uma resposta 401) pode fazê-lo incluindo um campo de cabeçalho *Authorization* na requisição.

Métodos

O protocolo HTTP permite um aberto conjunto de métodos a serem usados para indicar o propósito de uma requisição. Os três métodos mais usados são GET, HEAD, e POST.

Get

O método GET é usado para requerer um documento específico - quando você clica em um "link", o método GET é usado. GET deverá ser usado quando o acesso não modificará o estado do documento - mudando ou apagando informação. A semântica do método GET permite um GET condicional se a mensagem de requisição inclui um campo de cabeçalho *If-Modified-Since*. Um GET condicional requer que o recurso identificado seja transferido somente se este recurso foi modificado a partir da data informada pelo parâmetro do campo de cabeçalho *If-Modified-Since*.

O método GET condicional é usado para reduzir o tráfego desnecessário de dados na rede.

Head

O método HEAD é usado para requerer somente informação sobre o documento, e não o documento como um todo. O método HEAD é mais rápido que o método GET. Ele é sempre usado por clientes que usam *cache*, para ver se um documento foi modificado desde a última vez que foi acessado. Se ele não foi a cópia local pode, então ser reusada, caso contrário, a versão atualizada deverá ser retirada com o método GET. A metainformação contida nos cabeçalhos HTTP, enviada em resposta a uma requisição HEAD, deverá ser idêntica à resposta enviada a uma requisição GET. Este método é sempre usado para testar validade dos "links", acessabilidade, e modificações recentes.

Post

O método POST é usado para transferir dados do cliente para o servidor; ele é apropriado para cobrir funções como: anotação de um recurso existente; postar uma mensagem para um *bulletin board*, *newsgroup*, *mailing list*; prover um bloco de dados (usualmente um formulário) para um processo de manuseio de dados; estender uma base de dados através de uma operação de *append*.

Códigos numéricos de *status*

O elemento *Status-Code* enviado em uma resposta é um inteiro de 3 dígitos indicando a situação de uma requisição feita pelo cliente. Associado a cada código está uma frase descritiva. O código é para ser usado pela máquina e a descrição serve de referência. O cliente não examina a descrição e sim o código.

O primeiro dígito do código numérico define a classe da resposta. Os dois últimos não estão classificados em nenhuma regra. São definidos 5 (cinco) valores para o primeiro dígito:

Cod.	Classificação	Descrição
1xx	Informacional	Reservado para uso futuro
2xx	Sucesso	A ação foi recebida com sucesso, entendida, e aceita
3xx	Redireção	Uma outra ação deve ser executada para completar a requisição
4xx	Erro do Cliente	A requisição contém erro de sintaxe ou não pode ser realizada
5xx	Erro do Servidor	O servidor falhou ao realizar uma requisição aparentemente válida.

QUADRO 5. Valores de *StatusCod*

Os códigos numéricos de *status* definidos pelo protocolo HTTP/1.0 e a correspondente descrição, são apresentados na tabela abaixo. As descrições listadas são somente como recomendação, uma vez que não são interpretadas pelos clientes elas podem ser modificadas a gosto do implementador do servidor.

Código	Descrição
200	*OK*
201	*Created*
202	*Accepted*
204	*No Content*
301	*Moved Permanently*
302	*Moved Temporarily*
304	*Not Modified*
400	*Bad Request*
401	*Unauthorized*
403	*Forbidden*
404	*Not Found*
500	*Internal Server Error*
501	*Not Implemented*
502	Bad *Gateway*
503	*Service Unavailable*

QUADRO 6. Códigos Numéricos do *Status*

Segurança

Autenticação de Clientes

O esquema básico de autenticação do protocolo HTTP não é um método seguro para autenticação de usuários, nem protege a mensagem transmitida pela rede física de ser vista ou modificada.

Métodos Seguros

Se uma determinada aplicação necessita de um método seguro para garantir transações seguras entre um cliente e um servidor, através da rede Internet, a solução recomendada é o uso de um outro protocolo responsável exatamente por esta segurança. Um protocolo bastante usado em conjunto com o HTTP é o SSL (Secure *Socket*s Layer) desenvolvido pela Netscape e abordado no terceiro capítulo.

SEGUNDO CAPÍTULO
SEGURANÇA NO ACESSO

O bom senso é a coisa do mundo melhor partilhada, pois cada qual pensa estar tão bem provido dele, que mesmo os que são mais difíceis de contentar em qualquer outra coisa não costumam desejar tê-lo mais do que o têm.

Descartes, Discurso do Método, 1637

Informações, aplicativos e operações na rede estão ameaçadas se não for monitorado um conjunto de situações, entre elas a "porta dos fundos" e as "janelas da frente" da sua rede, pois *hackers* e funcionários descontentes, podem visar os "pontos de menor resistência" da rede, aqueles que ficam desprotegidos e vulneráveis a tentativas de acesso discado, ataques *replay*, cavalo de tróia, *backorifice* e muitas outras situações que podem ser extremamente perigosas e danosas as informações que trafegam na rede interna e Internet. Este capítulo tem por objetivo esclarecer e mostrar soluções para "fechar" algumas janelas e portas. Novamente saliento que não existe sistema seguro, como não existe casa ou cofre inviolável, mas sempre podemos aumentar a jornada de trabalho de usuários que buscam se apropriar de informações que lhes são indevidas.

2.1 FIREWALLS

O *Firewall* consiste em um conjunto de componentes organizados de uma forma a garantir certos requisitos de segurança. Normalmente o principal motivo para a instalação de um *Firewall* é proteger a rede privada da Internet, sendo que neste caso, uma política deve ser definida sobre o que pode e o que não pode passar por ele. Se uma determinada ação não é permitida, o *Firewall* garanti que todas as tentativas para realizá-la não lograrão êxito, bem como gerar registros de *log* sobre eventos suspeitos e alertar os administradores sobre tentativas que porventura possam comprometer a política de segurança.

Considerações de Projeto

Quando da configuração de um *Firewall* as principais decisões relativas à segurança da empresa já foram tomadas. No entanto, uma decisão deve ter sido tomada baseada no que é mais importante para a empresa: a segurança ou a facilidade de uso. Existem duas premissas básicas que podem melhor definir o conflito:

- Tudo que não é expressamente permitido é proibido – o *Firewall* é projetado para bloquear todos os serviços que não forem individualmente habilitados após uma criteriosa avaliação das necessidades e riscos, podendo no entanto, fazer com que os usuários possam enxergá-lo como um obstáculo;
- Tudo que não é expressamente proibido é permitido – o *Firewall* é projetado para bloquear somente os serviços expressamente definidos, deixando os administradores em posição reacionária, uma vez que terão que predizer quais ações dos usuários poderão enfraquecer a segurança do sistema e preparar-se para defender-se das mesmas.

Zonas de Riscos

Zonas de ricos correspondem às partes vulneráveis do sistema e todas as redes possuem algumas. No caso de uma rede conectada à Internet sem qualquer *Firewall,* ela como um todo está sujeita a ataques. Isto não implica que ela seja vulnerável, mas num caso como este, em que toda a rede pode ser alcançada por uma rede não confiável, torna-se necessário garantir a segurança de todos os *hosts* presentes na mesma. No caso da utilização de um *Firewall* a zona de risco é normalmente reduzida ao próprio *Firewall* ou ao subconjunto dos *hosts* da rede, diminuindo as preocupações dos administradores no que diz respeito a ataques diretos. O uso do *Firewall* representa a redução da zona de risco a um único ponto de falha, no entanto, se este for violado, esta novamente se expande, passando agora a incluir toda a rede protegida.

Existem várias maneiras pelas quais um *Firewall* pode falhar ou ser comprometido. O fato de um usuário ter um *login* de acesso direto ao próprio sistema de *Firewall* pode constituir em uma grande ameaça à segurança. Um situação problemática seria alguém acessar o *Firewall* e reconfigurá-lo de tal maneira que toda a rede privada se tornasse acessível a qualquer pessoa. No entanto, o pior desastre para um *Firewall* seria alguém comprometê-lo totalmente, e não ficar registrada qualquer informação de como ocorreu o ataque.

Componentes do *Firewall*

Os componentes básicos para a construção de um *Firewall* são:

Packet Filters

São responsáveis pela filtragem (exame) dos pacotes que trafegam entre dois segmentos de rede. Como um primeiro passo ao se implementar uma barreira de segurança em uma rede de computadores, é fundamental que se conheça os detalhes dos protocolos de comunicação utilizados. Na Internet, a atenção deve ser

voltada aos protocolos IP, TCP, ICMP e UDP. Estes são os principais protocolos a nível de rede e transporte (Modelo OSI) que são considerados e examinados ao se estabelecer regras de filtragem em um *packet filter* para a Internet. Este mecanismo de filtragem a nível de roteador possibilita que se controle o tipo de tráfego de rede que pode existir em qualquer segmento de rede; consequentemente, pode-se controlar o tipo de serviços que podem existir no segmento de rede. Serviços que comprometem a segurança da rede podem, portanto, ser restringidos. Logo, um *packet filter* não se encarrega de examinar nenhum protocolo de nível superior ao nível de transporte, como por exemplo o nível de aplicação que fica como tarefa dos *application gateways* (*proxy servers*). Portanto, qualquer falha de segurança a nível de aplicação não pode ser evitada utilizando somente um *packet filter*. O componente que realiza a filtragem de pacotes geralmente é um roteador dedicado, mas também pode ser um *host* de propósito geral configurado como roteador, e recebe a denominação de *screening router*. A filtragem que a maioria dos roteadores realizam são baseadas nas seguintes informações:

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

O roteador examina cada datagrama para verificar se ele se inclui em alguma das regras configuradas. Estas regras são baseadas na informação do cabeçalho do pacote que é disponibilizada para o processo de roteamento IP. Se um cruzamento é encontrado, e a regra permite o pacote, ele é despachado de acordo com a tabela de roteamento. No entanto, se a regra nega permissão ao pacote, o mesmo é descartado. Se não há regra definida para o pacote específico, um parâmetro *default* configurável pelo usuário determina quando este é roteado ou descartado.

No protocolo TCP existe um *flag* denominado ACK que é utilizado para confirmação de pacotes e também pode ser utilizado para detectar se o pacote é o primeiro de uma solicitação de conexão. Quando o *flag* não estiver setado significa que o pacote se refere a uma solicitação de conexão e, caso contrário, o pacote corresponde a alguma conexão já existente (quadro 1). Desta forma, o *packet filter* pode bloquear um serviço *inbound* (de fora para dentro; ou seja, o servidor está na rede interna) apenas não permitindo o fluxo de pacotes com o ACK setado destinado a um servidor interno associado a port (por exemplo, a port 23 do *telnet*) do serviço bloqueado. Em protocolos não orientados a conexão, por exemplo o protocolo UDP, não é possível tomar nenhuma decisão deste tipo; ou seja, nestes protocolos, nunca se sabe se o pacote que está chegando é o primeiro que o servidor está recebendo. Para fazer uma filtragem correta dos pacotes, é importante saber se o protocolo é bidirecional (pacotes fluem nos dois sentidos, cliente para servidor e vice-versa) ou unidirecional. Não se pode confundir serviços *inbound* (a rede interna provendo algum serviço) e serviços *outbound* (o cliente está na rede interna e o servidor na Internet) com pacotes *inbound* (pacotes que chegam na rede interna) e pacotes *outbound* (pacotes que saem da rede interna); ou seja, ambos os serviços apresentam pacotes *inbound* e *outbound* caso o protocolo seja bidirecional.

Eis alguns exemplos de regras de filtragem que poderiam ser aplicadas em um roteador:

- Bloquear todas as solicitações de conexão de *hosts* da rede externa com a sub-rede "X.X.8" (conectada em alguma interface do roteador), exceto conexões SMTP (Porta TCP número 25);
- Bloquear todas as conexões para e de certos sites considerados não confiáveis;
- Desabilitar *source routing* (roteamento de e para a máquina destino especificado no próprio pacote);
- Bloquear os serviços considerados inseguros tais como Xwindows, RPC, NFS, TFTP, SNMP, NIS, etc.

- Filtro Dependente do Serviço – As regras de filtragem de pacotes possibilitam a um roteador negar ou permitir o tráfego baseado em um serviço específico, uma vez que a maioria dos *Listeners* de serviços residem em portas TCP/UDP conhecidas. Por exemplo, o *Listener* Telnet para conexões remotas é na porta TCP número 23. Nesse caso, para bloquear todas as conexões Telnet, basta descartar todos os pacotes que contenham como destino uma porta TCP igual a 23.

- Filtragem Independente do Serviço – Há certos tipos de ataques que são difíceis de identificar utilizando-se as informações existentes no cabeçalho do pacote, porque são independente do serviço. Os roteadores podem ser configurados para prevenir esses tipos de ataques, porém as regras de filtragem precisarão de informações adicionais que poderão ser obtidas examinando-se a tabela de roteamento, inspecionando-se opções específicas do IP, conferindo fragmentos especiais de balanceamento, etc. Exemplos destes tipos de ataque incluem:

1. Disfarce do Endereço IP Fonte (*IP Spoofing*) – O intruso transmite pacotes de fora da rede como se estivesse dentro dela. Os pacotes contém um endereço IP adulterado para um de sistema interno. O intruso espera que o uso do endereço interno o permita penetrar em um sistema de segurança simples baseado apenas em que os pacotes de um específico servidor considerado confiável são aceitos e os demais descartados. É importante que o roteador tenha facilidades de filtragem por interfaces de rede. Ou seja, todas as interfaces disponíveis no roteador são submetidas às regras de filtragem, possibilitando que as regras sejam aplicadas considerando as seguintes informações:
   - A interface na qual o pacote chega;
   A interface pela qual o pacote sai.

   Se a filtragem é realizada por interface em ambos os sentidos, este ataque não funciona porque jamais um pacote pode chegar do mundo externo (Internet) tendo como endereço fonte o endereço de uma máquina que está na rede interna

b) Ataques por Roteamento de Fonte (*Source Routing*) – *Source routing* é um mecanismo de roteamento que consiste no seguinte: junto ao pacote, além das informações convencionais, envia-se a rota (as máquinas pelas quais deve passar) que o pacote deve seguir até o destino, de forma que as máquinas intermediárias por onde o pacote trafegar não utilizarão as suas potencialidades de roteador para definir a rota, seguindo estritamente a rota especificada no pacote. Utilizando IP *spoofing* e *source routing* fica fácil ao intruso obter sucesso em suas investidas (além de forjar o endereço ele garante que os pacotes seguirão direto para a sua máquina). Este tipo de ataque pode ser descartado rejeitando-se todos os pacotes que a opção de roteamento na fonte.

c) Ataque por Minúsculos Fragmentos – Nesse caso, o intruso usa a opção de fragmentação para criar fragmentos extremamente pequenos e com isso forçar que a informação de cabeçalho siga em um fragmento separado do pacote. Isso é planejado para regras de filtragem que examinam apenas o primeiro fragmento e permitem que os demais passem. Dependendo do roteador este ataque pode ser evitado descartando-se todos os pacotes cujo protocolo seja TCP e o campo de Fragmento *Offset* esteja setado.

d) *Source address*: o intruso forja o endereço fonte utilizando o endereço de uma máquina (externa ou interna) considerada confiável (*trusted*) pelo *Firewall*. Este ataque pode ter sucesso principalmente quando o intruso não precisa capturar (ou seja, estar em um caminho entre o *Firewall* e o *host* forjado) nenhum pacote e quando, caso a máquina forjada seja interna, não houver mecanismos de filtragem que impeçam o *ip spoofing* (citado anteriormente); a resposta ao ataque poderia ser o envio de alguma informação (por exemplo o arquivo *passwd*) via email diretamente ao intruso;

e) *Man in the middle*: além de forjar o endereço, nesse ataque o intruso deve estar no caminho entre o *Firewall* e o *host* confiável porque ele tem de capturar os pacotes que são, na realidade, enviados ao *host* confiável (daí a denominação do ataque).

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

- Confundindo o roteamento entre a máquina real (*host* confiável) e a máquina alvo;
- Utilizando um ataque onde somente a primeira resposta é requerida, de tal forma que o *reset* solicitado pela máquina real não importará;
- Inundando a máquina real com pacotes lixo (por exemplo, pacotes ICMP) enquanto o ataque ocorre, de forma que a máquina real ficará ocupada tentando processar os pacotes lixo que ela recebe;
- Utilizando *source routing*.

Múltiplos roteadores

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

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

Operações de um Packet Filter

Quase todos os dispositivos atuais de filtragem de pacotes operam da seguinte maneira:

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

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

Atividades Básicas de um Roteador

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

a) Realizar *logs* de acordo com a configuração especificada pelo administrador. Dessa forma, é possível analisar eventuais tentativas de ataque, bem como verificar a correta operação do sistema;
b) Retorno de mensagens de erros ICMP: aso um pacote seja barrado existe a possibilidade de se enviar ao endereço fonte alguma mensagem com o código de erro ICMP do tipo *host unreachable* ou *host administratively unreachable*. Entretanto, tais mensagens, além de causar um *overhead*, podem fornecer algumas informações sobre o *packet filter* ao intruso, pois dessa forma ele poderia descobrir quais os protocolos que são barrados e quais estão disponíveis; portanto, recomenda-se que não se retorne nenhum código ICMP de erro para a rede externa.

Características desejáveis em um roteador

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

a) Ter uma boa performance na filtragem dos pacotes: um *overhead* aceitável de acordo com as necessidades;
b) Pode ser um roteador dedicado ou um computador de propósito geral executando algum sistema de roteamento;
c) Permitir uma especificação de regras de forma simples;
d) Permitir regras baseadas em qualquer cabeçalho ou critério *meta-packet* (por exemplo, em qual interface o pacote chegou ou está saindo);
c) Aplicar as regras na ordem especificada;
d) Aplicar as regras separadamente para pacotes que chegam e partem em e de cada interface de rede;
e) Registrar informações sobre pacotes aceitos e rejeitados;
f) Ter capacidade de teste e validação.

Vantagens dos packet filters:

a) Pode ajudar a proteger toda uma rede, principalmente se este é o único *roteador* que conecta a rede interna à Internet;

b) A filtragem de pacotes é transparente e não requer conhecimento nem cooperação dos usuários;

c) Está disponível em muitos *roteador*.

Limitações dos Packets Filters

Definí-los pode se tornar uma tarefa um tanto complexa, uma vez que administradores de rede terão que ter um entendimento detalhado dos vários serviços da Internet, formato dos cabeçalhos de pacotes e valores específicos que podem ou devem ser esperados em cada campo. Se requisitos complexos de filtragem são suportados, a regra pode tornar-se muito longa e complicada, tornando difícil sua manutenção e gerenciamento. Por fim, há poucas facilidades de testes para verificar a correção das regras depois que as mesmas foram configuradas no roteador, fazendo com que a rede fique potencialmente exposta a ataques não testados.

Dependendo da complexidade das regras de segurança estabelecidas, do nível de envolvimento dos usuários e outros fatores, o administrador da rede pode ser levado a acreditar-se seguro com seu *Packet Filter*, quando na verdade ainda existem "portas" entre abertas para intrusos mais dedicados.

Qualquer pacote que passe pelo roteador pode ser utilizado para disparar o ataque. Não podemos esquecer que um ataque ocorre quando dados aparentemente não nocivos são enviados pelo roteador ao servidor e contém instruções que causam modificação nas permissões de acesso e em arquivos relacionados com a segurança, tornando fácil a entrada do intruso no sistema.

Bastion Host

Qualquer máquina configurada para desempenhar algum papel crítico na segurança da rede interna; constituíndo-se na presença pública na Internet, provendo os serviços permitidos segundo a política de segurança da empresa.

Marcus Ranum é um dos responsáveis pela popularidade deste termo na comunidade profissional de *Firewall*, segundo ele "*bastions* são áreas críticas de defesa, geralmente apresentando paredes fortes, salas para tropas extras, e o ocasional útil repositório de óleo quente para desencorajar os intrusos". (RANUN, et. al., 1997)

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

O *bastion host* tem responsabilidades diferentes do *packet filter*, dependendo do seu tipo. Alguns autores enfatizam que enquanto o *packet filter* atua em um nível mais baixo, o *bastion host* se encarrega de todos os níveis (referentes ao modelo OSI). Na realidade, um *host* pode acumular tanto as funções de filtragem de pacotes como também pode prover alguns serviços; neste caso, ele seria um *packet filter* e *bastion host* simultaneamente. Traçando um paralelo destes dois componentes em relação ao modelo OSI, o *packet filter* realiza algum exame dos pacotes até o nível 4 (transporte) enquanto que o *bastion host* se encarrega basicamente dos níveis superiores (fundamentalmente o de aplicação, nível 7).

Este tipo de máquina também recebe a denominação de *application gateway* porque funciona como um *gateway* a nível de aplicação. Os servidores disponíveis nos *bastion host* são denominados de *proxy servers*; ou seja, servidores por procuração que atuam como intermediários entre o cliente e o servidor. Neste caso, os serviços só podem ser providos via *bastion host*, obrigando o cliente (por exemplo, via regras de filtragem nos roteador) a acessar estas máquinas. Portanto, este é um mecanismo que garante que o serviço será provido de forma segura para

usuários internos e externos e impede que o *bastion host* seja desviado (a não ser que o roteador seja comprometido e as regras de filtragem alteradas).

O *proxy server* pode ser configurado para suportar apenas opções específicas de uma determinada aplicação e negar as demais, de acordo com as necessidades do administrador da rede. É importante notar também que o usuário é autorizado a acessar os serviços *proxy*, mas não a se logar no *gateway* de aplicação. Se isso acontecer, a segurança do *Firewall* fica comprometida, uma vez que um intruso poderia por exemplo acessar a conta *root* e instalar um "cavalo de tróia" para capturar senha e modificar arquivos de configuração de segurança.

Existem várias opções de projetos para prover segurança nos *bastions hots*, dentre as quais:

- Apenas os serviços que o administrador considere necessários devem ser instalados no *bastion host*. A razão para isso é simples: serviço não instalado não pode ser atacado. Geralmente um conjunto limitado de aplicações *proxy* como TELNET, DNS, FTP, SMTP e autenticação de usuários são instalados;
- O *bastion host* pode requerer autenticação adicional antes de um usuário ser autorizado a acessar os serviços *proxy*, constituindo-se no local ideal para instalar a autenticação forte, como por exemplo uma tecnologia de senha descartável onde um autenticador criptográfico gera um único código de acesso, podendo ainda cada serviço *proxy* requerer sua própria autenticação antes de autorizar o acesso;
- Cada *proxy* pode ser configurado para suportar apenas um subconjunto padrão de comandos da aplicações, fazendo com que comandos não suportados não sejam disponibilizados ao usuário, mesmo depois de sua autenticação;
- Cada *proxy* é configurado para permitir acesso apenas a sistemas específicos em servidores específicos. Isso implica que a restrição ao

comando ou opção pode ser aplicada a apenas um subconjunto de sistemas na rede protegida;

- Todo *proxy* mantém informação detalhada de auditoria, registrando todo o tráfego, toda conexão e sua duração. O arquivo de *log* é essencial para se descobrir e impedir a continuidade do ataque de intrusos;
- Cada *proxy* é independente dos outros instalados no *bastion host*, ou seja se ocorrer um problema ou uma vulnerabilidade for detectada, este poderá ser substituído sem afetar as demais aplicações. Por outro lado, se o usuário solicitar suporte para algum serviço não disponível, o administrador poderá facilmente instalar o *proxy* requerido no *bastion host;*
- Um *proxy* não faz acesso a disco, a não ser para ler seu arquivo de configuração inicial, o que faz com que um intruso tenha dificuldade para instalar "cavalos de Tróia", capturadores de senha e outros arquivos potencialmente perigosos no *bastion host.*

Tipos Especiais de *Bastion Hosts*

Dependendo da localização do *bastion host* dentro do *Firewall*, tem-se alguns tipos de máquinas com funções diferenciadas na segurança, que são:

- *Dual homed host*: trata-se de um computador com duas interfaces de rede conectadas cada uma a segmentos diferentes de rede. Uma das características fundamentais dessa configuração é que o roteamento direto (IP *forwarding*) é desabilitado e, portanto, todo o roteamento é realizado a nível de aplicação. Neste caso, todos os serviços segurados podem ser fornecidos via procuração (*proxy servers*) e somente o tráfego referente aos serviços habilitados via *proxy* e aqueles especificados pelas regras de filtragem circulam entre os dois segmentos de rede conectados ao *bastion host*;

- *Victim machines*: estas máquinas abrigam serviços que não são considerados fáceis de serem segurados. A máquina é configurada basicamente somente com os serviços fornecidos para garantir que nada mais significativo esteja a disposição do atacante caso a máquina seja comprometida. Geralmente uma *screened subnet* possui uma ou mais máquinas deste tipo;
- *Internal bastion hosts*: são aquelas máquinas com maior interação com as máquinas internas (por exemplo, uma máquina que recebe o *e-mail* e o reenvia a um servidor de correio eletrônico residente na rede interna).

Proxy Systems

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

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

O *proxy server* atua como um procurador que aceita as chamadas que chegam e checa se é uma operação válida. Após receber a chamada e verificar que a

solicitação é permitida, o servidor procurador envia adiante a solicitação para o servidor real. A procuração atua como servidor para receber a solicitação que chega e como um cliente quando envia adiante a solicitação. Depois que a sessão é estabelecida, a aplicação procuradora atua como uma retransmissora e copia os dados entre o cliente que iniciou a aplicação e o servidor. Devido ao fato de todos os dados entre o cliente e o servidor serem interceptados pelo *proxy* ele tem controle total sobre a sessão e pode realizar um *logging* tão detalhado quanto se desejar.

Para serviços que não apresentam características originais de *proxying*, os detalhes do funcionamento do procurador dependem de cada protocolo (serviço) em questão. Em contrapartida, a comunicação do programa cliente com o servidor *proxy* pode ser realizada de duas formas distintas:

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

Algumas vantagens dos *proxy systems* são:

- Permitem aos usuários acesso direto aos serviços na Internet: apesar de haver um procurador atuando em nome do cliente, este mantém a ilusão de estar ser comunicando diretamente com o servidor remoto;
- Mecanismos de *log*: como todo o tráfego dos serviços passa pelo servidor procurador, e tudo até o nível de aplicação, uma grande quantidade de informações podem ser registradas de acordo com as necessidades de auditoria e segurança.

Algumas desvantagens dos *proxy systems* são:

- Há um atraso significativo entre o surgimento de um novo serviço e um correspondente servidor *proxy*;
- Pode ser necessário utilizar diferentes servidores procuradores para cada serviço;
- Geralmente requerem modificações nos clientes, nos procedimentos ou em ambos;
- Alguns serviços não são viáveis para operar via procuradores (exemplo: *talk*, que é parte baseado em TCP e parte em UDP);
- Um serviço por procuração não protege contra todas as fraquezas dos protocolos, depende da habilidade de se determinar que operações são seguras em um determinado protocolo.

Quando um servidor procurador atende a um único serviço, recebe a denominação de servidor dedicado ou *application level*. Quando um servidor atende a uma certa quantidade de serviços este é denominado servidor genérico ou *circuit level*. Um procurador genérico não pode interpretar o protocolo de aplicação e precisa obter informações através de outros meios para poder atender aos serviços. Servidores genéricos simplesmente transmitem conexões TCP sem executar nenhum processamento ou filtragem de pacotes, entretanto é aparente para o

sistema externo que a conexão tem origem a partir de um *Firewall* e dessa forma as informações sobre a rede protegida são omitidas. *Circuit level* são frequentemente utilizados para conexões de dentro para fora, onde o administrador confia nos usuários internos. Sua principal vantagem é que o *bastion host* pode ser configurado como um *gateway* híbrido, suportando aplicações ou serviços *proxy* para conexões de fora para dentro e de *circuit level* de dentro para fora. Isso possibilita que o *firewall* fique mais simples para uso pelos usuários internos que desejam acesso aos serviços Internet, enquanto continua executando as funções de proteger a organização de um ataque externo.

Arquiteturas de *Firewalls*

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

a) ***Screened Host*** – esta arquitetura consiste de um *screening router* e *bastion host* com uma única interface de rede. Geralmente o *bastion host* pertence à rede privada e o *screening router* é configurado de tal forma que o *bastion host* é o único sistema da rede privada que pode ser alcançado a partir da Internet. O *screening router* é configurado para bloquear todo tipo de tráfego para a rede privada e permitir comunicação somente para portas específicas do *bastion host.* Esta não é uma arquitetura considerada muito segura. Veja a seguir algumas observações sobre como ela satisfaz ou não as estratégias de segurança:

- *Least privilege*: pode ser observada quando os serviços são fornecidos exclusivamente via procuradores; entretanto, o fato do *bastion host* estar situado na rede interna e além disso acumular privilégios de vários servidores pode ser um fator que vá contra o princípio do mínimo privilégio devido a sua posição crítica.
- *Defense in depth*: esta estratégia não é satisfeita principalmente porque basta que um dos componentes, roteador ou servidor, seja comprometido para que toda a rede interna esteja ao alcance do atacante.
- *Choke point*: o roteador é o *choke point* nesta arquitetura.
- *Weakest link*: o *bastion host* é o alvo mais visado pelos atacantes porque ele é o servidor de diversos serviços e está localizado junto à rede interna.
- *Fail safe*: esta não é uma arquitetura que permite falhas seguras de uma forma geral. Visto que basta comprometer o *bastion host* para se ter acesso a rede interna. Um certo nível de falha segura é possível da mesma forma como foi exposto para a arquitetura *screened subnet* caso o *screening router* seja configurado segundo a filosofia "o que não é expressamente permitido é proibido".
- *Universal participation*: caso o roteador seja a única via de acesso à Internet, tem-se participação involuntária dos usuários da rede interna. Entretanto, valem as mesmas observações feitas para a arquitetura *screened subnet*.
- *Diversity of defense*: pouca ou nenhuma oportunidade de se aplicar esta estratégia pois há um único roteador e *bastion host*.

b) ***Screend Subnet*** – nesta arquitetura cria-se uma rede isolada entre a Internet e a rede privada. Tipicamente essa rede é isolada utilizando-se *screening routers* que podem implementar uma variedade de níveis de filtragens. Normalmente uma *screened subnet* é configurada de tal forma que tanto a Internet

quanto a rede privada tenham acesso à mesma, porém um *bastion host* é o único ponto de acesso. Composta por componentes básicos (*packet filters e bastion hosts*) esta é uma arquitetura que apresenta múltiplos níveis de redundância e provê um bom esquema de segurança, constituindo-se em um exemplo clássico de arquiteturas de *firewall*.

Os componentes deste tipo de *firewall* incluem os seguintes:

- *Perimenter Network*: sub-rede situada entre a rede interna e a rede externa (Internet);
- Roteador externo: diretamente conectado à Internet e à *perimeter network*;
- Roteador interno: diretamente conectado à rede interna e à *perimeter network*;
- *Bastion hosts* residentes na *perimeter network*.

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

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

Uma das principais vantagens das *screened subnets* com bloqueio de roteamento entre as redes é que para atacá-las com sucesso o intruso deve reconfigurar o roteamento entre as três redes, sem desconectar e sem bloquear a ele mesmo, e sem que as mudanças de roteamento sejam notadas.

Análise frente as Estratégias de Segurança

Para melhor definir quão seguro esta arquitetura pode ser, veja a seguir algumas observações sobre como ela satisfaz ou não as estratégias de segurança:

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

características dos procuradores e das facilidades disponíveis nos programas clientes. De qualquer forma, a educação dos usuários é fundamental para que se tenha compreensão das medidas de segurança adotadas. Se não houver cooperação, o *Firewall* pode deixar de ser o único ponto de acesso à Internet bastando para isso que um usuário descontente acesse um provedor via linha discada.

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

c) ***Firewalls*** **usando** ***Screening Routers*** – nesta arquitetura é utilizado apenas um *screening router* para proteção da rede privada, liberando-se dessa forma a comunicação direta entre os múltiplos *hosts* da rede privada e os múltiplos *hosts* da Internet. A zona de risco neste caso é igual ao número de *hosts* da rede privada e o número e tipos de serviços aos quais o *screening router* permite tráfego. É muito difícil detectar a destruição total de um *firewall* baseado somente em *screening router*. Se um roteador comercial (que não mantém registros de *log*) é usado e a senha de seu administrador é comprometida, toda a rede privada pode estar suscetível a ataques. *Firewalls* baseados em *screening router* são populares por serem de implementação barata e permitirem acesso à Internet de qualquer ponto da rede privada, porém não devem ser utilizado para proteger informações sensíveis ou confidenciais, uma vez que não se constituem numa solução muito segura.

Observações

Em se tratando de *firewalls*, não se pode facilmente afirmar qual arquitetura é melhor para uma solução particular, desde que existem vários fatores que devem ser considerados quando da análise da situação, tais como, custo, políticas corporativas, tecnologias de rede existentes, políticas intra-organizacionais etc.

Existem algumas observações que são relevantes no que diz respeito ao uso de *firewall* de uma maneira geral. Primeiramente, um *firewall* é um dispositivo de fortalecimento da segurança do ponto de vista de gerenciamento da rede. O tamanho da zona de risco é crucial para o projeto. Se ela é pequena, a segurança pode ser mantida e controlada facilmente. No entanto, se a segurança for comprometida os danos podem ser maiores.

Outro aspecto importante a considerar na construção de um *firewall* é que ele não deve ser projeto a vácuo. Na inicialização de um *firewall* deve-se fazer um balanço de tempo e dinheiro, segurança e risco.

Finalmente, é importante que quando se decida implementar um *firewall* resista-se à ânsia de começar do nada. Aprender com as experiências alheias, com suas falhas e sucessos é muito importante.

TERCEIRO CAPÍTULO
SEGURANÇA NAS TRANSAÇÕES

Segurança é minimizar a vulnerabilidade.
Vulnerabilidade é qualquer fraqueza que pode ser explorada para se violar um sistema ou as informações que ele contém.

ISO – Basis Reference Model, 1989.

A conquista por usuários em *sites* com transações comerciais ou mesmo um *Home Banking* é dificultada pela insegurança que existe no tráfego de informações sigilosas. Os usuários temem que ao inserir o número do seu cartão de crédito ou de um documento pessoal como o CPF, esta informação possa estar sendo recebida também por terceiros, ou alterada indevidamente. A segurança de transações é de fato, um dos fatores mais importantes para o ganho de credibilidade com os usuários(clientes) de um *site*.

Existem diversos tipos de mecanismos de segurança que podem e estão sendo utilizados para garantir este tráfego sigiloso da informação. Nas transações são utilizados algoritmos de criptografia dos dados, autenticação eletrônica, verificação da integridade das mensagens e identificação da origem.

Um protocolo é uma série de passos, envolvendo um ou mais participantes, objetivando executar uma tarefa. A escolha de um protocolo adequado e eficiente é fundamental para a segurança dos dados que são compartilhados durante as transações bem como do resultado das tarefas realizadas pelas aplicações. Comumente são utilizados à nível de protocolos o SSL, SHTTP, TLS e PCT que apresentam soluções como autenticação na conexão e criptação dos dados.

Inicialmente serão definidos alguns conceitos básicos, para após discutirmos protocolos e sistemas disponíveis, finalizando com alguns comentários sobre a segurança em redes TCP/IP.

3.1 CRIPTOGRAFIA

É um método antigo para "embaralhar" informações. Foi sendo aperfeiçoado e temporariamente as chaves utilizadas para criptografar os dados são aumentadas de tamanho para se adaptarem aos novos mecanismos de quebra e a processadores mais velozes. Existem basicamente dois sistemas criptográficos, o Simétrico ou de chave única e o Assimétrico ou de chave pública.

Criptografia assegura isolamento, mantendo a informação escondida de qualquer um para quem a informação não é intencional.

Existem compromissos de desempenho associados à criptografia, que deve ser usada depois de analisar os riscos de segurança e identificar consequências graves se os dados não forem confidenciais e a identidade dos remetentes não estiver garantida. Em redes internas e em redes que usam a Internet apenas para navegação na Web, correio eletrônico e transferência de arquivos, a criptografia normalmente não é necessária. Já para organizações que conectam sites particulares, usando VPN, a criptografia é recomendada para proteger o caráter confidencial dos dados da organização.

A meta da criptografia é que mesmo que o algoritmo seja conhecido, sem à chave apropriada, um intruso não possa interpretar a mensagem. (OPPENHEIMER, 1999)

Criptoanálise

É a técnica de recuperação de informações cifradas que podem ser através do algoritmo conhecido ou não. Quando não se conhece o algoritmo a criptoanálise possui três variações: quando se tem uma certa quantidade de texto cifrado e nenhum texto simples – somente texto cifrado; quanto possui algum texto cifrado e o texto simples correspondente – texto simples conhecido e; quando possui a capacidade de criptografar peças de texto simples de sua própria escolha – texto simples escolhido. No caso de ataque a um algoritmo perfeitamente conhecido como é o caso do DES, possui dois métodos:

Exaustão (Força Bruta)

Pelo método de pura exaustão, aplica-se o algoritmo a um determinado texto cifrado, variando-se à chave até que seja produzido um texto em claro, descobre-se desta forma qual à chave usada e todos os textos cifrados com ela são consequentemente decifrados. Mesmo quando a sistemática de exaustão for realizada por computadores muito rápidos, o uso de um grande número de chaves inviabilizaria o processo.

Quebra do algoritmo

A "quebra do algoritmo", consiste na descoberta de brechas em sua estrutura, as quais o criptoanalista consegue explorar reduzindo a exaustão a tempos razoáveis. Um algoritmo é considerado realmente seguro quando não existe nenhuma maneira mais rápida de quebrá-lo que não seja a da exaustão ou força bruta. Dois fatores são necessários, mas não suficientes, para tornar possível o descobrimento de fraquezas nos algoritmos:

- mesmo utilizando-se de sequências e aparência aleatória, os algoritmos têm comportamento determinístico;
- a linguagem a ser cifrada possui redundâncias, ou seja, frequência de repetição das letras.

Sistema simétrico - chave única

O sistema de chave única o utiliza a mesma chave, a qual deve ser mantida em segredo, para criptação e decriptação dos dados. Um sistema com algoritmo de chave única tem como vantagem ser muito mais rápido nos processos de criptografar/descriptografar do que os sistemas que utilizam chave pública (um par de chaves), pode ser implementado em hardware por ser extremamente flexível. Com esse ganho em velocidade, os sistemas que utilizam chave única são os mais adequados para criptografia off-line, onde o usuário necessita apenas armazenar localmente seus arquivos cifrados e transmiti-los, através de outros meios.

Em contrapartida, a desvantagem desses tipos de sistemas é se a mensagem precisa ser transmitida e com à chave. O receptor e o transmissor precisam combinar uma senha ao utilizar um meio seguro para transmitir esta informação. Um meio realmente seguro para transmitir a informação tem um alto custo e é difícil de se obter, e se este fosse o caso não haveria a necessidade da criptação dos dados, as mensagens poderiam ser encaminhadas por ele.

Sistema assimétrico - chave pública

Os sistemas de chave pública começaram a aparecer na segunda metade da década de 70. Em 1976, Whitfield Diffie e Martin Hellman e independentemente Ralph Merkle introduziram o conceito de criptografia de chave pública e privada.

Este sistema utiliza uma chave para criptografar os dados e outra para descriptografar, onde um usuário de um sistema transforma em pública uma chave para todos aqueles que queiram mandar-lhe mensagens, porém somente ele possuirá à chave de decriptação (privada).

Os algoritmos de chave pública vieram preencher a desvantagem do sistema de chave única, o problema de distribuição de chaves. Esta tarefa, difícil e perigosa de ser realizada através de algoritmos de chave única, fica extremamente simples se existe à chave pública onde todos do sistema têm acesso, sem a necessidade de um

meio seguro de transmissão, pois à chave privada que será utilizada para descriptografar a mensagem recebida fica restrita ao seu proprietário.

Os sistemas de chaves pública fornecem ao mesmo tempo recursos para preservar o caráter confidencial e recursos de autenticação. Usando chaves assimétricas, um destinatário pode verificar se a mensagem veio do usuário ou *host* do qual parece ter sido originado.

Pode-se utilizar as chaves para garantir a confidencialidade dos dados e juntamente para verificar a assinatura digital. Para isso, depois de criptografar o documento com à chave privada, pode-se também criptografá-lo com à chave pública do destinatário. O destinatário decodifica o documento duas vezes. Se o resultado final for o texto não formatado, o destinatário saberá que o documento veio da origem esperada e que era para o destinatário e mais ninguém (a descriptografia é feita com à chave pública da origem e com a privada do destinatário).

Algoritmos de Criptografia Simétrica

DES - Data Encrypt Standard (RFC1829)

O DES se transformou no principal e mais difundido algoritmo de chave única do mundo e o único até hoje a se transformar em padrão.

O DES é um ciframento composto que cifra blocos de 64 *bits* (8 caracteres) em blocos de 64 *bits*, para isso ele se utiliza de uma chave composta por 56 *bits* mais 8 *bits* de paridade (no total são 64 *bits*). A rigor, é uma substituição monoalfabética, mas as técnicas publicadas de quebra de substituições monoalfabéticas se aplicam apenas a alfabetos pequenos.

A questão da segurança do DES criou polêmica desde sua criação. Existem muitas especulações, inclusive sobre a existência de uma *trap door* (uma entrada por onde seria mais fácil o deciframento por parte do governo americano através da NSA) e também a respeito do número de *bits* da chave, do número de interações, do

formato das caixas S e uma série de problemas que foram apontados já na época da publicação do algoritmo e até bem pouco tempo atrás ainda não passavam de especulações.

Através dos anos, existiram vários ataques bem sucedidos contra variantes do DES com poucas interações. Em 1982 o DES foi facilmente quebrado com 4 interações, alguns anos depois o mesmo ocorreu para 6 interações. A teoria de Shamir e Biham explica que o DES com menos de 16 *bits* é mais facilmente quebrado pelo método do "texto conhecido" do que pela "força bruta".

A criptografia DES está disponível para a maioria dos roteadores e muitas implementações de servidores, embora o governo federal tenha descredenciado o efetivo do DES em 1988 (TANENBAUN, 1994).

IDEA

É um algoritmo desenvolvido no ETH Zurique na Suíça. Utiliza uma chave de 128*bits* e é considerado muito seguro. É atualmente um dos melhores algoritmos conhecidos e nenhum ataque prático bem sucedido foi publicado, apesar de numerosas tentativas.

IDEA é patenteado nos Estados Unidos e na grande maioria dos países europeus. A patente é conhecida como *Ascom-Tech.*

RC4

É um sistema de criptografia modelado por *RSA Data Security, Inc.* Era de uso privado comercial, até que foi publicado um código fonte nas notícias da USENET, reivindicando ser o equivalente do RC4. O algoritmo é muito rápido, sua segurança não é conhecida, mas quebrá-lo não parece trivial. Por causa de sua velocidade, tem uso em muitas aplicações. Aceita chaves de comprimento arbitrário. É essencialmente um gerador de números psedo-aleatórios.

Algoritmos de Criptografia Assimétricos

RSA (Rivest-Shamir-Adelman)

Baseia-se na propriedade da complementaridade dos polinômios. à chave é dividida em duas partes, uma sendo o complemento da outra; uma delas é privada, de conhecimento somente do seu proprietário e a outra de conhecimento público. O texto criptografado por uma delas somente pode ser descriptografado pela outra, e nunca pela parte que criptografou a mensagem. (CARUSO; STEFFEN, 1991)

Este algoritmo resolve o problema da distribuição da chave, típico do sistema simétrico. Possui três fases. Na fase 1 - é determinado as chaves públicas e privadas. A fase 2 - envolve a criptografia da mensagem pelo remetente que usa à chave pública do destinatário, e finalmente a fase 3 - que realiza a descriptografia da mensagem pelo destinatário que utiliza da sua chave privada. Pode-se utilizar este tipo de algoritmo para autenticação, sendo que o remetente deve utilizar sua chave privada para criptografar a mensagem e encaminhá-la juntamente com sua chave pública ao destinatário. O destinatário realiza a descriptografia usando à chave pública do remetente, como somente o proprietário possui à chave privada, prova assim, a identidade do remetente.

De acordo com Rivest et al. (in TANENBAUM, p. 618, 1994), "a fatoração de um número de 200 dígitos requer 4 bilhões de anos de tempo de computação; faturar um número de 500 dígitos exige 10^{25} anos. Em ambos os casos, os autores supõem o melhor algoritmo conhecido e um tempo de 1s para cada instrução." Por isso é considerado um dos métodos mais seguros e viáveis, pois mesmo com o crescente aumento de performance dos equipamentos, "decorrerão séculos antes que se torne viável faturar um número com 500 dígitos". (Idem)

Mas devemos saber que o RSA é muito vulnerável a ataques *chosen plaintext* e também há um ataque *timing* que pode ser usado para quebrar muitas implementações de RSA. Acredita-se que o algoritmo de RSA é confiável quando usado corretamente, mas deve-se ter muito cuidado quando o usar para evitar estes ataques.

Diffie-Hellman

É considerado um sistema seguro quando chaves suficientemente longas e geradores formais são usados. A base do algoritmo de Diffie-Hellman é a dificuldade do problema de logaritmo discreto (acredita-se que é computacionalmente equivalente a fatorar números inteiros grandes).

Deve-se notar que os resultados que concluem isso fazem precomputações, é possível computar logaritmos discretos eficazmente relativo a um específico eeficiente nº primo. O trabalho necessário para a precomputação é aproximadamente igual ou ligeiramente mais alto que o da fatoração de um número composto do mesmo tamanho. Em prática isto significa que se o mesmo início é usado para um número grande de trocas, deveria ser maior que 512 pedaços em tamanho, preferentemente 1024 pedaços.

Aplicativo PGP (RSA)

Programa desenvolvido por Philip Zimmermann em 1991, que inicialmente sofreu sanções e chegou a ser proibido durante dois anos nos EUA, por infringir a patente do algoritmo RSA. Hoje é totalmente legal, sem restrições de uso e é também um *freeware*.

Utilizando para criptografia de mensagens de correio eletrônico, autenticação e arquivos. Do ponto de vista do usuário é um sistema de chave pública mas internamente é híbrido. Quando criptografa um arquivo, ele cria uma chave única, que é encaminhada ao destinatário usando um cabeçalho que é criptografado com o algoritmo de chave pública RSA. Assim o RSA serve como "canal seguro" de transmissão da chave única. É realizado desde modo pois a criptografia somente pelo método RSA tem pouca performance comparado com método de chave única.

As chaves são armazenadas em *keyrings*, as públicas normalmente em um arquivo chamado *pubring.pgp* e as chaves privadas no arquivo *secring.pgp*. à chave privada é armazenada criptografada (chave única), sendo é protegida por uma frase senha (128*bits*) digitada no momento da especificação das chaves (pública e privada). Toda vez que for utilizar à chave privada (descriptografar ou assinar uma mensagem) esta frase senha deve ser digitada.

Funções Hash de Criptografia

MD5 (RFC1321/1828)

É um algoritmo de *hash* seguro desenvolvido RSA Data Security, Inc. Pode ser usado com um *hash* de tamanho arbitrário de 128 *bits*. MD5 é muito utilizado e considerado razoavelmente seguro.

Porém, foram reportados fraquezas potenciais, e "Keyed MD5" (tipicamente usado para autenticação) foi publicamente quebrado.

MD2, MD4 são algoritmos *dhash* mais antigos. Foram detectadas falhas de modelagem e seu uso não é recomendado. O código fonte está disponível no site citado nas referências bibliográficas.

O mais recente algoritmo *hash* é o RIPEMD-160, projetado para substituir o MD4 e MD5. Produz um cabeçalho de 20 *bits*, executando documentos a 40Mb/s em um pentium 90MHz, disponível em domínio público, ver referências bibliográficas. (Menezes, et. al., 1996)

SHA (SHS)

Algoritmo de *hash* criptografado publicado pelo Governo dos Estados Unidos. Produz um valor de 160 *bits* de comprimento arbitrário. Por ser um algoritmo considerado recente, não se tem nenhuma publicação de quebras no algoritmo e portanto considerado seguro.

3.2 PROTOCOLOS SEGUROS

Um protocolo é uma série de passos, envolvendo um ou mais participantes, objetivando executar uma tarefa. A escolha de um protocolo adequado e eficiente é fundamental para a segurança dos dados que são compartilhados durante as transações bem como do resultado das tarefas realizadas pelas aplicações.

SSL

Segurança das transações ou comunicação segura referem-se a habilidade de duas entidades na Internet para conduzir uma transação privada e com autentificação e assinaturas digitais. Comércio na Internet e WWW fundamentalmente dependem de transações confiáveis.

O SSL é um protocolo para segurança de transações na WWW, seu principal objetivo é fornecer privacidade e confiabilidade entre duas aplicações que se comunicam entre si, seu projeto prevê a existência de diversos tipos de aplicações e de sistemas operacionais e procura estabelecer um processo de negociação e emprego de funções de autenticação mútua, criptografia de dados e integridade para transações na Internet.

Trabalha no nível mais alto de qualquer protocolo de transporte como o TCP, mas não é dependente do TCP, também executa sobre protocolos de aplicações como o HTTP e FTP.

É composto de duas camadas: no nível inferior suportado pelo protocolo de transporte (como o TCP), fica o *SSL record protocol* usado para encapsular os diversos protocolos de nível superior e prover os serviços de fragmentação, compressão, autenticação de mensagem e criptografia; no nível superior além do protocolo de aplicação, são inseridos protocolos auxiliares como o *SSL Handshake Protocol*, o *Change Cipher Spec Protocol* e o *Alert Protocol*.

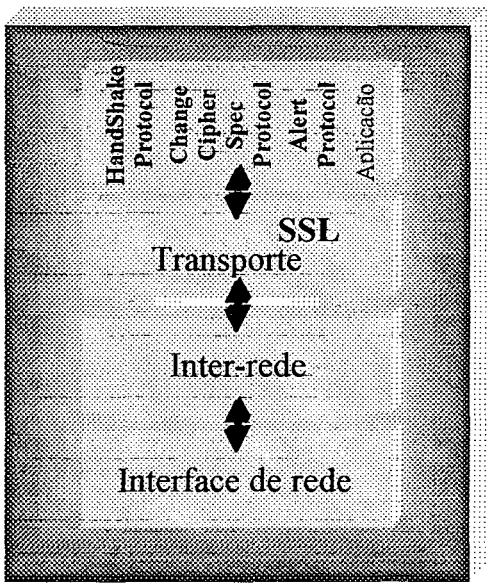

FIGURA 8. SSL Arquitetura

Dois conceitos importantes para entender o funcionamento do SSL são sessão e conexão:

- Sessão: é uma associação entre um cliente e um servidor. São criadas pelo *Handshake protocol*. Sessões definem um conjunto de parâmetros de segurança, que são compartilhados entre as múltiplas conexões. São utilizadas para garantir a negociação de novos parâmetros de segurança para cada conexão.

- Conexão: é o meio de transporte. As conexões possuem relacionamentos ponto-a-ponto. As conexões são transitórias. Cada conexão está associada a uma sessão.

A comunicação é através do estabelecimento de uma sessão, caracterizado por um Estado de Sessão e um Estado de conexão. Entre qualquer par de comunicação, podem existir múltiplas conexões seguras. Em teoria, também podem existir múltiplas sessões simultâneas entre aplicações, mas esta característica não é usada na prática.

Estado de Sessão

- *Session Identifier*: sequência arbitrária de *bytes* escolhida pelo servidor para identificar um estado de sessão ativo ou resumível.
- *Peer Certificate*: certificado X.509.v3 para o *peer*. Pode ser nulo.
- *Compression Metod*: o algoritmo para compressão dos dados antes da criptografia.
- *Cipher Spec*: especifica o algoritmo de criptografia (pode ser null, DES, RC4, RC2 etc) e um algoritmo de *hash* (MD5 ou SHA-1) usado pelo MAC.
- *Master Secret*: chave secreta de 48 *bits* compartilhada entre cliente e servidor.
- *Is Resumable*: flag que indica se a sessão pode ser utilizada para novas conexões.

Estado de Conexão

- *Server and Client random*: sequencia de *bytes* que são escolhidos pelo servidor e cliente para cada conexão.
- *Server write MAC secret*: chave secreta usada nas operações MAC nos dados enviados pelo servidor.
- *Client write MAC secret*: chave secreta usada nas operações MAC nos dados enviados pelo cliente.
- *Server write key:* chave convencional de criptação usada para criptografar dados enviados pelo servidor e descriptografar pelo cliente.

- *Client write key:* chave convencional de criptação usada para criptografar dados enviados pelo cliente e descriptografar pelo servidor.
- *Initialization vectors*: quando um bloco cifrado é usado, o *initialization vectors(IV)* é mantido para cada chave. Este campo é primeiramente inicializado pelo *SSL Handshake Protocol*. Depois disso, o bloco final de texto cifrado de cada registro é preservado para usar com o IV do registro seguinte.
- *Sequence numbers*: cada parte mantém sequência separada de números para transmitir e receber mensagens de cada conexão. Quando uma parte envia ou recebe a *change cipher spec message*, a sequência de números apropriada é setada para zero.

Record Protocol

O *record protocol* pega uma mensagem da aplicação para transmitir, fragmenta os dados em blocos, opcionalmente comprime os dados, aplica MAC, criptografa, adiciona um cabeçalho, e transmite a unidade resultante em um segmento TCP. Os dados recebidos são descriptografados, verificados, descomprimidos, agrupados os blocos e entregue ao usuário final.

A figura 9 descreve a operação global do *record protocol*.

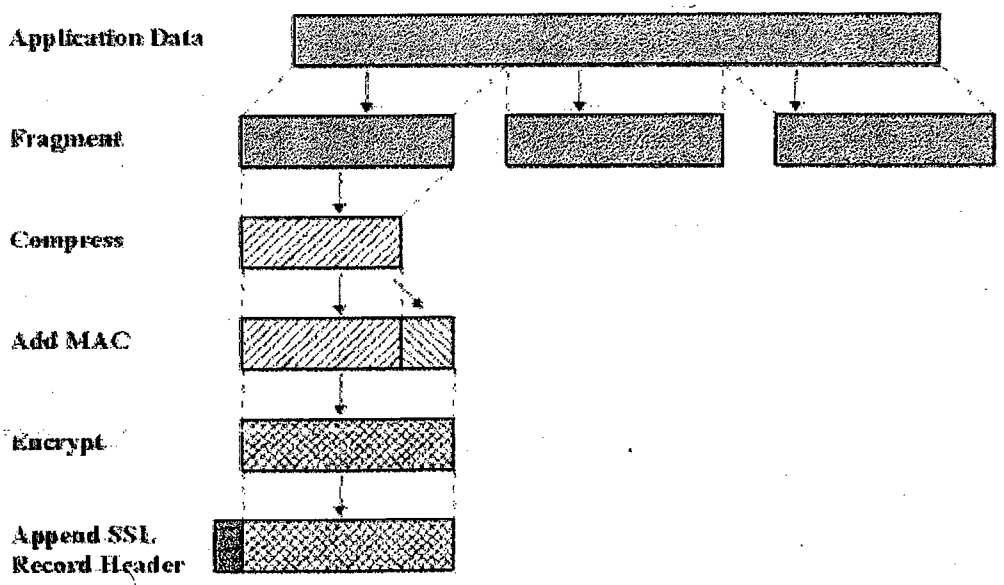

FIGURA 9. Operação do SSL Record Protocol
Fonte: (WILLIAM, 1998)

- Fragmentação: fragmenta os blocos de informação em registros *SSLPlaintext* de 214 ou menos.
- Compressão/descompressão: transforma os registos *SSLPlaintext* em um registro *SSLCompressed* e vice-versa, através do algoritmo definido no Estado de Sessão. Por padrão nenhum algoritmo é definido.
- Autenticação da mensagem (MAC): à chave secreta compartilhada é usada num algoritmo de *hash* (MD5 ou SHA-1).
- Criptografia: a mensagem após passar pelos passos anteriores, é então criptografada usando um sistema simétrico. Os seguintes sistemas são permitidos: DES(56*bits*), 3DES(168*bits*), Fortezza(80*bits*), DES-40(40*bits*), RC2-40(40*bits*), IDEA(128(*bits*), RC4-40(40*bits*) e RC4-128(128*bits*). A criptação é feita no bloco inteiro, incluindo o MAC.
- Cabeçalho: último passo do *SSL record protocol*, que consiste dos seguintes campos: *Content type* (protocolo de alto nível usado para processo de fragmentação), *Major version* (número da versão mais recente do SSL), *Minor version* (número da versão mais baixa), *Compressed lenght* (o comprimento em *bytes* dos fragmentos de texto simples (ou do texto comprimido, se foi utilizada compressão)).

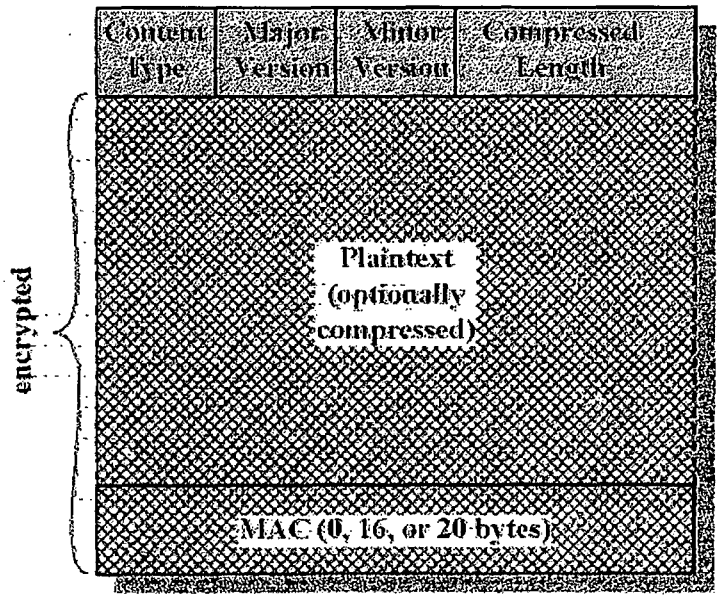

Figura 10. Formato do SSL Record
Fonte: (WILLIAM, 1998)

Change Cipher Spec Protocol

Serve para sinalizar transições nas estratégias de criptografia. Consiste de uma simples mensagem, que consiste de um simples *byte* com o valor 1. Pode ser transmitida pelo cliente e pelo servidor para notificar à outra parte que o registro subsequente será protegido por chaves de criptografia recém negociadas.

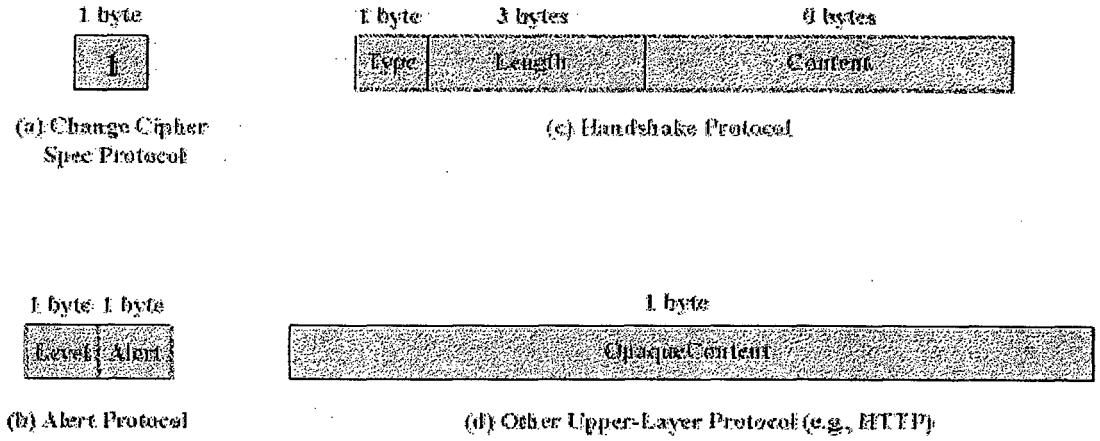

Figura 11. SSL – protocolos da camada de aplicação
Fonte: (WILLIAM, 1998)

Alert Protocol

Supervisiona erros na camada do *Record Protocol* e provê troca de mensagens de alerta para sinalizar erros de sequência de mensagens, erros de certificação e de criptografia.

Cada mensagem neste protocolo consiste em dois *byte*s. O primeiro *byte* tem o valor 1(advertência) ou 2 (fatal) que define o grau de severidade na mensagem. Se o grau é 2(fatal), SSL imediatamente finaliza a conexão. Outras conexões na mesma sessão podem continuar, mas não serão feitas novas conexão até que a sessão esteja estabilizada. O segundo *byte* contém um código que indica o tipo de alerta:

- *unexpected_message*: recebida mensagem não apropriada;
- *bad_record_mac*: recebido MAC incorreto;
- *decompression_failure*: falha na função de descompressão;
- *handshake_failure*: remetente impossibilitado de negociar parâmetros de segurança;
- *ilegal_parameter*: um campo na mensagem *handshake* está fora do intervalo ou está inconsistente;
- *close_notify*: notifica o destinatário que o remetente não enviará mais mensagens nesta conexão;
- *no_certicate*: pode ser enviado em resposta a um pedido de certificado, se nenhum certificado apropriado está disponível.
- *bad_certificate*: o certificado recebido está corrompido(não foi verificada sua veracidade);
- *unsupported_certificate*: o tipo de certificado recebido não é suportado;
- *certificate_revoked*: o certificado não foi reconhecido pela autoridade certificadora;
- *certificate_expired*: o certificado expirou(prazo);
- *certificate_unknown*: algum outro problema encontrado na certificação.

Handshake Protocol

A parte mais complexa do SSL. Este protocolo permite que o servidor e o cliente autentiquem cada parte e negociem a criptografia, o algoritmo MAC e as chaves que serão usadas para proteger os dados enviados pelo SSL *record*. O *handshake protocol* é executado antes de qualquer dado da aplicação ser transmitido.

Consiste em uma série de mensagens trocadas entre o servidor e o cliente. Todas as mensagens possuem três campos:

- Type (1*byte*): indica um dos 10 tipos de mensagens possíveis.
- Length (3 *bytes*): o comprimento da mensagem em *bytes*.
- Content (> 1 *byte*): os parâmetros associados com esta mensagem.

Tipo Mensagem	**Parâmetros**
hello_request	null
client_hello	version, random, session id, cipher suite, compression method
server_hello	version, random, session id, cipher suite, compression method
certificate	chain of X.509v3 certificates
server_key_exchange	parameters, signature
certificate_request	type, authorities
server_done	null
certificate_verify	signature
client_key_exchange	parameters, signature
finished	hash value

QUADRO 7. SSL Handshake Protocol – Tipos de Mensagens
Fonte: (WILLIAM, 1998)

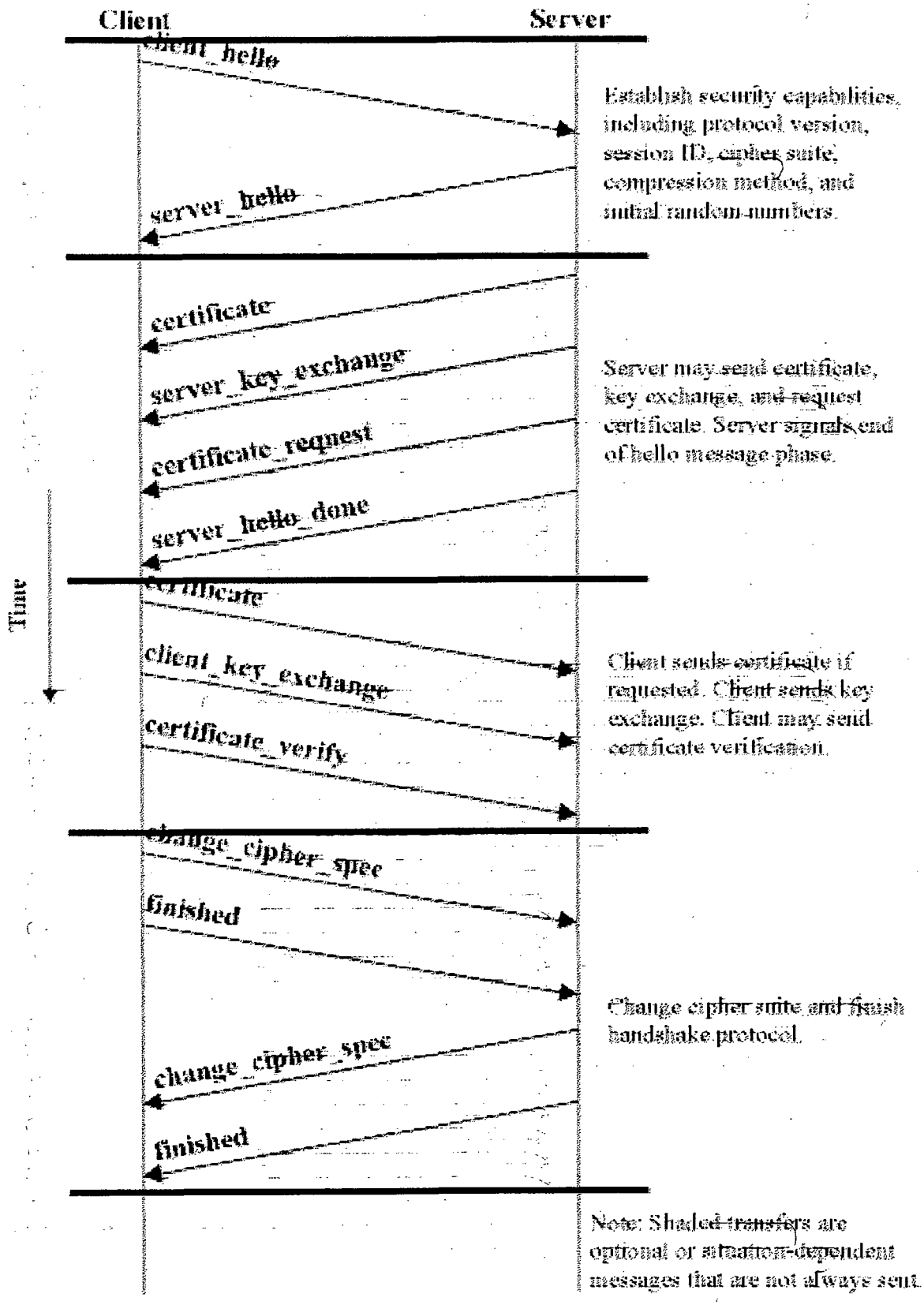

FIGURA 12. Mensagens do Handshake Protocol
Fonte: (WILLIAM, 1998)

Fase 1. Estabelecendo a Segurança

É usada para iniciar a conexão lógica e estabilizar a capacidade de segurança que irá ser associada a ela. A troca é iniciada pelo cliente, que envia a mensagem **client_hello** com os seguintes parâmetros:

- *Version*: a máxima versão do SSL suportada pelo cliente;
- *Random*: estrutura randômica gerada pelo cliente, utilizado durante a mudança de chave para prevenir ataques *replay*.
- *Session ID*: identificador da variável de sessão. Um valor diferente de zero indica que o cliente deseja atualizar os parâmetros da conexão existente ou criar uma nova conexão nesta sessão. O valor zero indica que o cliente deseja estabelecer uma nova conexão em uma nova sessão.
- *CipherSuite*: é uma lista que contém à combinação de algoritmos de criptografia suportados pelo cliente. Cada elemento da lista define uma algoritmo de troca de chave e um algoritmo *CipherSpec*.
- *Compression Method*: lista dos métodos de compressão suportados pelo cliente.

Após enviar a mensagem client_hello, o cliente aguarda pela mensagem **server_hello**, que possui os mesmos parâmetros da client_hello, mas o campo *Version* contém a mais baixa versão SSL suportada pelo cliente e a mais alta suportada pelo servidor; *Random* é gerado no servidor e independe do campo gerado no cliente; *SessionId* se o valor do cliente era diferente de zero, o valor permanece, caso contrário, o campo assume o valor da nova sessão; *CipherSuite* contém o número do algoritmo selecionado pelo servidor para criptografar os dados e; *Compression* possui o número do método de compressão selecionado pelo servidor, dentre os sugeridos pelo cliente.

Os seguintes métodos de criptografia são suportados: RSA, Diffie-Hellman fixo, Ephemeral Diffi-Hellman, Anonymous Diffie-Hellman e Fortezza, *Cipher algorithm* (RC4, RC2, DES, 3DES, DES40; IDEA, Fortezza), MACAlgorithm (MD5 ou SHA-1).

Fase 2. Autenticação do Servidor e chave de troca

O servidor inicia esta fase enviando seu certificado (*certificate*), se ele necessitar ser autenticado a mensagem irá contér um ou uma cadeia de certificadores X.509. A próxima mensagem a *server_key_exchange* será envia se for requerida; ela não é requerida em dois casos: 1. o servidor já encaminhou o certificado com os parâmetros do Diffie-Hellman Fixo, ou 2. à chave do RSA será usada. Para os outros algoritmos será necessário encaminhar a mensagem com os parâmetros específicos de cada método.

O servidor pode agora solicitar ao cliente (se não estiver utilizando o método *Anonymous* Diffie-Hellman) sua certificação. A mensagem *certifica_request* inclui dois parâmetros:*certificate_type* e *certificate_authorities*. O tipo do certificado indica à chave pública do algoritmo em uso e o segundo parâmetro é uma lista de autoridades de certificação.

A mensagem final desta fase, sempre requerida, é a mensagem *server_done*, que pelo servidor encaminha o final da mensagem de *server_hello* e mensagens associadas. O servidor irá esperar pela resposta do cliente. Esta mensagem não possui parâmetros.

Fase 3. Autenticação do Cliente e chave de troca

Após receber a mensagem *server_done*, o cliente deve verificar se o servidor forneceu um certificado válido e validar os parâmetros do *server_hello*. Se os resultados forem aceitáveis, o cliente encaminha uma ou mais mensagens ao servidor. Se o servidor solicitar um certificado, o cliente inicia esta fase

encaminhando a mensagem *certificate*. Se não possuir certificação válida, então encaminha a mensagem de *no-certificate*.

O conteúdo da próxima mensagem de *client_key_exchange* depende do tipo da chave trocada, que pode ser:

- RSA: o cliente gera uma chave privada (*pre-master*) de 48 *bits* e criptografa com à chave publica do certificado do servidor ou uma chave RSA temporária da mensagem *server_key_exchange*. Ela é utilizada mais tarde para formar à chave privada *master* no DSS.
- *Ephemeral or Anonymous* Diffie-Hellman: o cliente encaminha os parâmetros públicos.
- Diffie-Hellman fixo: os parâmetros públicos são encaminhados como uma mensagem certificada, só que o conteúdo da mensagem é *null*.
- Fortezza: os parâmetros do Fortezza são encaminhados.

Fase 4. Finalizando

O cliente encaminha a mensagem *change_cipher_spec* e copia as pendências do *CipherSpec* no atual *Ciher Spec*. Esta mensagem não é considerada parte do *handshake protocol* mas é enviada usando o *Change Cipher Spec Protocol*. O cliente então imediatamente encaminha a mensagem *finished* que verifica se a troca de chave e a autenticidade do processo foram finalizadas com sucesso. O conteúdo desta mensagem é a concatenação de dois valores de *hash*.

Em resposta ao recebimento dos valores, o servidor envia a mensagem proprietária *change_cipher_spec*, que transfere pendências para a atual *CipherSpec*, e envia a mensagem de finalização. Neste ponto o *handshake* foi completado e o cliente e o servidor iniciam as trocas de dados na camada de aplicação.

TLS/PCT

Com a rápida disseminação do SSL, foi necessário estabelecer padrões que fossem utilizados na plataforma desenvolvida pela *Netscape*. A *Microsoft* lançou o PCT que é compatível com o SSL, mas diferente porque tem uma fase de *Handshake* aprimorada que elimina vários bugs encontrados nesta fase do SSL. E também o TLS que é uma iniciativa de padronização da IETF, sendo que seu objetivo é produzir um padrão para a Internet. O TLS é similar ao SSL e basicamente a nível de *record* muda o algoritmo, sendo no TLS é utilizado o HMAC (RFC2104) ao invés do MAC do SSL.

O PCT foi projetado para fornecer segurança na comunicação entre duas aplicações (um cliente e um servidor), autenticar o servidor e (opcionalmente) o cliente. PCT trabalha sobre um protocolo de transporte (por ex. TCP) para transmissão de dados e recepção.

O PCT é independente do protocolo de aplicação. Um protocolo de aplicação (por ex. HTTP, FTP, TELNET etc.) pode ficar sobre o PCT transparentemente. Inicia com uma fase de *Handshake* que negocia o algoritmo criptográfico (simétrico) e à chave de sessão e também a autenticação do servidor para o cliente (e, opcionalmente, vice-versa), baseado em certificado de chaves assimétricas.

Além da criptografia e autenticação, o PCT verifica a integridade das mensagens que usam uma função *hash* código baseada no Código de Autenticação da Mensagem (HMAC).

O formato de registro do protocolo PCT é compatível com o do SSL. Servidores que implementam ambos os protocolos podem distinguir entre clientes PCT e SSL pelo campo do número da versão permanece no mesmo local da primeira mensagem em ambos protocolos.

HTTPS

É a utilização do protocolo HTTP em conjunto com o protocolo SSL que foi desenvolvido para fornecer uma camada de segurança entre à camada de transporte e os protocolos de aplicação como HTTP, Telnet, FTP, SMTP etc.

Pode ser considerado, sob o ponto de vista do *browser* como um protocolo único que é obtido pela associação do protocolo HTTP com SSL, pois é necessário utilizar "https://" para URLs HTTP com SSL, enquanto a URL "http://" continua sendo usada para HTTP sem SSL. A razão principal para utilizar método de acesso diferente é para que *browsers* que não suportam https possam ser impedidos de submeter informações através de formulários que esperam-se sejam seguros, isto é, se um documento veiculado por um servidor HTTP normal contém um campo de formulário para ser preenchido e submetido com uma URL "https" é porque o autor espera que o formulário seja submetido seguramente, um browser sem SSL nem ao menos tentará submeter este formulário pois aparecerá uma mensagem de erro ao usuário. Se a URL não fosse separada, o browser tentaria encaminhar o formulário passando a informação contida nele pela rede sem nenhum tipo de mecanismo de segurança e a submissão falharia do mesmo modo.

S-HTTP (RFC2660/AUG., 1999)

É uma extensão do protocolo HTTP, proposta pelo EIT em 1994 que fornece transações seguras pela incorporação de criptografia e mecanismos de autenticação no protocolo HTTP permitindo segurança em transações fim-a-fim entre cliente e servidor Web.

A especificação do HTTP original possui pouco suporte para mecanismos de criptografia que se destinam para prover segurança nas transações WWW.

O S-HTTP provê mecanismos de comunicação segura entre um cliente-servidor HTTP o que habilita um canal de transações comerciais ou um largo canal de alcance de aplicações. Foi modelado para ser um protocolo flexível com modos de operação, mecanismos de administração chaves, modelos de confiança, algoritmos de criptografia e formatos de encapsulamento por opção de negociação entre as partes da transação.

Características

É um protocolo seguro de comunicações orientado a mensagem, projetado para uso junto com HTTP. Foi modelado para coexistir com o modelo de mensagens do HTTP e ser facilmente integrado com suas aplicações.

Fornece uma variedade de mecanismos de segurança para os seus clientes e servidores, garantindo opções de serviço de segurança apropriados para à grande potencialidade da WWW.

O protocolo provê simetria para o cliente e servidor (e mesmo tratamento para solicitações e respostas, como também para preferências de ambas as partes) enquanto preserva o modelo de transação e características de implementação do HTTP.

Podem ser incorporados vários padrões de formato de mensagens criptografadas nos clientes de S-HTTP e servidores, particularmente, mas a princípio não limitado. Clientes atentos S-HTTP podem comunicar-se com S-HTTP servidores inconscientes e vice-versa, embora tais transações não necessitem utilizar as características de segurança.

S-HTTP não requer cliente-lado de certificados de chave pública (ou chaves públicas), suporta modos de operação simétricos de chave única. Significa que transações privadas podem acontecer sem exigir dos usuários que tenham uma chave pública estabelecida.

Suporta transações seguras fim-a-fim, em contraste com o mecanismo de autenticação original do HTTP que requer que o cliente tente acesso e seja negado antes do mecanismo de segurança ser empregado.

Extremamente flexível, possui grande quantidade de algoritmos criptográficos, modos e parâmetros. A negociação de opções é usada para permitir aos clientes e servidores que concordem em modos de transação (por exemplo, se o pedido deveria ser assinado, criptografado ou ambos); algoritmos criptográficos (RSA ou DSA para assinatura, DES ou RC2 por criptografar etc.); e seleção do certificado.

3.3 SET

Modelado para proteger transações que utilizem cartões de crédito na Internet. A atual versão, SETv3, surgiu da necessidade de administradoras de cartões de crédito (1996) que desejavam garantir a integridade dos seus clientes e das compras por eles realizadas. Grande empresas trabalharam na especificação incial: Microsoft, IBM, Netscape, RSA, Terisa and VeriSign.

SET não é um sistema de pagamento. É um conjunto de protocolos de segurança e formatos que habilitam o usuário a utilizar o seu cartão de crédito.

Fornece basicamente três serviços:

- Mantém seguro o canal de comunicação entre todas as partes envolvidas no processo.
- Utiliza certificação digital (X.509).
- Garante privacidade das informações disponíveis para as partes em uma transação, quando e onde for necessário.

Avaliação

Inicialmente precisamos analisar o comércio e os requerimentos para o SET, ferramentas de chave, e participantes nas transações.

A especificação de requerimentos para que o usuário confie no sistema e realize um pagamento seguro com seus cartões de crédito na Internet e em outras redes:

- Confidencialidade do pagamento e informação ordenada: confidencialidade reduz os riscos de fraudes ou acessos de terceiros mal-intencionados. Utiliza para isso criptografia.
- Garantir a integridade dos dados transmitidos: isto é, garantir que não existam mudanças nos dados durante a transmissão. Assinaturas digitais são utilizadas para manter integridade.
- Prover autenticação de *cardholder* para um usuário legítimo de conta de cartão de crédito: reduz a incidência de fraudes e altos custos de processamento. Assinaturas digitais e certificados são usados para autenticação.
- Fornecer autenticação entre a *merchant* e a instituição financeira.
- Assegurar o uso das melhores práticas de segurança e técnicas de modelagem de sistemas para proteger todas as partes legítimas em uma transação comercial eletrônica: SET é especificado num alto nível de algoritmos criptográficos e protocolos.
- Criar um protocolo que não dependa de mecanismos de segurança no transporte: SET pode operar sobre um protocolo de transporte, mas não interfere com o uso de seus mecanismos de segurança, como um IPSec e SSL/TLS.

Características

- **Confidencialidade das informações**: *cardholder account* e informações de pagamento estão seguras quando trafegam na rede. SET previne o comércio de número de cartões de crédito utilizando criptografia (DES).
- **Integridade dos Dados**: informações de pagamento enviadas para a loja incluem dados da ordem, dados pessoais e instruções de pagamento. SET garante que esta mensagem não será modificada no seu trânsito, usando RSA assinaturas digitais, e SHA-1 , que provê a integridade da mensagem.
- **Autenticação da *Cardholder account***: habilita a loja para verificar se o *cardholder* é usuário legítimo de um cartão de crédito (número). SET utiliza certificados X.509v3 com assinaturas RSA.

Note que ao contrário de protocolos como IPSec e SSL/TLS, SET oferece só uma escolha para cada algoritmo de criptografia. Isto faz sentido, porque SET é uma aplicação simples de avaliar, considerando que IPSec e SSL/TLS são para suporte de outras aplicações.

QUARTO CAPÍTULO
AUTENTICAÇÃO

"A história deve ser reescrita a cada nova geração porque embora o passado não mude, o presente está sempre em mudança; cada geração tem novas perguntas a fazer ao passado e encontra novas áreas de simpatia em que faz reviver diferentes aspectos de seus predecessores"

(Christopher Hill)

As técnicas de autenticação podem ser muito simples, muito complexas ou ambas. Incluem equipamentos e aplicativos que associam os recursos ao usuário e determinam os direitos de acesso que ele possui, certificação para execução de serviços e prova de identidade digital.

4.1 AUTENTICAÇÃO

Existem vários procedimentos para prover autenticação em um ambiente distribuído. O primeiro é verificar ou autenticar a identidade de um usuário. Existem três métodos básicos: 1) algo conhecido pelo usuário - o uso de senhas de acesso, sendo o mais comumente utilizado, mas não o mais seguro; 2) algo de posse do usuário – como o uso de chaves e 3) algo do próprio usuário – como impressões digitais e padrões de retina, que é o método mais seguro e dispendioso. Todos métodos conhecidos e viáveis num sistema distribuído.

Este processo pode ser alvo de usuários indevidos que submetem informações que podem confundir sistemas de autenticação e forjar acesso "autorizado". Os ataques mais comuns envolvem:

- *Eavesdropping*: intercepção de informações;
- Gerenciamento de senhas múltiplas: armazenagem do Id do usuário e senha em sistemas com banco de dados que mantêm informações de autenticações, ou mesmo sistemas que armazenam localmente estas informações em qualquer arquivo.
- *Replay*: cópia de informação de autenticação enquanto é transmitida na rede, até mesmo se é codificada, e utiliza para ter permissão de acesso;

Uma solução para estes problemas é o uso de sistemas de gerenciamento de certificados. O certificado é gerado por uma autoridade certificadora, que verifica frequentemente seu prazo de validade e autentica os pacotes que trafegam, permitindo acesso fácil a vários recursos. Pode ser agregado com um autenticador, que é um valor randômico e único, gerado para cada comunicação. Assim, nenhuma comunicação compartilha o mesmo autenticador, evitando assim o *Replay*. Há duas maneiras básicas para gerenciar certificação: a primeira envolve o uso de listas de certificados que possuem certificados de autoridades de certificação confiáveis como é exemplificado pela X.509, onde qualquer serviço solicitado deve antes examinar a lista e verificar a autenticidade da identidade do cliente. A segunda é um centro de distribuição de certificado, onde os clientes obtêm um certificado para cada serviço que desejem utilizar. O cliente apresenta o certificado ao serviço quando for utilizá-lo.

Listas de Certificado

Estão baseadas em criptografia de chave pública. As autoridades de autentificação verificam as identidades dos usuários e suas chaves públicas. Os certificados são autenticados por uma autoridade de certificação de assinatura digital.

X.509

É um serviço de diretórios que provê a localização das listas de certificado e assume que existe uma autoridade de certificação confiável para criar estes certificados. O certificado é assinado pelo usuário (criptografado com sua chave privada) que é o proprietário do certificado e de sua chave pública. Elementos contidos em um certificado X.509:

- *V (Version)* versão do X.509. O *default* é a versão de 1988.
- *SN (Serial Number)* o número de série que é um valor inteiro e único dentro da autoridade certificadora.
- *AI (Algorithm Identifier)* identifica o algoritmo usado para assinar os certificados da autoridade certificadora. A autoridade certificadora assina cada certificado com sua chave privada.
- *CA (Issuer or certificate authority)* cria os certificados.
- *T$_{A}$ (Period of validity)* período de validade.
- *A (Subject)* identidade verificada pelo certificado
- *Ap (Public-key information)* chave pública e identificação do algoritmo para o certificado
- *Signature* assinatura do certificado. É um código *hash* de todos os elementos e é criptografada com à chave privada da autoridade certificadora o que assegura a integridade destas informações.

Qualquer usuário de posse da chave pública da autoridade certificadora pode descriptografar e verificar a autenticidade de cada certificado na lista de certificado. O X.509 possui três procedimentos para autenticação usando a lista de certificado: autenticação *one-way*, autenticação de *two-way* e autenticação de *three-way*.

- *One-way* protege a integridade da mensagem. O remetente criptografa a mensagem utilizando sua chave privada. O destinatário a descriptografa usando à chave pública original e verifica na lista de certificado. Só então o usuário é autenticado. A mensagem utiliza marcas de tempo (*timestamp* - marca o tempo e hora da execução do processo no envio da mensagem) garantindo que esta mensagem não esteja sendo repetida de forma indevida.
- *Two-way* permite verificar o remetente e o destinatário. Em adição ao modo *one-way* faz também a autenticação do destinatário. A mensagem utiliza um *timestamp* tanto no processo do remetente quanto no do destinatário.
- *Three-way* empregado quando o destinatário e o remetente não têm relógios sincronizados ou não desejam confiar nos relógios. Além de passar por autenticação *two-way*, o remetente envia ao destinatário uma cópia da cópia recebida e valida as informações que chegam até ele sem a necessidade de verificar o tempo dos processos.

Classes de Certificado

O X.509 fornece diferentes classes de certificados. Cada classe possui um grau diferente de verificação que reflete o nível de confiabilidade da informação autenticada nas listas de certificado.

O nível mais baixo é Classe 1. Para um certificado classe 1, a identidade é verificada pelo assunto da mensagem de um endereço de *e-mail*, e a autoridade envia uma resposta a este endereço. A classe de nível mais alta, classe 4, que requer apresentação física do assunto e a autoridade de certificado tem que verificar o segundo plano da mensagem. Podem ser empregados certificados X.509 dentro do protocolo SSL.

Centro de Distribuição de Certificado

Fornece um local confiável para administrar a distribuição de todos os certificados. Assim, o centro de distribuição de certificado se torna um elemento crítico no sistema distribuído. Este local mantém uma cópia das chaves privadas de todos os usuários válidos e serviços do sistema no "gerente de banco de dados" de certificado. O centro de distribuição de certificado usa estas chaves privadas para distribuir os certificados de serviço para os usuários quando eles precisam utilizá-los. Estes certificados devem ser apresentados pelo cliente para usar um serviço. Cada certificado de serviço é único para o usuário e o serviço. Certificados de serviço são só válidos para um tempo limitado e um serviço particular. Se o certificado de serviço de um usuário expirou, o usuário tem que obter um novo certificado de serviço no centro de distribuição de certificado.

Kerberos

Kerberos foi desenvolvido pelo MIT e é considerado o primeiro protocolo de autenticação distribuído em uso difundido. É um sistema de autenticação feito em três partes, pois utilizando uma terceira parte para armazenamento das informações (banco de dados) libera as outras partes de manterem localmente uma estrutura de armazenamento. Ao invés disso, mantém um único banco de dados *master* com informações de autenticação denominado *Kerberos Database Management System* (KDBM). Este KDBM é atualizado por diversos *Key Distribution Services* (KDS) que possuem cópias (*read-only*) do banco de dados *master* ajudando-o assim a evitar uma sobrecarga e a utilizar o sistema de autenticação se o KDBM não estiver disponível. Trabalha em quatro fases, como descrito na sequência:

Fase 0 – Registro do Usuário

Antes de estabelecer uma sessão, o usuário deve se cadastrar (registrar) em um KDC. Uma vez feito a seu cadastro, o Id de usuário e senha são armazenados criptografados no *Kerberos Database Manager System*(KDBM). Neste ponto usuário é considerado registrado e está preparado para utilizar os serviços disponíveis no protocolo Kerberos.

Fase 1 – Obtendo um *Ticket*

A certificação da autenticação é chamada de *ticket*. A fase 1 é solicitar um *ticket* que pode ser usado para conseguir um *ticket* de serviço de um *Ticket Granting Service* (TGS) na fase 2. O *ticket* obtido é criptografado com à chave privada do TGS. O *ticket* contém informação de identificação para o cliente, como: chave privada de sessão temporária para criptografar as mensagens entre o TGS e o cliente, a identidade do cliente, a identidade do TGS, tempo de vencimento do *ticket*, o endereço na rede do cliente. O *ticket* é criptografado com à chave privada do TGS, de forma que só o TGS pode verificar se ele é válido quando receber o *ticket* do cliente, pois somente o TGS e o Kerberos conhecem à chave.

Quando um cliente encaminha um *ticket* com o intuito de solicitar um serviço, ele envia também um autenticador, cujo objetivo é evitar que uma entidade armazene o *ticket* e o utilize posteriormente em ataques do tipo *replay*. O autenticador é criptografado com à chave de sessão.

Kerberos *Authentication Service* V5

IDT = Ticket Granting Service's ID
IDc = Client's ID
IDs = Server's ID
Ni = Nonce value
Kc = Client's private-key
Ks = Application Server's ID
KT = Ticket Granting Server's private-key

K1 = System Ticket
K2 = Service Ticket
Ts = Starting time stamp
Te = Ending time stamp
E(a,K) = Applications of encryption algorithm to a with key K.

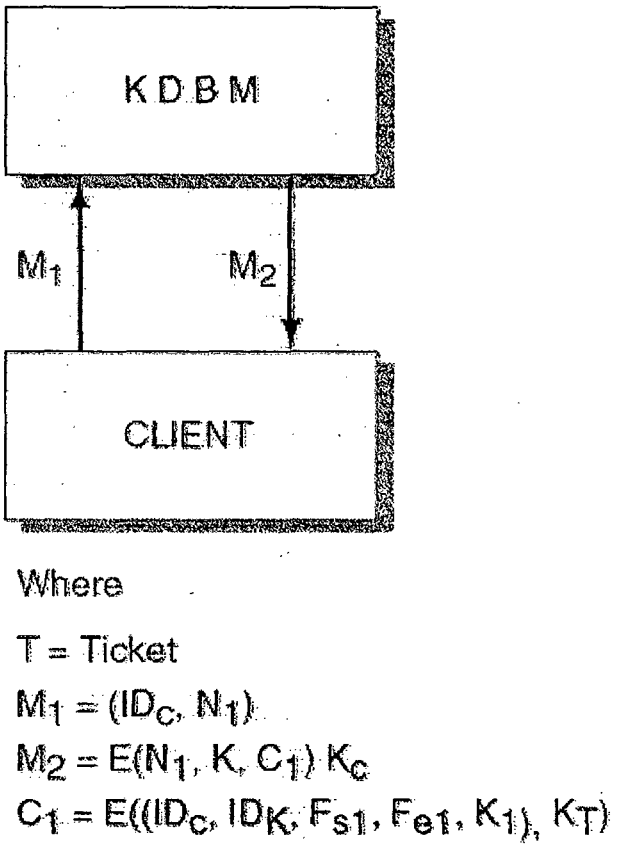

FIGURA 13. Kerberos - Detalhes Fase 1.
Fonte: (GALLI, 1999)

Fase 2 – Obtendo um *ticket* de serviço

Os *tickets* para serviços registrados no Kerberos são obtidos através do envio de uma solicitação (criptografada com à chave de sessão) ao TGS, que inclui o nome do cliente, endereço na rede, autenticador e o nome do serviço que o cliente deseja utilizar. O TGS após verificar a solicitação (descriptografar) prepara um pacote para o cliente. Este pacote inclui o *ticket* de serviço criptografado com à

chave privada do serviço (somente o serviço poderá verificar o *ticket*), o nome do serviço, a validade do *ticket*, e o autenticador. Este pacote é criptografado com a chave de sessão que é compartilhada com o cliente e o serviço.

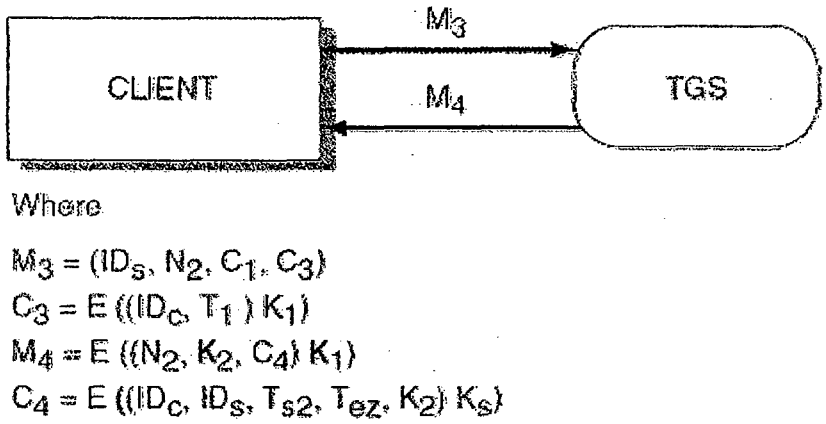

FIGURA 14 . Kerberos - Detalhes Fase 2
Fonte: (GALLI, 1999)

Fase 3. Usando o Serviço

Nesta fase o cliente que possui o *tichet* de serviço está apto a utilizá-lo. O *ticket* permitiu que o serviço autenticasse a identidade do cliente. Para usar o serviço, o cliente enviou um pacote com o *ticket* de serviço e sua identidade. Se o *ticket* foi considerado válido (após a descriptografia) e está dentro do prazo permitido, o cliente está autorizado para usar o serviço solicitado.

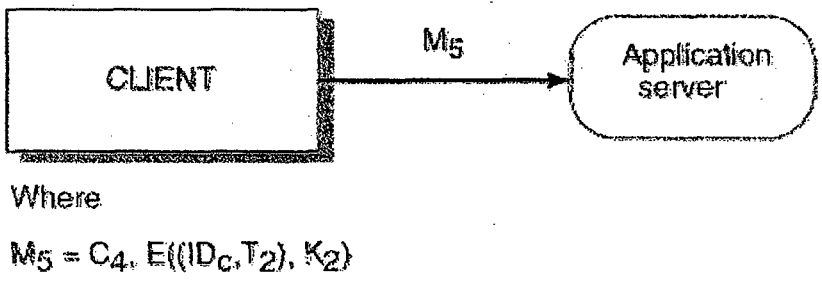

FIGURA 15. Kerberos - Detalhes Fase 3
Fonte: (GALLI, 1999)

O lado prático de Kerberos é sua integração com o nível de aplicação. Aplicações típicas como FTP, Telnet, POP e NFS têm sido integradas com o sistema Kerberos. A última versão avaliada Kerberos 5 *Release* 1. *Patchlevel* 1.1 tem pleno funcionamento em diversas plataformas, entre elas: Digital Unix (OSF/1) 3.2, Digital Unix (OSF/1) 4.0, HPUX 10, FreeBSD 2.1, (i386) NetBSD 1.1, (i386) NetBSD 1.2, (m68k) NetBSD 1.2, (sparc) NetBSD 1.2, (i386) Linux 2.x, Ultrix 4.2, Irix 5.3, AIX 3.2.5, SunOS 4.1, Solaris 2.4, Solaris 2.5.1. As plataformas de SO Windows e Windows NT não estão com pleno funcionamento pois foram relatados problemas de conflitos com aplicações 32*bits*.

4.2 IDENTIFICAÇÃO DIGITAL

Identificações Digitais são o equivalente eletrônico de carteiras de identidade, passaportes e cartões de associados. Você pode apresentar uma Identificação Digital eletronicamente para provar sua identidade ou seu direito a acessar informações ou serviços on-line.

Identificações Digitais, também conhecidas como certificados digitais, vinculam uma identidade a um par de chaves eletrônicas que pode ser usado para criptografar e assinar informações digitais. Uma Identificação Digital torna possível verificar o direito de alguém utilizar determinada chave, ajudando a evitar a falsificação de chaves para uso por impostores.

Conforme consagrado internacionalmente, as chaves de identificação são concedidas por Autoridades Certificadoras ou *Certification Authorities*. As autoridades certificadoras, em regra, são empresas privadas encarregadas de averiguar a identidade de pessoas para fins de emissão de uma espécie de identidade eletrônica, no intuito de possibilitar a realização de operações identificadas nas redes de computadores.

No Brasil, uma das autoridades certificadoras em atividade denomina-se Certisign. A Certisign, para fins de identificação de pessoas em meio eletrônico, seguindo práticas internacionais, faz uso das presunções inerentes aos registros públicos.

Explicando melhor (procedimento de emissão de um certificado digital): inicialmente a Certisign mantém um contrato para a emissão de assinaturas digitais registrado em um ofício do registro de títulos e documentos, com o fim de dar publicidade do mesmo para terceiros quaisquer. A pessoa que desejar receber uma certificado digital, deverá, de início, aderir a esse contrato.

Um vez que a pessoa interessada manifeste junto a companhia a vontade de receber um certificado digital de identificação, a Certisign enviará, via correio, um termo de adesão ao seu contrato padrão (onde se encontra consignado o número de identificação digital a ser utilizado em meio eletrônico) antes citado e um requerimento: o interessado deverá se dirigir a um oficio de notas (versão geral ou *general label*) para o reconhecimento de firma por autenticidade (não confundir com reconhecimento por semelhança) da assinatura aposta no termo de adesão, bem como tirar cópia autenticada de alguns documentos de identificação como, v.g., R.G., CPF, comprovante de residência etc.

Uma vez concluída tal tarefa, o interessado deverá enviar via correio o termo e as cópias acima relacionadas para CertiSign, que se encarregará de levar tais documentos para registro junto a um ofício de registro de títulos e documentos. A finalidade desse novo registro é dar publicidade para a relação jurídica existente entre a empresa que realizará a identificação em meio digital (autoridade certificadora) e o interessado, bem como consignar, publicamente, que o número do certificado presente no termo de adesão corresponde a uma pessoa determinada.

Após a realização do procedimento acima, a Autoridade Certificadora enviará para o interessado à chave que o identificará em meio eletrônico. Nesse sentido, quando duas pessoas identificadas em meio eletrônico pela Autoridade Certificadora iniciarem a troca documentos, utilizando-se, para tanto, das chaves de criptografia, ambas, de forma prévia, poderão verificar o número do certificado uma

da outra (é impossível a emissão de dois certificados iguais, a vinculação do certificado com a pessoa identificada é de caráter personalíssimo - as autoridades certificadoras, em regra, garantem tal condição e assumem expressamente essa responsabilidade). Sabedoras da segurança da tecnologia empregada, bem como da peculiaridade de que o número contido no certificado digital corresponde, exclusivamente, a uma pessoa determinada, com registro em assento público, ambos os interlocutores terão a seu favor uma presunção *iuris tamtum* de que os certificados empregados para o estabelecimento da comunicação devem estar sendo utilizados por pessoas cujos dados foram registrados no ofício de títulos e documentos, estando aptas, portanto, para realização de operações eletrônicas de forma identificada. (LIMA NETO, 1999)

Uma Identificação Digital normalmente contém:

- A chave pública do proprietário
- nome do proprietário
- A data de vencimento da chave pública
- Nome do emissor (a AC que emitiu a Identificação Digital)
- número de série da Identificação Digital
- A assinatura digital do emissor

O formato mais aceito de Identificações Digitais é definido pelo padrão internacional CCITT X.509; assim, certificados podem ser lidos ou escritos por qualquer aplicativo compatível com X.509.

Utilização das assinaturas digitais

Identificações Digitais podem ser usadas para diversas transações eletrônicas, incluindo troca de mensagens, comércio eletrônico, *groupware* e transferência eletrônica de fundos. O popular *Commerce Server da Netscape* exige uma Identificação Digital para cada servidor seguro.

Por exemplo, o cliente de um shopping eletrônico que utilize o software do servidor *Netscape* solicita a Identificação Digital desse servidor para autenticar a identidade do operador do shopping e o conteúdo fornecido pelo comerciante. Sem autenticar o servidor, o comprador não deve confiar ao operador ou comerciante informações críticas como um número de cartão de crédito. A Identificação Digital é útil para estabelecer um canal seguro para a comunicação de informações importantes ao operador do shopping.

Shoppings virtuais, serviços bancários e outros serviços eletrônicos têm se tornado cada vez mais comuns, oferecendo a conveniência e flexibilidade do funcionamento ininterrupto direto de sua casa. No entanto, suas preocupações sobre privacidade e segurança podem evitar que você aproveite essa nova mídia para seus negócios pessoais. Apenas a criptografia não é suficiente, pois ela não fornece uma prova da identidade do remetente das informações criptografadas. Sem garantias especiais, você se arrisca a que alguém se passe por você on-line. Identificações Digitais consideram esse problema, fornecendo um meio eletrônico de verificação de identidades. Usadas com a criptografia, as Identificações Digitais fornecem uma solução de segurança mais completa, garantindo a identidade de todos os envolvidos em uma transação.

Da mesma forma, um servidor seguro deve ter sua própria Identificação Digital, para garantir aos usuários que é executado pela organização à qual afirma ser afiliado e que seu conteúdo é legítimo.

Identificações Digitais usam técnicas de criptografia de chave pública que utilizam duas chaves relacionadas, uma chave pública e uma chave privada. Na criptografia de chave pública, ela é disponibilizada para qualquer pessoa que deseje se corresponder com o proprietário do par de chaves. à chave pública pode ser usada para verificar uma mensagem assinada com à chave privada ou para criptografar mensagens que só possam ser decodificadas com à chave privada. A segurança das mensagens criptografadas dessa forma depende da segurança da chave privada, que deve ser protegida contra uso não-autorizado.

Em uma Identificação Digital, um par de chaves é vinculado a um nome de usuário e outras informações de identificação. Quando instalada em um navegador da Web, uma Identificação Digital funciona como uma credencial eletrônica que os sites podem verificar. Isso permite que elas substituam caixas de diálogo de senhas para informações ou serviços que exijam associação ou que restrinjam acesso a usuários específicos.

Uma Identificação Digital é assinada pela Autoridade de Certificação que a emitiu. Várias Identificações Digitais podem ser anexadas a uma mensagem ou transação, formando uma cadeia de certificados em que cada Identificação Digital confirma a autenticidade da anterior. A autoridade de certificação de nível superior deve ser independente e ter a confiança do destinatário.

A autenticação permite ao destinatário de uma mensagem digital confiar na identidade do remetente e na integridade da mensagem.

Assinatura Digital

Uma assinatura digital funciona para documentos eletrônicos como uma assinatura à mão para documentos impressos. A assinatura é um elemento de dados que não pode ser falsificado e que garante que uma determinada pessoa escreveu ou de alguma forma concordou com o documento ao qual a assinatura está anexada.

Na verdade, uma assinatura digital permite um grau maior de segurança do que uma assinatura à mão. O destinatário de uma mensagem assinada digitalmente pode verificar se o remetente é aquele cuja assinatura está anexada e se o conteúdo não foi alterado de forma intencional ou acidental depois de a mensagem ter sido assinada. Além disso, assinaturas digitais seguras não podem ser repudiadas; o signatário de um documento não pode renegá-lo alegando que a assinatura foi falsificada.

Em outras palavras, Assinaturas Digitais permitem a "autenticação" de mensagens digitais, garantindo ao destinatário a identidade do remetente e a integridade da mensagem.

Utilização da Assinatura Digital

Quando se recebe mensagens assinadas digitalmente, pode-se verificar a Identificação Digital do signatário para determinar se não ocorreu nenhuma falsificação ou adulteração.

Quando se envia mensagens, pode assiná-las e anexar sua Identificação Digital para garantir ao destinatário que a mensagem foi realmente enviada pelo remetente. Várias Identificações Digitais podem ser anexadas a uma mensagem, formando uma cadeia hierárquica, onde uma Identificação Digital garante a autenticidade da anterior. No fim de uma hierarquia de Identificações Digitais, há uma Autoridade de Certificação superior, que não precisa de uma Identificação Digital de qualquer outra autoridade para ser confiável. à chave pública da Autoridade de Certificação superior deve ser conhecida de forma independente, como por exemplo por sua divulgação irrestrita. Quanto melhor você conhecer o destinatário da mensagem, menor é a necessidade de anexar uma Identificação Digital.

Você também pode usar uma Identificação Digital para se identificar perante servidores seguros, como servidores da Web baseados em associação.

Geralmente, depois de obter uma Identificação Digital, você pode configurar seu aplicativo de correio eletrônico ou navegação na Web com funções de segurança para usar a Identificação Digital automaticamente.

Imagine que um usuário "A" deseja enviar uma mensagem assinada a um outro usuário "B". Ele cria uma síntese da mensagem usando uma função *hashing* na mensagem. A síntese funciona como uma "impressão digital" da mensagem; se qualquer parte for modificada, a função retorna um resultado diferente. Em seguida, ele criptografa a síntese da mensagem com sua chave privada. Essa síntese criptografada está na assinatura digital da mensagem.

Então ele envia a mensagem e a assinatura digital para usuário "B". Quando "B" as recebe, decodifica a assinatura com à chave pública de "A", revelando assim a síntese da mensagem. Para verificar a mensagem, ele utiliza nela a mesma função de *hashing* usada por "A" e compara o resultado com a síntese

recebida de "A". Se forem idênticas, "B" pode confiar que a mensagem veio de "A" e que não foi alterada desde que ela a assinou. Se as sínteses não forem iguais, a mensagem teve uma origem diferente ou foi alterada depois de ter sido assinada.

Observe que usar uma assinatura digital não criptografa a mensagem. Se "A" desejar garantir a privacidade da mensagem, deverá também criptografá-la com à chave pública de "B". Depois, apenas "B" poderá ler a mensagem, decodificando-a com sua chave privada.

Não é possível que alguém localize uma mensagem cujo resultado de *hashing* seja um valor determinado ou que existam duas mensagens cujo resultado de *hashing* seja idêntico. Se isso fosse possível, um intruso poderia anexar uma mensagem falsa à assinatura de "A". Funções de *hashing* específicas foram criadas para que seja inviável descobrir um resultado idêntico, e dessa forma essas funções são consideradas adequadas para uso em criptografia.

Uma ou mais Identificações Digitais podem acompanhar uma assinatura digital. Se não houver uma Identificação, o destinatário (ou terceiro) pode verificar a autenticidade da chave pública.

Prazo de Validade de uma Assinatura Digital

Normalmente, uma chave vence após um determinado período, como um ano, e um documento assinado com uma chave vencida não pode ser aceito. No entanto, há muitos casos em que é necessário que documentos assinados sejam considerados como legalmente válidos por um período muito superior a dois anos; exemplos são arrendamentos de longo prazo e contratos. Ao registrar o contrato com um serviço de selo de data/hora digital no momento de sua assinatura, ela pode ser validada mesmo após o vencimento da chave.

Se todas as partes do contrato mantiverem uma cópia do selo de data/hora, cada qual pode provar que o contrato foi assinado com chaves válidas. Na verdade, o selo de data/hora pode provar a validade de um contrato mesmo se à chave de um signatário for comprometida em algum momento após a assinatura. Qualquer documento assinado digitalmente pode receber um selo de data/hora, garantindo que a validade da assinatura possa ser verificada após o vencimento da chave.

Selo de data/hora digital

Um serviço de selo de data/hora digital (DTS) emite selos que associam uma data e hora a um documento digital de forma criptograficamente confiável. O selo de data/hora digital pode ser usado posteriormente para provar que um documento eletrônico existia na ocasião declarada em seu selo de data/hora. Por exemplo, um físico que tivesse uma idéia brilhante poderia escrever sobre ela com um processador de textos e usar um selo de data/hora no documento. O selo e o documento juntos podem provar mais tarde que o cientista merece o prêmio Nobel, embora um arqui-rival possa ter sido o primeiro a divulgar a idéia.

Aqui está um exemplo de como um sistema de selo de data/hora funciona. Imagine que uma pessoa assina um documento e deseja anexar a ele um selo de data/hora. Ela cria uma síntese de mensagem do documento com uma função de *hashing* segura e depois envia a síntese (mas não o próprio documento) ao DTS, que retorna a ela um selo de data/hora consistindo na síntese da mensagem, na data e hora em que ela foi recebida pelo DTS e na assinatura do DTS. Como a síntese da mensagem não revela qualquer informação sobre o conteúdo do documento, o DTS não pode bisbilhotar o documento ao qual anexa o selo. Mais tarde, pode-se apresentar o documento e o selo de data/hora para provar quando ele foi escrito. Um computador verifica a síntese de mensagem do documento, garante que ela seja idêntica à síntese do selo de data/hora e depois verifica a assinatura do DTS no selo.

Para ser confiável, um selo de data/hora não pode ser falsificável. Considere os requisitos para um DTS do tipo descrito:

1. O próprio DTS deve ter uma chave longa se quisermos que os selos de data/hora sejam confiáveis, por exemplo, por várias décadas.
2. A chave privada do DTS deve ser armazenada com extrema segurança, como em um equipamento à prova de adulterações.

3. A data e hora devem vir de um relógio que também esteja nesse equipamento, que não possa ser reiniciado e que mantenha a hora exata por anos ou até mesmo décadas.
4. Não deve ser possível criar selos de data/hora sem os elementos desse equipamento à prova de adulterações.

Um DTS criptograficamente confiável que usa apenas software foi implementado pela Bellcore; ele evita muitos dos requisitos descritos, como o equipamento à prova de adulterações. O DTS da Bellcore combina basicamente valores de *hashing* em estruturas de dados chamadas árvores binárias, cujos valores "raiz" são periodicamente divulgados em jornais. Um selo de data/hora consiste em um conjunto de valores de *hashing* que permite a um verificador recalcular a raiz da árvore. Como as funções de *hashing* não têm volta, o conjunto de valores de *hashing* de validação não pode ser falsificado. As informações associadas ao documento pelo selo de data/hora representam a data da publicação.

O uso de um DTS pode parecer extremamente importante, se não essencial, para manter a validade de documentos durante muitas décadas. Imagine que um locador e um inquilino assinem um contrato de arrendamento de vinte anos. As chaves públicas usadas para assinar o contrato devem expirar em dois anos. Soluções como certificar as chaves novamente ou assinar o contrato a cada dois anos com novas chaves exigem a cooperação de ambas as partes vários anos após a assinatura original. Se uma parte ficar insatisfeita com o arrendamento, pode se recusar a cooperar. A solução é registrar o arrendamento com o DTS no momento da assinatura original; ambas as partes podem então receber uma cópia do selo de data/hora, que pode ser usado muitos anos depois para garantir a integridade do contrato original.

Validade dos documentos digitais

Se assinaturas digitais substituírem assinaturas à mão, elas devem ter o mesmo *status* legal, ou seja, documentos assinados digitalmente devem criar o mesmo vínculo legal. A NIST declarou que seu Digital *Signature Standard* (Padrão de Assinatura Digital) proposto deve ser capaz de "provar a um terceiro que os dados foram realmente assinados pelo gerador da assinatura." Além disso, os pedidos de compra do governo norte-americano serão assinados por um padrão semelhante; isso significa que o governo aceitará a autoridade legal de assinaturas digitais nos tribunais. Algumas pesquisas jurídicas preliminares demonstram que as assinaturas digitais atendem aos requisitos de assinaturas com valor legal para a maioria dos objetivos, incluindo seu uso comercial conforme definido no UCC (*Uniform Commercial Code*). Uma decisão da GAO (*Government Accounting Office*) solicitada pela NIST também entende que as assinaturas digitais atenderão aos requisitos legais de assinaturas à mão.

No entanto, como a validade dos documentos com assinaturas digitais nunca foi questionada em tribunais, seu *status* legal ainda não está bem definido. Com requerimentos semelhantes, os tribunais promulgarão decisões judiciais que definirão coletivamente os métodos, tamanhos de chave e medidas de segurança aceitáveis para que uma assinatura digital tenha valor legal.

Assinaturas digitais podem vir a ter uma autoridade legal maior do que assinaturas à mão. Se um contrato de dez páginas for assinado à mão na décima página, não é possível ter certeza de que as nove primeiras páginas não foram alteradas. No entanto, se o contrato tiver sido assinado digitalmente, um terceiro pode verificar que nenhum *byte* do contrato foi alterado.

Atualmente, se duas pessoas desejarem assinar digitalmente uma série de contratos, podem primeiro assinar um contrato em papel onde concordam em se submeter no futuro a quaisquer contratos assinados digitalmente por eles com um método de assinatura e um tamanho de chave mínimo determinados.

Vários esforços estão sendo feitos para legalizar o uso de assinaturas digitais. O Estado de Utah (EUA) implementou leis que qualificam as assinaturas digitais. Uma legislação semelhante está em estudo na Califórnia e em Nova York, e outros estados começam a fazer o mesmo.

QUINTO CAPÍTULO
VIRTUAL PRIVATE NETWORK

"O conhecimento amplia a vida. Conhecer é viver uma realidade que a ignorância impede desfrutar".

(Da Sabedoria Logosófia)

São redes que possibilitam um acesso privado de comunicação, utilizando-se redes públicas já existentes, como a Internet. O termo refere-se a combinação de tecnologias que asseguram a comunicação entre dois pontos, através de um "túnel" que simula uma comunicação ponto-a-ponto inacessível à "escutas clandestinas" e interferências.

As VPNs podem ser usadas de duas maneiras. No primeiro caso, existe uma conexão (sempre através de um tunelamento via Internet) entre duas redes privadas como por exemplo, entre a matriz de uma corporação, em um ponto, e um escritório remoto, em outro ponto, ou entre a rede da matriz e a rede de um parceiro. Neste tipo de conexão, a manutenção do túnel entre os dois pontos é mantida por um servidor VPN dedicado ou por existentes Internet *firewalls*. Na verdade, para estes exemplos, as VPNs podem ser encaradas como funções *Firewalls* melhoradas. Este tipo de VPN é chamada de *extranet*.

Outra forma de se usar uma VPN é conectando-se um computador remoto individual à uma rede privada, novamente através da Internet. Neste caso, a VPN é implementada através de um *software* dentro do computador remoto. Este computador poderá usar uma conexão *dial-up* local para conectar-se a Internet, possibilitando assim o alcance à rede privada.

As VPNs permitem portanto, "virtualizar" as comunicações de uma corporação, tornando-as "invisíveis" a observadores externos e aproveitando a infra-estrutura das comunicações existentes.

As VPNs permitem estender as redes corporativas de uma empresa à pontos distantes da mesma, como outros escritórios, filiais, parceiros e até mesmo uma residência. Porém, ao invés de se utilizar de um grande número linhas dedicadas para a interconexão entre seus diversos pontos, o que onera muito o custo da rede (aluguel de linhas dedicadas, manutenção de diversos links para cada conexão, manutenção de equipamentos para diferentes conexões, uso de vários roteadores, monitoramento de tráfego nas portas de acesso remoto, grande número de portas, etc), uma VPN aproveita os serviços das redes IP espalhadas mundialmente, inclusive a Internet, ou até mesmo os provedores de serviços baseados em IP *backbones* privados, os quais apesar de limitados em alcance, poderão oferecer um uma melhor performance de serviço que a Internet, em detrimento do aumento de custos. Fazendo-se então, uma mistura de serviços prestados pela Internet e serviços prestados por IPs *backbones* privados, uma corporação poderá tirar vantagens sobre a performance do serviço e a redução dos custos.

Dependendo das distâncias envolvidas, do número de sites que se quer conectar, da quantidade de dados que se pretende transmitir, entre outras variáveis, o custo com as comunicações pode ser reduzido significativamente. Alguns estudos comprovam que estes custos podem representar cerca de 60% do custo total de operação da rede, portanto, em certos casos, os projetos de VPN podem ser lucrativos, sobretudo nos casos em que links internacionais estão envolvidos, ou mesmo links nacionais de longa distância

Outra grande vantagem das VPNs é que elas podem permitir acesso a qualquer lugar acessado pela Internet e, como a Internet está presente em praticamente todos os lugares do mundo, conexões potenciais de VPNs poderão ser facilmente estabelecidas. Assim, no lugar de chamadas à longa distância, os usuários desta rede poderão, por exemplo, fazer ligações via Internet local, cuja tarifação é bem menor.

Como as VPNs possuem plataformas independentes qualquer computador configurado para uma rede baseada em IP, pode ser incorporado à VPN sem que uma modificação seja necessária, a não ser a instalação de um *software* para acesso remoto.

Ao contrário das redes privadas tradicionais que necessitam de vários links dedicados E1 (2Mbps), os quais, como dito anteriormente, acarretam diversos custos mensais fixos, mesmos quando os links não estão sendo utilizados, as redes VPNs utilizam um único link com uma banda menor (512Kbps a 768Kbps), com custo variável de acordo com sua utilização. Este único link também permite a existência de somente um roteador do lado do cliente para reunir todos os serviços de Internet e WAN, o que também permitirá redução nos custos de suporte e manutenção.

Existe ainda o fato de redes VPNs serem facilmente escalonáveis. Para se interconectar mais um escritório à rede, deve-se contatar o provedor de serviço para a instalação do link local e respectiva configuração dos poucos equipamentos nas premissas do cliente. Da mesma forma, no momento em que a utilização da rede esbarrar na banda disponível no link local alugado do provedor, basta requisitar um aumento desta banda para se determinar uma melhora considerável no desempenho da rede.

O gerenciamento da rede pode ser realizado pela própria empresa utilizadora da VPN, sendo que as alterações ocorridas na rede, como endereçamento, autenticação de usuários e determinação de privilégios de rede, são efetuadas de forma transparente ao provedor de serviço, levando a uma maior flexibilidade.

As VPNs permitem então:

- uma difusão da rede corporativa de uma empresa a custos mais baixos;
- acesso seguro e fácil de usuários remotos às redes corporativas;
- comunicação segura entre usuários da rede;
- escalabilidade, etc

5.1 ALGUMAS CONSIDERAÇÕES PARA A IMPLEMENTAÇÃO DE UMA VPN

Existe um aspecto primordial que deve ser levado em consideração para o desenvolvimento de VPNs sobre a estrutura da rede já existente: **a segurança**.

Os protocolos TCP/IP (*Transmission Control Protocol /Internet Protocol*) e a própria Internet, não foram originalmente projetados tendo a segurança como prioridade, porque o número de usuários e os tipos de aplicações não requeriam maiores esforços para a garantia da mesma. Mas, se as VPNs são substitutos confiáveis para as linhas dedicadas e outros links de WAN, tecnologias capazes de garantir segurança e performance tiveram que ser acrescentadas à Internet. Felizmente, os padrões para segurança de dados sobre redes IPs evoluíram de tal forma que permitiram a criação de VPNs

As tecnologias que possibilitaram a criação de um meio seguro de comunicação dentro da Internet asseguram que uma VPN seja capaz de:

- **Proteger a comunicação de escutas clandestinas**: a privacidade ou proteção dos dados é conseguida pela **criptografia** que, através de transformações matemáticas complexas, "codifica" os pacotes originais, para depois, decodificá-los no final do túnel. Esta codificação é o aspecto mais difícil e crítico em sistemas que implementam a criptografia.
- **Proteger os dados de alterações**: esta proteção é alcançada através de transformações matemáticas chamadas de "*hashing functions*", as quais criam "impressões digitais" utilizadas para reconhecer os pacotes alterados.
- **Proteger a rede contra intrusos**: a **autenticação** dos usuários previne a entrada de elementos não autorizados. Vários sistemas baseados em "*passwords*" ou "*challenge response*", como o protocolo CHAP (*Challenge Handshake Authentication Protocol*) e o RADIUS (*Remote Dial-in Service Protocol*), assim como *tokens* baseados em hardware e certificados digitais, podem ser usados para a autenticação de usuários e para controlar o acesso dentro da rede.

5.2 ETAPAS DA CONEXÃO ATRAVÉS DE UMA VPN

Primeiramente é feita a **autenticação** entre os dois pontos. Essa autenticação permite ao sistema enxergar se a origem dos dados faz parte da comunidade que pode exercer acesso a rede. Será o *laptop* de algum funcionário ou um roteador de um filial? Ou será alguém se passando por um usuário que faz parte da comunidade?

Em seguida, o servidor VPN verifica quais serviços que o usuário tem permissão para acessar, monitorando assim, o subsequente tráfego de dados. Este passo é chamado de autorização e visa negar acesso a um usuário que não está autorizado a acessar a rede como um todo, ou simplesmente restringir o acesso de usuários.

Uma vez formado o túnel, seu ponto de partida adiciona cabeçalhos especiais aos pacotes que serão endereçados ao outro ponto do túnel, para em seguida, criptografar e encapsular toda a informação na forma de novos pacotes IPs. Os cabeçalhos internos permitirão então a autenticação da informação, e serão capazes de detectar qualquer alteração dos dados enviados.

5.3 PROTOCOLOS DE TUNELAMENTO

Tunelamento é o encapsulamento ponto-a-ponto das transmissões dentro de pacotes IP. O tunelamento permite:

- tráfego de dados de várias fontes para diversos destinos em uma mesma infra-estrutura;
- tráfego de diferentes protocolos em uma mesma infra-estrutura;
- garantia de QoS (*Quality of Service*), direcionado e priorizando o tráfego de dados para destinos específicos.

As VPNs são, geralmente redes dinâmicas ou seja, as conexões são formadas de acordo com as necessidades das corporações. Assim, ao contrário das linhas dedicadas utilizadas por uma estrutura de rede privada tradicional, a VPN não mantém links permanentes entre dois pontos da rede da corporação, pelo contrário, quando uma conexão se faz necessária entre dois pontos desta corporação, ela é criada e quando a mesma não for mais necessária, ela será desativada, fazendo com que a banda esteja disponível para outros usuários.

Os túneis podem consistir de dois tipos de pontos finais: um computador individual ou uma *LAN* com um *gateway* seguro, que poderá ser um roteador ou um *Firewall*. Porém, somente duas combinações desses pontos finais, são consideradas nos projetos de VPNs. No primeiro caso, tunelamento *LAN-to-LAN*, um *gateway* seguro em cada ponto servirá de interface entre o túnel e a *LAN* privada. Desta forma, usuários de ambas as *LANs* poderão utilizar o túnel transparentemente para comunicarem entre si.

Um segundo caso, tunelamento *Client-to-LAN*, é aquele utilizado por usuários remotos que desejam acessar a *LAN* corporativa. O cliente, ou seja, o usuário remoto, inicia o tunelamento em seu ponto, para a troca de tráfego com a rede corporativa. A ferramenta para esta comunicação é um *software* instalado em seu computador, que permite transpor o *gateway* que protege a *LAN* de destino.

Os principais protocolos de tunelamento são:

GRE (Generic Routing Protocol)

Túneis GRE são geralmente configurados entre roteadores fonte e roteadores destino (pacotes ponto-a-ponto). Os pacotes designados para serem enviados através do túnel (já encapsulados com um cabeçalho de um protocolo como, por exemplo, o IP) são encapsulados por um novo cabeçalho (cabeçalho GRE) e colocados no túnel com o endereço de destino do final do túnel. Ao chegar a este final, os pacotes são desencapsulados (retira-se o cabeçalho GRE) e continuarão seu caminho para o destino determinado pelo cabeçalho original.

Desvantagens:

- Os túneis GRE são, geralmente, configurados manualmente, o que requer um esforço grande no gerenciamento e manutenção de acordo com a quantidade de túneis: toda vez que o final de um túnel mudar, ele deverá ser manualmente configurado.
- Embora a quantidade de processamento requerida para encapsular um pacote GRE pareça pequena, existe uma relação direta entre o número de túneis a serem configurados e o processamento requerido para o encapsulamento dos pacotes GRE: quanto maior a quantidade de túneis, maior será o processamento requerido para o encapsulamento.
- Uma grande quantidade de túneis poderá afetar a eficiência da rede.

PPTP (Point-to-Point Tunneling Protocol), L2F (Layer-2 Forwarding) ,L2TP (Layer 2 Tunneling Protocol)

Ao contrário do GRE, estes protocolos são utilizados em VPDNs (*Virtual Private Dial Network*), redes que proporcionam acesso à rede corporativa por usuários remotos, através de uma linha discada (provedor de acesso).

PPTP

O protocolo PPTP é um modelo "voluntário" de tunelamento, ou seja, permite que o próprio sistema do usuário final, por exemplo, um computador, configure e estabeleça conexões discretas ponto-a-ponto para um servidor PPTP, localizado arbitrariamente, sem a intermediação do provedor de acesso. Este protocolo constrói as funcionalidades do protocolo PPP (*Point-to-Point Protocol* - um dos protocolos mais utilizados na Internet para acesso remoto) para o tunelamento dos pacotes até seu destino final. Na verdade, o PPTP encapsula pacotes PPP utilizando-se de uma versão modificada do GRE, o que torna o PPTP

capaz de lidar com outros tipos de pacotes além do IP, como o IPX (*Internet Packet Exchange*) e o NetBEUI (*Network Basic Input/Output System Extended User Interface*), pois é um protocolo baseado na camada 2 do modelo OSI (enlace).

Neste modelo, um usuário disca para o provedor de acesso à rede, mas a conexão PPP é encerrada no próprio servidor de acesso. Uma conexão PPTP é então estabelecida entre o sistema do usuário e qualquer outro servidor PPTP, o qual o usuário deseja conectar, desde que o mesmo seja alcançável por uma rota tradicional e que o usuário tenha privilégios apropriados no servidor PPTP.

L2F

Foi um dos primeiros protocolos utilizado por VPNs. Como o PPTP, o L2F foi projetado como um protocolo de tunelamento entre usuários remotos e corporações. Uma grande diferença entre o PPTP e o L2F, é o fato do mesmo não depender de IP e, por isso, é capaz de trabalhar diretamente com outros meios como *FRAME RELAY* ou ATM.

Este protocolo utiliza conexões PPP para a autenticação de usuários remotos, mas também inclui suporte para TACACS+ e RADIUS para uma autenticação desde o inicio da conexão. Na verdade, a autenticação é feita em dois níveis: primeiro, quando a conexão é solicitada pelo usuário ao provedor de acesso; depois, quando o túnel se forma, o *gateway* da corporação também irá requerer uma autenticação.

A grande vantagem desse protocolo é que os túneis podem suportar mais de uma conexão, o que não é possível no protocolo PPTP. Além disso, o L2F também permite tratar de outros pacotes diferentes de IP, como o IPX e o NetBEUI por ser um protocolo baseado na camada 2 do modelo OSI.

L2TP

Este protocolo foi criado pela IETF (Internet Engennering Task Force) para resolver as falhas do PPTP e do L2F. Na verdade, utiliza os mesmo conceitos do L2F e assim como este, foi desenvolvido para transportar pacotes por diferentes meios, como X.25, *frame-relay* e ATM e também é capaz de tratar de outros pacotes diferentes de IP, como o IPX e o NetBEUI (protocolo baseado na camada 2 do modelo OSI .

O L2TP é porém, um modelo de tunelamento "compulsório", ou seja, criado pelo provedor de acesso, não permitindo ao usuário qualquer participação na formação do túnel (o tunelamento é iniciado pelo provedor de acesso). Neste modelo, o usuário disca para o provedor de acesso à rede e, de acordo com o perfil configurado para o usuário e ainda, em caso de autenticação positiva, um túnel L2TP é estabelecido dinâmicamente para um ponto pré-determinado, onde a conexão PPP é encerrada.

PPTP x L2TP

Apesar de parecidos, ambos os protocolos, L2TP ou PPTP, diferenciam-se quanto suas aplicações, ou melhor, a escolha do protocolo a ser utilizado é baseado na determinação da posse do controle sobre o túnel: controlado pelo usuário ou pelo provedor de acesso.

No protocolo PPTP, o usuário remoto tem a possibilidade de escolher o final do túnel, destino dos pacotes. Uma grande vantagem desta característica é que, quando os destinos mudam com muita frequência, nenhuma modificação (configuração) nos equipamentos por onde o túnel passa se torna necessária. Além disso, os túneis PPTP são transparentes aos provedores de acesso e nenhuma outra ação, além de prover serviço de acesso à rede, se faz necessária. Usuários com perfis diferenciados em relação aos locais de acesso – diferentes cidades, estados e países – se utilizam deste protocolo com mais frequência pelo fato de se tornar

desnecessária a intermediação do provedor no estabelecimento do túnel. Somente é necessário saber o número local para o acesso e o sistema do usuário, seu *laptop*, realizará o resto.

A desvantagem do protocolo L2TP é que, como o controle está na mão do provedor, o mesmo esta fornecendo um serviço extra que poderá ser cobrado.

IPSec

PPTP, L2F e L2TP não incluem criptografia ou processamento para tratar chaves criptográficas, o que é bastante recomendado para garantir a segurança dos pacotes. Por isso, surgiu um dos mais importantes protocolos, criado para garantir a segurança da próxima geração de pacotes IP (IPv6) e que, no momento, vem sendo utilizado com protocolos IPv4.

O IPSec permite ao usuário, ou ao *gateway* seguro que está agindo em seu favor, autentificar ou criptografar cada pacote IP, ou ainda, fazer os dois processos simultaneamente. Assim, separando os processos de autenticação e de criptografia, surgiram dois diferentes métodos para a utilização do IPSec, chamados de modos: no modo transporte, somente o segmento da camada de transporte de um pacote IP é autentificado ou criptografado; a outra abordagem, autenticação e criptografia de todo o pacote IP, é chamada de modo túnel. Enquanto que no modo transporte o IPSec tem provado ser eficiente para várias situações, no modo túnel ele é capaz de prover uma proteção maior contra certos ataques e monitoração de tráfego que podem ocorrer na Internet.

O IPSec é baseado em várias tecnologias de criptografias padronizadas para proverem confiabilidade, integridade de dados e confiabilidade. Por exemplo, o IPSec utiliza:

- *Diffie-Hellman-Key-exchanges* para entregar chaves criptográficas entre as partes na rede pública.
- *Public-key-criptography* para sinalizar trocas do tipo *Diffie-Hellman* e garantir a identificação das duas partes, evitando assim, ataques de intrusos no meio do caminho.
- DES e outros algorítmos para criptografar dados.
- Algorítimos para a autenticação de pacotes que utilizam "*hashing functions*".
- Certificados digitais para validar chaves públicas.

Existem duas maneiras para lidar com a troca de chaves e gerenciamento numa arquitetura IPSec: chaveamento manual (*manual keying*) e *Internet Key Exchange* (IKE) para gerenciamento automático de chaves. Enquanto o chaveamento manual pode ser usado em VPNs com um número pequeno de sites, o IKE deve ser obrigatoriamente em VPNS que suportam um grande número de sites e usuários remotos.

O IPSec tem sido considerado a melhor evolução para ambientes IP por incluir fortes modelos de segurança - criptografia, autenticação e troca de chaves - mas não foi desenvolvido para suportar outros tipos de pacotes além do IP. No caso de pacotes multiprotocolos, devem ser usados PPTP ou L2TP que suportam outros tipos de pacotes.

5.4 SOLUÇÕES PARA VPNS

Existe quatro componentes básicos para a implementação de uma VPN baseada na Internet: a Internet, *gateways* seguros, servidores com políticas de segurança e certificados de autenticidade.

A Internet provê a "sustentação" de uma VPN e os *gateways* seguros são colocados na fronteira entre a rede privada e a rede pública para prevenirem a

entrada de intrusos, e ainda são capazes de fornecer o tunelamento e a criptografia antes da transmissão dos dados privados pela rede pública. Os *gateways* seguros podem ser: roteadores, *Firewalls*, *hardwares* específicos e *softwares*.

Como roteadores necessitam examinar e processar cada pacote que deixa a LAN, parece natural incluir-se a criptografia dos pacotes nos roteadores. Por isso existe no mercado dois produtos que desempenham esta função em roteadores: *softwares* especiais (*software* adicionado ao roteador) ou placas com coprocessadores que possuem ferramentas de criptografias (*hardware* adicionado ao roteador). à grande desvantagem dessas soluções é que, se o roteador cair, a VPN também cairá.

Os *firewalls*, assim como os roteadores, também devem processar todo o tráfego IP, neste caso, baseando-se em filtros definidos pelas mesmas. Por causa de todo o processamento realizado nos *firewalls* , elas não são aconselhadas para tunelamento de grandes redes com grande volume de tráfego. A combinação de tunelamento e criptografia em *firewalls* será mais apropriada para redes pequenas com pouco volume de tráfego (1 a 2Mbps sobre um link de LAN). Da mesma forma que os roteadores, os *firewalls* podem ser um ponto de falha de VPNs.

A utilização de hardware desenvolvido para implementar as tarefas de tunelamento, criptografia e autenticação é outra solução de VPN. Esses dispositivos operam como pontes, implementando a criptografia, tipicamente colocadas entre o roteador e os links de *WANs*. Apesar da maioria desses *hardwares* serem desenvolvidos para configurações *LAN-to-LAN*, alguns produtos podem suportar túneis *client-to-LAN*. à grande vantagem desta solução é o fato de várias funções serem implementadas por um dispositivo único. Assim, não há necessidade de se instalar e gerenciar uma grande quantidade de equipamentos diferentes, fazendo com que esta implementação seja muito mais simples que a instalação de um *software* em um *firewall*, a reconfiguração de um roteador ou ainda a instalação de um servidor RADIUS, por exemplo.

Uma VPN desenvolvida por *software* também é capaz de criar e gerenciar túneis entre pares de *gateways* seguros ou, entre um cliente remoto e um *gateway*

seguro. Esta é um a solução que apresenta um custo baixo, mas desaconselhada para redes que processam grande volume de tráfego. Sua vantagem, além do baixo custo, é que esta implementação pode ser configurada em servidores já existentes e seus clientes. Além disso, muitos desse *softwares* se encaixam perfeitamente para conexões *client-to-LAN.*

A política de segurança dos servidores também é outro aspecto fundamental para implementação de VPNs. Um servidor seguro deve manter uma lista de controle de acesso e outras informações relacionadas aos usuários, que serão utilizadas pelos *gateways* para a determinação do tráfego autorizado. Por exemplo, em alguns sistemas, o acesso pode ser controlado por um servidor RADIUS.

Por último, certificados de autenticidade são necessários para verificar as chaves trocadas entre sites ou usuários remotos. As corporações podem preferir manter sèu próprio banco de dados de certificados digitais para seus usuário através de um servidor de certificado ou, quando o número de usuários for pequeno, a verificação da chaves poderá requerer o intermédio de uma terceira parte, a qual mantém os certificados digitais associados à chaves criptográficas, pois a manutenção de um servidor para isso será muito onerosa.

5.5 A ESCOLHA DA MELHOR SOLUÇÃO PARA VPNs

Vejamos as vantagens e desvantagens de cada solução para a implementação de VPNs: apenas *software*, *software* auxiliado por hardware e hardware específico.

O encapsulamento aumenta o tamanho dos pacotes, consequentemente, os roteadores poderão achar que os pacotes estão demasiadamente grandes e fragmentá-los, degradando assim a performance da rede. A fragmentação de pacotes e a criptografia poderão reduzir a performance de sistemas discados a níveis inaceitáveis mas a compressão de dados poderá solucionar este problema. No entanto, a combinação de compressão com encapsulamento, irá requerer um poder

computacional mais robusto para atender às necessidades de segurança. Por isso, uma VPN implementada através de hardware, devido ao seu poder computacional irá alcançar uma melhor performance. Este tipo de implementação também fornece uma melhor segurança - física e lógica - para a rede, além de permitir um volume de tráfego maior. A desvantagem desta implementação é um custo mais alto e o uso de hardware especializado.

Já uma VPN implementada através de *software* terá critérios menos rígidos de segurança mas se encaixa perfeitamente para atender às necessidades de conexão de pequenos volumes que não precisam de grandes requisitos de segurança e possuem um custo menor.

Quanto a performance de VPNs implementadas por *software* assistido por hardware, está dependerá da performance dos equipamentos aos quais o *software* está relacionado.

A tabela a seguir resume os aspectos a serem considerados para a escolha do tipo de solução para a implementação de uma VPN.

Solução	**Apenas *software***	***Software* assistido por Hardware**	**Hardware especializado**
Performance	baixa	média-baixa	alta
Segurança	plataforma fisicamente e logicamente insegura	plataforma fisicamente e logicamente insegura	fisicamente e logicamente seguro
Aplicações possíveis	*dial-up* a uma taxa de 128Kbps para dados ISDN	ISDN à velocidades T1	Velocidades *dial-up* até 100Mbps
Produtos	*Firewalls*, *Softwares* de VPNs	Cartões de criptografia para roteadores, PCs (*Personal Communication Services*)	*Hardware* especializado

QUADRO 8. Soluções para implementação de VPNs

5.5 PERFORMANCE E QoS

Dois fatores determinam a performance das VPNs:

A velocidade das transmissões sobre a Internet ou sobre outra rede ou *backbone* IP

A eficiência do processamento dos pacotes (estabelecimento de uma seção segura, encapsulamento e criptografia de pacotes) em cada ponto da conexão: origem e destino.

A Internet não foi projetada inicialmente para garantir níveis confiáveis e consistentes de tempo de resposta. Na verdade, a Internet é um meio de comunicação *"best effort"*, ou seja, realiza o máximo de esforço para prestar o serviço a qual é destinada: a transmissão de dados da origem ao destino. Além disso, o criptografia e o processo de tunelamento podem influir bastante na velocidade de transmissão dos dados pela Internet. Contudo, muitas redes corporativas, não podem ficar a mercê dessas flutuações de performance e acesso da Internet.

Alguns provedores de serviço resolveram o problema de velocidade de transmissão oferecendo acordos de Qualidade de Serviço (QoS), e garantia de banda à níveis específicos. Mas , atualmente, o método mais eficaz para adquirir QoS é mandar o tráfego VPN sobre o próprio IP backbone do provedor de serviço, ou seja, sobre o *frame relay* do provedor ou sobre circuitos ATM, e não sobre a Internet.

Também existem hoje, diversos grupos de estudos ou *Task Forces* da IETF tentando solucionar alguns dos problemas relacionados a performance e Qualidade de Serviço na Internet. São eles :

- **RFC2211**, "*Specification of the Controlled-Load Network Element Service*,", J. Wroclawski, September 1997.
- **RFC2212**, "*Specification of Guaranteed Quality of Service*,", S. Shenker, C. Partridge, R. Guerin, September 1997.

- **RFC2208**, "*Resource ReSerVation Protocol (RSVP) Version 1 – Applicability Statement, Some Guidelines on Deployment*," A. Mankin, F. Baker, S. Bradner, M. O'Dell, A. Romanow, A. Weinrib, L. Zhang, September 1997.

5.6 TECNOLOGIAS DE VPN ATUAIS

Atualmente, existem dois tipos de tecnologias em estudo para a implementação de VPNs, além do protocolo PPTP:

- As soluções de VPN proprietárias
- padrão S/WAN

Soluções proprietárias

A transmissão segura dos dados na Internet pode ocorrer através de VPNs com tecnologias proprietárias já disponíveis no mercado. Existem soluções de VPN criadas tanto por vendedores de *Firewall* quanto por desenvolvedores de produtos de TCP/IP. De forma resumida, as VPNs de tecnologia proprietária são implementadas através de *Firewalls* ou de *Tunnel Servers*.

Para empresas que ainda não possuem um *Firewall* e planejam implementar uma VPN, é aconselhável já optar por um *firewall* que ofereça este serviço de forma integrada.

Se a empresa já possui um *firewall* que não suporta VPN, pode utilizar a tecnologia de *Tunnel Server*, que é uma solução combinada de *software* e *hardware* capaz de criar um "túnel de IP" onde os dados transportados pelo TCP/IP podem ser transmitidos de forma segura, criptografados.

O padrão S/WAN

O padrão S/WAN (*Secure Wide Area Network*) permitirá a criação de produtos de VPN de que sejam compatíveis entre si, eliminando os atuais problemas de interoperabilidade das soluções proprietárias. O padrão S/WAN será baseado no IPSec, a nova versão "segura" do TCP/IP, podendo ainda implementar outros níveis de seguridade utilizando inclusive SSL.

As VPNs implementadas utilizando a Internet são provavelmente mais lentas que as que utilizam linhas dedicadas, o que a torna pouco recomendável para aplicações sensíveis ao tempo de transmissão, já que é complicado controlar o fluxo das mensagens através da Internet.

Apesar da criptografia, as VPNs não são tão seguras quanto o uso de linhas dedicadas.

CONCLUSÃO

A arquitetura inicial das redes e principalmente da Internet não foi implementada visando mecanismos concretos de segurança. Esta segurança não era o principal objetivo de seus criadores, pelo menos inicialmente, o objetivo principal era montar uma estrutura funcional. E devemos ver que isso aconteceu com muitos louvores, pois a Internet e as redes em geral se tornaram muito populares e essenciais para quase todo o tipo de serviço.

Já a questão da segurança não obteve o mesmo grau de crescimento, pois as redes (*hardware* e *software* disponível) tiveram uma ascensão muito rápida e nem todos os itens foram avaliados pelo foco de que "não estamos sozinhos". Situação esta, hoje invertida, pois a preocupação dos últimos anos da comunidade científica tem sido exatamente criar mecanismos que garantam confidencialidade e integridade das informações.

Só que como qualquer outro sistema, os sistemas computacionais não podem afirmar que utilizando os mecanismos aqui apresentados durante o decorrer do trabalho, deixem sua rede segura. Este é um termo muito forte e muito utilizado, só que na verdade não pode ser aplicado como se escreve ou lê, pois comprovadamente todo sistema (não somente os computadorizados) possuem o seu "calcanhar de Aquiles".

Estas "brechas" podem e devem ser reduzidas, como está sendo feito nas redes. Só que são anos de liberdade, então teremos alguns anos até chegar a um patamar de segurança máxima, que restrinja realmente os acessos indevidos e perdas de consistência de informações vitais.

Além de se analisar, neste estudo, as formas de garantir mais segurança as redes, também foi analisado o invasor, o *Hacker*.

Muitos se denominam *Hackers*, mas na verdade são seres humanos com grandiosidade pessoal inflada. Muitos somente repassam o que encontram em revistas do gênero e troca de informações com colegas do "ramo". São

principalmente adolescentes que procuram diversão anarquizando pequenos usuários que são confiantes demais na sua sorte e que não utilizam nenhuma forma de resguarde de seus dados e de seu acesso. Este tipo de *hacker* é perigoso somente pequenos usuários. Mas qualquer usuário deve estar atento às páginas com este tipo de informação, pois assim, estarão conhecendo os aplicativos e métodos desta "turma".

Mas existem os *hackers* verdadeiros. Estes não são tão fáceis de achar, muito pelo contrário, só os que não são *hackers* no sentido real da palavra (uma pessoa muito especializada) é que gostam de publicidade. Os verdadeiros *hackers* são realmente muito especializados, conhecem protocolos, sistemas de autenticação, ferramentas de desenvolvimento como ou quase como os seus próprios desenvolvedores.

Na verdade não tem muito segredo, se unem pela especialidade e juntam conhecimento, exatamente como qualquer grupo de trabalho organizado faria. Só que seus objetivos são diversos. Podem ser organizacionais, espionagem, sabotagem, terrorismo cibernético etc.

Mas fique tranqüilo, estes são uma parcela bem pequena de todos os ditos *hackers* e como já sabemos seus métodos de trabalho podemos dificultar seu serviço criando mecanismos de atualizações, inovações de protocolos de segurança, autenticidade e certificação de usuários, que deixam nenhuma rede inviolável, mas dificultam bastante a invasão, sendo que mesmo que ela ocorra pode em muitos casos ser rastreada.

Os mecanismos disponíveis devem ser avaliados de acordo com a necessidades da rede com relação a custos e a serviços utilizados. Devem ser levantados os processos reais e a partir deles definir um projeto de segurança. Se será necessário o uso de mecanismos de autenticação ou a instalação de um *firewall*, é o projeto que vai indicar. Um item que deve ser amplamente discutido e avaliado são os problemas internos à rede, pois à grande maioria de problemas referentes à segurança estão dentro da empresa e não somente nos seu acesso ou site da Internet.

Os *firewalls* são barreiras importantes no controle de acesso indevido. Podem ser utilizados desde o tipo mais simples com um único roteador ou até a arquitetura de *firewall Screened*. O *screened host* já fornece uma certa garantia de invasões, mas neste mundo virtual, redundâncias podem salvar a rede de acessos não autorizados utilizando a arquitetura de *screened subnet*. Esta arquitetura simula uma rede isolada (dois roteadores) entre a Internet e a rede privada, sendo que a redundância do segundo roteador (passará novamente por filtros) aumenta as chances de "derrubar" uma acesso indevido.

A nível de transações existe um pacote de oportunidades, sendo que o mais utilizado e validado por sites é o SSL. Considerado hoje como padrão para transações de informações à nível de protocolo. Existem possibilidades de se aumentar o grau de criptografia nos pacotes, mas todos os métodos sugerem criptografia com chave pública somente na chave única utilizada para criptografar os dados, pois a maioria dos métodos de criptação são muito lentos. Para fins comerciais o padrão é o SET, que já possui mecanismos internos que avaliam as situações de compra e venda, além da transmissão segura dos dados.

A integridade e confiabilidade dos dados que trafegam na rede é requisito fundamental para mensagens comerciais e sigilosas. Também pode ser feito utilizando protocolos como o SSL, só que ao invés de criptografar a mensagem a ser encaminhada ao usuário, somente com à chave pública dele, a criptografa também com sua chave privada.

As assinaturas digitais correspondem a um grau de segurança superior ao convencional "assinado à mão", pois pode-se verificar a assinatura da mensagem e se o conteúdo dela não foi alterado. Ainda não são consideradas legalmente, mas a nível de segurança no transporte dos dados e identificação do remetente possuem alto grau de segurança, o que deve ser levado em consideração no momento de optar por esta implementação.

Chegamos num ponto onde a combinação destes três tópicos citados acima (controle de acesso, transações seguras e autenticação) ou o uso de apenas um ou outro (dependendo da necessidade levantada) restringe e dificulta ataques, invasões,

acessos não previstos, roubo de informação, perda de informação e muitos outros processos não desejados pelos administradores, mas, não a torna invulnerável.

As VPNs são uma proposta de combinação de alguns métodos com padrões conhecidos, o que aumenta a segurança e reduz os custos pois a Internet já fornece a base para as VPNs. A arquitetura destas redes permite autenticação de todos os seus usuários o que reduz drasticamente os problemas com acessos indevidos a serviços disponibilizados pela rede.

Junto com todos estes mecanismos deve estar um bom plano de segurança e de contingência. Pois como já salientado, o nível de segurança tem de ser mantido por atualizações de *software* e, se for o caso, *hardware*, em períodos regulares determinados pelo plano. Também devem ser mantidos controles sobre os acessos dos usuários (*log*) que mantenham verificação constante sobre suas ações e serviços, facilitando assim a avaliação das permissões de serviços, pois qualquer irregularidade demostra falha no controle de acesso.

Muitas opções estão disponíveis, então verifique atenciosamente o seu caso e quais mecanismos se enquadram as suas necessidades. Opte por ferramentas já validadas por outras empresas e conceituadas no mercado, pois "lançar moda" nesta área pode ser um risco muito sério.

E como já mencionado antes, se deseja saber o quanto a sua atual infraestrutura é segura, simule ataques conhecidos contra ela ou monte uma infraestrura similar a rede a ser avaliada e proceda com todo o tipo de invasão. Resumindo torne-se um bom *hacker*, nos dois sentidos do advérbio (seja um bom profissional e use suas habilidades para fazer o bem).

Este trabalhou procurou fornecer base conceitual para que outras pesquisas sobre o tema pudessem advir. Como é uma área em constante dinâmica é sempre importante recorrer as soluções mais recentes e aprimorar o conhecimento aqui adquirido. Como sugestão poderia ser utilizada em laboratórios de redes, onde se busca-se fornecer bases para à segurança de redes.

REFERÊNCIAS BIBLIOGRÁFICAS

BRETCHAPMAN, D. **Building Internet Firewalls**. Sebastopol : O'Reilly & Associates, Inc, p. 544, 1995.

CARUSO, CAA.; STEFFEN, FD. **Segurança em informática**. Rio de Janeiro : Livros Técnicos e Científicos. p. 272, 1991.

COMER, DE.; STEVENS, DL. **Internetworking with TCP/IP**: design, implementation, and internals. New Jersey : Prentice Hall. 2 ed., v. II, p. 508, 1994.

COMER, DE.; STEVENS, DL. **Internetworking with TCP/IP**: client-server programming and applications. New Jersey : Prentice Hall. v. III, p. 508, 1994.

DORASWAMY, N.; HARKINS, D. IPSec: The New Security Standard for the Internet, Intranets, and Virtual Private Networks. New Jersey : Prentice Hall, 1999. ISBN: 0130118982

GALLI, D. L. **Distributed Operating Systems**. New Jersey : Prentice Hall. p.480, 1999. ISBN: 0130798436

JENNINGS, R. **Usando o Windows NT Server 4**. Rio de Janeiro : Campus. p.1117, 1997.

LIU, C; et. al. **Mananging Internet, Information Services**. Sebastopol : O'Reilly & Associates, Inc, dez., 1994.

MENEZES, A.; OORSCHOT, PV., VANSTONE, S. **Handbook of applied cryptograpy**, 1996.

OPPENHEIMER, P. **Projeto de Redes Top-Down**. Rio de Janeiro : Campus. p. 492, 1999.

OLIVO, LCC. **Direito e Internet: a regulamentação do ciberespaço**. Florianópolis : UFSC. p.154, 1998.

PARKER, D.B. **Computer Security Management**. New Jersey : Prentice-Hall, 1997.

RAMOS, AM. **Interface de controle de acesso para o modelo de gerenciamento OSI**. Dissertação apresentada e aprovada no Curso de Pós-graduação em Ciência da Computação da Universidade Federal de Santa Catarina/UFSC, 1994.

RANUM, M. et ali. **Web Security Sourcebook**. Paperback. p. 368, 1997.

SOARES, LFG.; LEMOS, G.; COLCHER, S. **Redes de Computadores:** das LANs, MANs e WANs às redes ATM. Rio de Janeiro : Campus. 2 ed., p.705, 1995.

STARLIN, G. **TCP/IP**. Rio de Janeiro : Book Express. p. 351, 1999.

TANENBAUN, AS. **Redes de Computadores**. Rio de Janeiro : Campus. 2 ed., p.786, 1994.

MACGREGOR, R. **Secure Electronic Transactions:** credit card payment on the Web in theory and practice. IBM Corporation. p. 338, 1997. ISBN number 0738404691

WILLIAM, S. Cryptography and Network Security : Principles and Practice. New Jersey : Prentice Hall. 2 ed., p. 528, 1998. ISBN: 0138690170

Sites de Organizações Internacionais de Padronização

http://www.rfc-editor.org (RFC's)

http://www.gocsi.com/

http://www.iana.org/

http://csrc.nist.gov/

http://www.nsi.org/

http://www.cert.org/

Sites por Tópico Abordado

Segurança na Internet

http://www.notivagos.cjb.net/
Apostila sobre Segurança na Internet

http://www.process.com
Networking Solutions

http://www.modulo.com.br
Segurança para redes, internet e intranet

http://www.protocols.com
Definições e Segurança dos Protocolos

Firewalls

http://www.microsoft.com/WINDOWS2000/library/resources/reskit/samplechapters/inbe/inbe_vpn_hidv.asp
VPNs and Firewalls

http://www.cisco.com/warp/public/779/ibs/solutions/
Internet Business Solutions

http://*routerbits*.com
Security and Your Internet Rights

http://www.nclab.hanyang.ac.kr/resource/seminar/network
Research of Data Access Protocol

http://www.modulo.com/heatsite/test1.htm#why
Protegendo sua rede contra ameaças internas
NetVital Technologies - Mar/98 - Dan Sigal

VPNC (Virtual Private Network Consortium)

http://www.vpnc.org/

http://www.microsoft.com/ntserver/commserv/techdetails/overview/vpnovw.asp
Virtual Private Networking Overview, from Microsoft

http://www.shiva.com/remote/prodinfo/vpn/index.html
Virtual Private Networking: The Next Revolution in Corporate Productivity, from Intel

http://www.compatible.com/vpn_now/education.html
What is VPN?, from Compatible

http://www.cisco.com/warp/public/cc/cisco/mkt/servprod/dial/justify/profiles/avpnnbc.htm
Access VPNs for Enterprises, from Cisco

http://www.cisco.com/warp/public/cc/sol/mkt/ent/vpne/tech/qsvpn_wp.htm
Quality of Service for Virtual Private Networks, from Cisco

http://www.timestep.com/downloads/ipsec.pdf
Understanding the IPSec protocol suite, from TimeStep

http://www.microsoft.com/windows/server/technical/networking/NWPriv.asp
Microsoft Privacy Protected Network Access: Virtual Private Networking and Intranet Security, from Microsoft

http://www.microsoft.com/ntserver/commserv/techdetails/prodarch/understandingpptp.asp
Understanding Point-To-Point Tunneling Protocol (PPTP), from Microsoft

http://www.timestep.com/Html/Crypto.htm
Basic Cryptography: a primer, from TimeStep

http://www.cisco.com/warp/public/cc/sol/mkt/ent/vpne/evpn2_pg.ppt
Enterprise VPN Solutions: Security Tutorial, from Cisco

http://www.indusriver.com/en/
Comprehensive Policy Management for the Enterprise VPN, from Indus River

http://www.microsoft.com/windows2000/library/resources/reskit/samplechapters/inbe/inbe_vpn_naxe.asp
Point-to-Point Tunneling Protocol

Identificação Digital

http://www.teiajuridica.com/af/doceletr.htm
Aspectos Jurídicos do Documento Eletrônico
José Henrique Barbosa Moreira Lima Neto

http://www.certisign.com.br/
Centro de Certificação

http://verisign.mandic.com.br/client/help/semail.htm#6
Mensagens eletrônicas seguras

Aplicativo PGP

http://www.mantis.co.uk/pgp/pgp.html
Pretty Good Privacy

http://www.pegasus.esprit.ec.org/people/arne/pgp.html
Covers MIT version 2.6.2.

http://www.huygens.org/~arne/pgpdoc1/pgpdoc1.html
(© Copyright 1990--1994 Philip Zimmermann)
PGP User's Guide, Volume I: Essential Topics

http://www.huygens.org/~arne/pgpdoc2/vulnerabilities.html
(© Copyright 1990--1994 Philip Zimmermann)
PGP Vulnerabilities

SSL

http://developer.netscape.com/tech/security/ssl
How SSL Works

http://msdn.microsoft.com/library/devprods/vs6/vstudio/vsentpro/veconsecuredcommunicationsssl.htm
Secured Communications and SSL

Kerberos

http://web.mit.edu/kerberos/www/index.html
Kerberos: The Network Authentication Protocol

http://www.isi.edu/gost/brian/security/kerberos.html
The Moron's Guide to Kerberos, Version 1.2.2

http://www.isi.edu/gost/publications/kerberos_neumam-tso.html
Kerberos: An Authentication - Service for Computer Networks

http://www.microsoft.com/windows2000/library/howitworks/security/kerberos.asp
Windows 2000 Kerberos Authentication

http://www.microsoft.com/windows2000/library/howitworks/security/kerbint.asp
Windows 2000 Kerberos Interoperability

Algoritmos de criptografia

http://theory.lcs.mit.edu/~rivest/crypto-security.html
Cryptography and Security

ftp.funet.fi:/pub/crypt/cryptography/asymmetric/rsa.

http://rschp2.anu.edu.au:8080/crypt.html

http://www.farcaster.com/
Miscellaneous Works

http://www.itl.nist.gov/fipspubs/fip180-1.htm
Secure Hash Standard

ftp.funet.fi:/pub/crypt/hash/sha

ftp.funet.fi:/pub/crypt/hash/mds

http://www.esat.kuleuven.ac.be/~bosselae/ripemd160.html#What
(RIPEMD-160 e SHA-1)

http://www.ee.udel.edu/~redwinsk/seminar/sld10.htm
Routhing Algorithm

PCT 1.0

http://www.microsoft.com/Windows/ie/support/docs/tech30/security.htm
Security
http://msdn.microsoft.com/library/devprods/vs6/vstudio/vsentpro/veconsecuredcommunicationsssl.htm
Secured Communications and SSL

HTTPs

http://penta.ufrgs.br/redes296/https/ (ftp)

GLOSSÁRIO

Abend parada ou desligamento imprevisto.

Abuso de privilégio quando um usuário realiza uma ação que ele não estava autorizado a realizar.

Accountability é a propriedade que garantirá que qualquer ação possa ser rastreada e identificada de forma única e individual. Medidas pelas quais a identidade de um usuário possa ser determinantemente associada com o acesso à máquinas, material, e o tempo, e nível de acesso.

Acesso interação entre um sujeito e um objeto que permite a informação fluir de um para o outro (definição de processamento de dados); a capacidade de entrar em um prédio seguro (definição de segurança física).

Administração de segurança as regras gerenciais e os controles suplementares estabelecidos para fornecer um nível aceitável da proteção para dados.

Algoritmo assimétrico (*Asymmetric Algorithm*) é um algoritmo de criptografia que usa duas chaves: uma chave pública e uma chave privada, onde à chave pública pode ser distribuída abertamente enquanto à chave privada é mantida secreta. Os algoritmos assimétricos são capazes de muitas operações, incluindo criptografia, assinaturas digitais e acordo de chave.

Algoritmo hash produz um valor *hash* de partes de dados como uma mensagem ou chave de sessão. Os algoritmos mais conhecidos e utilizados são MD2, MD4, MD5 e SHA-1.

Algoritmo simétrico algoritmo de criptografia que usa somente uma chave, tanto para criptografar como para descriptografar. Esta chave deve ser mantida secreta para garantir a confidencialidade da mensagem. Também conhecido como algoritmo de chave secreta.

Analisador de protocolos dispositivo de hardware ou *software* que oferece vários recursos para solução de problemas de rede, inclusive decodificações, testes pré-programados de solução de problemas e geração de tráfego.

Análise de Custo Benefício avaliação dos custos para prover proteção de dados para um sistema versus o custo de perder ou comprometer aqueles dados.

API um método pelo qual um programa pode obter acesso ao sistema operacional ou modificá-lo.

Aplicativo produto de *software* que resulta da criação de um programa, frequentemente usado como sinônimo para o código de programação (código-fonte) com base no qual é criado. Os aplicativos se distinguem pelo ambiente ao qual se destinam (por exemplo, Windows, DOS, Macintosh, UNIX) e pela sua finalidade.

ARP protocolo da Internet usado para mapear um endereço IP em um endereço MAC. Definido na RFC 826.

ARPANET rede criada no final dos anos 60 pela ARPA (hoje DARPA), para estimular pesquisas sobre o tema "redes de computadores" através de contratos com os departamentos de computação de várias universidades americanas e com algumas poucas corporações privadas. Muito do conhecimento atual sobre redes é resultado direto da ARPANET.

ASN.1 método para descrição de dados que é utilizado por muitos padrões. Referência no CCITT.

Ataque o ato de tentar desviar dos controles de segurança de um sistema. Um ataque pode ser ativo, tendo por resultado a alteração dos dados; ou passivo, tendo por resultado a liberação dos dados. Nota: O fato de um ataque estar acontecendo não significa necessariamente que ele terá sucesso. O nível de sucesso depende da vulnerabilidade do sistema ou da atividade e da eficácia de contramedidas existentes.

Ataque dirigido à dados (*Data Driven Attack*) forma de ataque em que o ataque é codificado nos dados normais e inócuos que são executados por um usuário ou pelo outro *software*. No caso dos *firewalls*, um ataque dirigido aos dados interessante, desde que possa pelo *firewall* no formato de dados e lançar um ataque.

Ativo (*Asset*) tudo que faz parte da operação de um sistema ou desenvolvimento (ex.: hardware, *software*, documentação, equipe e dados).

ATM Padrão internacional para relé de célula no qual vários tipos de serviços (como voz, vídeo ou dados) são transportados em células de comprimento fixo (53*bytes*).

ATM fórum Organização internacional fundada em conjunto no ano de 1991 pelas empresas *Cisco Systems, NET/Adaptive, Northern Telecom e Sprint* que desenvolve e promove a implementação de acordos baseados em padrões para tecnologia ATM. O ATM fórum se expande sobre os padrões oficiais desenvolvidos por entidades internacionais de padrões e desenvolve acordos de implementação antes dos padrões oficiais.

Auditoria revisão e exame dos registros e das atividades do sistema para avaliar sobre sua confiabilidade, executados com independência.

Autenticação verificação reivindicada de uma identidade. O processo de determinar a identidade de um usuário que esteja tentando alcançar um sistema

Autenticar verificação da identidade de um usuário, de dispositivo, ou de outra entidade em um sistema computadorizado, frequentemente como um pré-requisito a permitir o acesso aos recursos em um sistema.

Autenticidade garantia da identidade dos usuários.

Autoridade de Proteção de Dados (*Data Protection Authority*) é uma autoridade de supervisão interna responsável pela monitoração e implementação da Política de Segurança

Autorização o processo de determinar que tipos de atividades são permitidos. Geralmente, a autorização está no contexto da autenticação: uma vez que você autentica um usuário, está também autorizando acessos e/ou atividades. O poder

dado pela gerência aos indivíduos específicos que permite que aprovem transações, procedimentos, ou sistemas; inclusive conceder direitos de acesso a um usuário, a um programa, ou a um processo.

Backbone uma rede que conecta muitas outras redes e atua como caminho principal para o tráfego entre essas redes.

Backup cópia rotineira dos dados para assegurar a recuperação dos que forem perdidos ou corrompidos.

Baseline nível mínimo aceitável de segurança necessária para proteger um sistema.

Bastion Host um sistema montado para resistir a ataques que é instalado numa rede potencialmente sujeita a ataques. *Bastion hosts* são frequentemente componentes de *firewalls*, ou podem ser servidores Web externos ou sistemas de acesso público.

BGP Protocolo de roteamento entre domínios que troca informações sobre possibilidade de alcance com outros sistemas BGP. O BGP versão 4 (BGP4) é o protocolo de roteamento entre domínios predominante em uso na Internet.

Cache forma de replicação na qual informações aprendidas durante uma transação anterior são usadas para processar transações posteriores.

Cavalo de Tróia (*Trojan horse*) um programa de computador com função aparentemente ou realmente útil que contem as funções (escondidas) adicionais que exploram secretamente as autorizações legítimas do processo provocando perda da segurança. Tipo de ataque em que um *software* aparentemente inofensivo, inicia de forma escondida, ataques ao sistema.

Certificação a avaliação detalhada das características técnicas e não técnicas da segurança de um sistema e de outras proteções, baseadas no processo de credenciamento, que estabelece a extensão a que um projeto e/ou uma execução particulares se encontram em relação a um conjunto especificado de exigências da segurança.

CGI interface usada para executar uma aplicação em um Servidor Web quando requisitada por um cliente.

Chave de criptografia um código utilizado por um algoritmo de criptografia para embaralhar e desembaralhar os dados.

Chave privada código digital usado para descriptografar/criptografar informações e fornecer assinaturas digitais. Essa chave deve ser mantida em segredo pelo proprietário; que tem uma chave pública correspondente.

Chave pública um código digital usado para criptografar/descriptografar informações e verificar assinaturas digitais. Essa chave pode se tornar amplamente disponível; ela tem uma chave privada correspondente.

Checksum um valor calculado a partir de parte de dados que pode ser usado para verificar que o dado não foi alterado.

Classificação determinação da política de segurança que designa níveis de proteção para as informações de acordo com seu grau de confidencialidade (ex.: pública, restrita, secreta, etc.) . Também pode ser usada para os aspectos de integridade e disponibilidade.

Cliente nó ou programa de *software* que solicita serviços de um servidor.

Cliente/Servidor sistemas de rede de computação distribuída nos quais as responsabilidades de transações são divididas em duas partes: o cliente e o servidor. Os clientes se baseiam nos servidores para serviços como armazenamento de arquivos, impressão e capacidade de processamento.

Comprometimento uma violação da política da segurança de um sistema de tal maneira que a divulgação desautorizada da informação sensível pode ter ocorrido

Computer abuse uso impróprio ou mal intencionado dos recursos da tecnologia da informação.

Computer fraud crimes relacionados a computadores envolvendo deliberada alteração, falsificação ou divulgação de dados com a intenção de obter algo de valor (normalmente ganho monetário). Um sistema de computador deve ter sido envolvido na realização ou no encobrimento do ato ou série de atos fraudulentos. Um sistema computadorizado deve ter sido envolvido com a manipulação imprópria de dados de entrada; saída ou resultados; aplicações; arquivos de dados; operações de computador; comunicações; ou hardware de computador ou *software* de sistemas.

Computer Misuse uso impróprio ou desautorizado dos recursos da tecnologia da informação, incluindo o abuso de privilégios por usuários autorizados.

Comunicações Seguras assegura a autenticidade das telecomunicações através de medidas tomadas para negar à pessoas desautorizadas o acesso a estas informações.

Confiabilidade a extensão na qual uma rede ou sistema de computador fornece serviço seguro e isento de erros.

Confiança (*Assurance*) medida de confiança que a arquitetura e as características de segurança de um sistema de informação automatizada garantam e reforcem a política de segurança.

Confidencial um tipo de classificação de informação, que se for divulgada ou usada sem autorização, trará sérios prejuízos para uma organização.

Confidencialidade propriedade de certas informações que não podem ser disponibilizadas ou divulgadas sem autorização para pessoas, entidades ou processos. O conceito de garantir a informação sensível confidencial, limitada para um grupo apropriado de pessoas ou organizações

Contramedidas (*Countermeasure*) mecanismos ou procedimentos colocados num sistema para reduzir o risco (ameaças, impacto ou vulnerabilidades).

Controle de acesso prevenção e controle do uso não autorizado de um recurso. Tarefas executadas por hardware, *software* e controles administrativos para monitorar a operação do sistema, garantindo a integridade dos dados, identificando o usuário, registrando os acessos e as mudanças no sistema e permitindo o acesso aos usuários.

Controle de acesso discrecionário os meios de restringir o acesso aos objetos baseados na identidade do usuário, no que ele sabe (senha), processos e/ou os grupos a que ele pertence. Os controles são discrecionários no sentido que um

objeto com alguma permissão de acesso é capaz de passar essa permissão (talvez indiretamente) para todo sistema. Compare "o controle de acesso imperativo".

Controles procedimentos usados para controlar o sistema de tal maneira que ele esteja de acordo com critérios especificados. Qualquer ação, procedimento, técnica ou qualquer outra medida que reduza a vulnerabilidade de uma ameaça a um sistema.

Credenciamento Declaração formal de uma autoridade credenciadora que o sistema está aprovado para operar num modo seguro, usando um conjunto de proteções prescritas. Credenciamento é a autorização gerencial para operação de um sistema e é baseado numa certificação de processos assim como outras características gerenciais. A declaração de credenciamento determina as responsabilidades de segurança de acordo com padrões aprovados por autoridades e mostra que cuidados devem ser tomados com a segurança.

Criptografia assimétrica sistema de criptografia que é usado para criptografar uma mensagem com uma chave diferente da que é usada para descriptografar a mensagem.

Criptografia princípios, maneiras e métodos para transformar ininteligível informações, e para restaurar informação criptografada para uma forma inteligível.

Criptografia simétrica sistema de criptografia que proporciona caráter confidencial aos dados. Quando duas estações finais usam a criptografia simétrica elas devem concordar sobre o algoritmo a usar e sobre à chave de criptografia que compartilharão.

Cryptographic Checksum uma função de sentido único aplicada a um arquivo para produzir uma "impressão digital original" do arquivo para uma futura referência. Os sistemas *checksum* são os primeiros meios de detectar alterações no *filesystem* no UNIX.

Defense in Depth (Defesa em profundidade) método de segurança em que cada sistema da rede é configurado para o mais alto nível de segurança. Pode ser usado junto com um *firewall*.

DES Algoritmo de criptografia simétrico com chave de 56 *bits*. Existe também uma variação chamada 3DES ou triplo DES, em que se usa três vezes à chave de 56*bits*. Mesmo resultando em uma chave de 168 *bits*, um tipo de ataque chamado "*meet in the middle*" pode quebrar um triplo DES com o mesmo esforço que seria necessário para quebrar um algoritmo de 112 *bits*. Algoritmo padrão de criptografia desenvolvido pelo *National Bureau of Standards* dos EUA

Desafio/Resposta (Challenge/Response) uma técnica de autenticação na qual um servidor emite um desafio desconhecido ao usuário, que computa uma resposta usando algum processo do *token* de autenticação.

Desastre uma circunstância em que um negócio é julgado incapaz de funcionar em consequência de alguma ocorrência natural ou criada.

Descriptografia aplicação inversa de um algoritmo de criptografia a dados criptografados, restaurando assim os dados a seu estado original não-criptografado.

Detecção de Intrusos a detecção dos arrombamentos ou das tentativas de arrombamento por processos manuais ou através dos sistemas que operam sobre os registros ou a outra informação disponível na rede.

DHCP protocolo padrão da Internet que permite que endereços IP sejam colocados em pool e atribuídos como necessários aos clientes. Oferece mecanismo para alocação dinâmica de endereços IP, a fim de minimizar a configuração e permitir que os endereços sejam reutilizados quando os *hosts* não necessitarem mais deles. Definido na RFC 2131.

Diffie-Hellman um algoritmo assimétrico que permite um acordo de chaves: as duas partes trocam suas chaves públicas e as usam em conjunto com suas chaves privadas para gerar uma terceira chave secreta compartilhada. Um curioso que veja as chaves públicas mas não tenha o acesso à chave confidencial de um ou de outro não pode descobrir a terceira chave compartilhada.

Difusão mensagem enviada a todos os nós em uma rede. Comparar com *multicast* e *unicast*.

Disponibilidade a informação deve ser entregue para a pessoa certa, quando ela precisar.

Disponibilidade prevenção de interrupções na operação de todo o sistema (hardware + *software*); uma quebra do sistema não deve impedir o acesso aos dados.

DNS sistema usado na Internet para traduzir nomes de nós de rede em endereços.

DNS Spoofing usar o endereço DNS de um outro sistema corrompendo o cache do sistema da vítima, ou comprometendo um DNS para um domínio válido.

DSA é um algoritmo assimétrico que permite criar assinaturas digitais.

DSS padrão do governo dos EUA que combina DSA e SHA-1 para especificar um formato para assinatura digital.

Dual Homed *Gateway* É um sistema que tem duas ou mais interfaces de rede, cada uma delas conectada a uma rede diferente. Nas configurações de *firewall*, normalmente atua bloqueando ou filtrando algum ou todo o tráfego que tenta passar de uma rede para outra.

Eavesdropping interceptação de tráfego de informação de uma maneira desautorizada com o uso dos métodos à exceção de escuta clandestina.

Embezzlement o roubo de propriedade por alguém a quem a detinha.

Endereço IP endereço de 32 *bits* atribuído a *hosts* que usam TCP/IP. Um endereço IP pertence a uma entre cinco classes (A, B, C, D e E) e é escrito sob a forma de quatro octetos separados por pontos (formato decimal com pontos). Cada endereço consiste em um número de rede, um número de sub-rede opcional e um número de *host*.

Endereço MAC endereço padronizado da camada de enlace de dados exigido para cada porta ou dispositivo que se conecta a uma LAN. Outros dispositivos na rede usam esses endereços para localizar portas específicas na rede. Os endereços MAC têm 6 *bytes* de comprimento e incluem um código de fabricante de 3 *bytes* que é controlado pelo IEEE.

End-to-End Encryption proteção por criptografia de uma informação veiculada através de um sistema de telecomunicações do ponto de origem até o ponto de destino.

Estação de trabalho computador cliente em uma LAN ou WAN que é usado para executar aplicativos e está conectado a um servidor do qual ele obtém dados compartilhados com outros computadores. O termo é também usado para descrever um PC de alto preço que utiliza um microprocessador de alto desempenho e uma arquitetura proprietária para criar o que alguns chamam de sistema "aberto".

Failure access um acesso desautorizado e geralmente inadvertido aos dados resultando de uma falha de hardware ou do *software* no sistema.

FDDI padrão de LAN que especifica uma rede de passagem de símbolos de 100Mbps usando cabos de fibra ótica e uma arquitetura de anel duplo para proporcionar redundância.

File protection a combinação de todos os processos e procedimentos estabelecidos num sistema de informações, que foram especificados para inibir acessos não autorizados, contaminação ou eliminação de um arquivo.

Firewall roteador ou servidor de acesso remoto (ou vários roteadores ou servidores de acesso) designado como *buffer* entre redes conectadas. Um *firewall* utiliza listas de acesso e outros métodos para garantir a segurança de uma rede. Sistema ou combinação de sistemas que protege a fronteira entre duas ou mais redes.

***Firewall* a nível de aplicação** (*Application-Level Firewall*) um sistema de *firewall* em que o serviço é fornecido pelos processos que mantêm o estado da conexão de TCP. *Firewalls* a nível de aplicação frequentemente reendereçam o tráfego de modo que o tráfego que sai pareça ter sido originado do *firewall*, ao invés do *host* interno.

Five Domains um tipo de modelo para a segurança da rede de uma organização que inclui a proteção para o acesso Internet, o *workgroup* LANs, usuários móveis, e escritórios remotos, assim como a integração total de uma estratégia detalhada de segurança na infraestrutura da organização.

Frame Relay protocolo comutado da camada de enlace de dados padrão do mercado que manipula vários circuitos virtuais entre dispositivos conectados. O Frame Relay é mais eficiente que o X.25, o protocolo para o qual ele é geralmente considerado um substituto.

Fraud manipulação de um sistema de informações de negócios em uma tentativa de fazer mal uso de algo de um empregador ou de um colega de trabalho.

FTP protocolo de aplicativo, parte da pilha de protocolos TCP/IP, usado para transferir arquivos entre nós de redes. O FTP é definido na RFC 959.

Full-duplex capacidade de transmissão de dados simultâneos entre uma estação de envio e uma estação receptora.

Functionality diferentemente da garantia, é o comportamento funcional de um sistema. As exigências da funcionalidade incluem, por exemplo: confidencialidade, integridade, disponibilidade, autenticação e não repúdio.

Gateway função de interligar redes distintas (usando protocolos distintos, com características distintas). Atua em todas as camadas do modelo de referência OSI, resolvendo problemas de diferença entre as redes que interliga, tais como: tamanho dos pacotes que transitam nas redes, forma de endereçamento, temporizações, forma de acesso, padrões de linguagem interna de formato de correios eletrônicos etc.

Gerenciamento de Mudanças o conjunto de procedimentos apropriados para controlar mudanças num sistema de hardware e estrutura de *software* com o intuito de assegurar que as mudanças não permitirão violações da política de segurança de sistemas.

Grupo de acesso classe para qual um usuário, um programa ou um processo num sistema de informação é atribuído baseado nos recursos que ele pode usar.

Hacking uma tentativa desautorizada em alcançar uma base de dados de um sistema. Usado frequentemente na referência a uma pessoa que tente entrar numa base de dados de um sistema, de uma posição remota passando pela rede controla.

Handshake processo pelo qual duas entidades de protocolos se sincronizam durante o estabelecimento de uma conexão.

hashing functions um conjunto de funções usadas para criar e destruir objetos hash.

High Level Security Policy uma indicação geral da segurança pretendida aponta para o controle forte da disponibilidade, confidencialidade e integridade da informação.

Host 1. qualquer computador de uma rede que usa o protocolo internet (IP); 2. um computador que abriga sites ou diretórios de arquivos para download.

***Host*-based Security** a técnica de proteger um sistema individual do ataque. Segurança baseada no *host* depende do sistema operacional e da versão.

HTML linguagem de descrição de páginas para criar arquivos que podem ser formatados e exibidos por navegadores da Web.

IANA Organização operada sob patrocínio da ISOC que delega autoridade para alocação do espaço de endereços IP, atribuição de nomes de domínios e atribuição de números de sistemas autônomos para o NIC e outras organizações. A IANA também mantém um banco de dados de identificadores de protocolos atribuídos usados na pilha de protocolos TCP/IP.

ICMP protocolo TCP/IP da camada de rede que informa erros e oferece outras informações relevantes ao processo de pacotes IP. Documentado na RFC 792.

IEEE organização profissional cujas atividades incluem o desenvolvimento de padrões de comunicações e redes. Os padrões de LANs do IEEE são os padrões de LANs predominantes hoje em dia.

IETF força-tarefa que consiste em numerosos grupos de trabalho responsáveis pelo desenvolvimento de padrões da Internet e TCP/IP.

IGMP usado por *hosts* IP para informar sua condição de membros de grupos de *multicast* a um roteador de *multicast* adjacente. Definido na RFC 1112.

Impacto a consequência para uma organização da perda de confidencialidade, disponibilidade e/ou integridade de uma informação. O impacto deve ser analisado quanto a modificação, destruição, divulgação ou negação. Relaciona-se a imagem da empresa, ao dano, a perdas financeiras ou legais e a outros problemas poderiam ocorrer como consequência de uma ruptura da segurança.

Informação controlada toda informação não pública e que esteja formalmente classificada.

Informação de Autenticação Informação usada para estabelecer a validade de uma identidade reivindicada.

Insider Attack um ataque originado de dentro da rede protegida

Integridade 1. garantia da veracidade da informação, que não pode ser corrompida (alterações acidentais ou não autorizadas). 2. Manutenção de dados protegidos enquanto eles atravessam a rede.

Internet termo usado para fazer referência à maior inter-rede global, que conecta centenas de milhares de redes de todo o mundo usando a pilha de protocolos TCP/IP.

Inter-rede coleção de redes interconectadas por roteadores.

IP protocolo da camada de rede na pilha TCP/IP que oferece um serviço de inter-rede sem conexões. O IP fornece recursos para endereçamento, especificação de tipo de serviço, fragmentação e remontagem, e ainda segurança, Definido pela RFC 791.

IP Splicing/Hijacking um ataque no qual uma sessão ativa, já estabelecida, é interceptado pelo atacante. Os ataques deste tipo podem ocorrer depois que uma autenticação foi feita, permitindo ao atacante passar-se pelo papel de um usuário já autorizado.

IP Spoofing - Um ataque em que um sistema assume ilicitamente a personalidade de outro sistema usando seu endereço de rede IP.

IPSec um conjunto de padrões abertos que mantém o caráter confidencial dos dados, a integridade de dados e a autenticação entre parceiros não hierárquicos participantes da camada IP. Documentado nas RFCs 1825 a 1829.

IPv6 substituto para a versão atual do IP (versão 4), O IPv6 inclui suporte para ID de fluxo no cabeçalho do pacote, que pode ser usado para identificar fluxos,

ISDN protocolo de comunicações, oferecido pelas companhias telefônicas, que permite às redes telefônicas transportarem dados, voz e outros tipos de tráfego de origem.

ISO organização internacional responsável por uma ampla faixa de padrões, inclusive padrões relevantes para redes. A ISO desenvolveu o modelo de referência de redes OSI.

ISP empresa que fornece acesso à Internet para outras empresas e indivíduos.

Kerberos sistema de autenticação que oferece segurança de usuário para *host* a protocolos da camada de aplicação, como FTP e Telnet.

LAN redes de dados de alta velocidade e baixa taxa de erros que cobre uma área geográfica relativamente pequena (até alguns milhares de metros).

Linha dedicada linha de transmissão reservada por uma operada de comunicações para uso particular de um cliente.

Link canal de comunicações de rede que consiste em um circuito ou caminho de transmissão e em todos os equipamentos relacionados entre um transmissor e um receptor.

Log arquivos onde o *firewall* (informações sobre tentativas de acesso não autorizadas) e o *proxy* (é configurável, como por ex.: os endereços acessados, tempo de acesso etc.) armazenam informações sobre acesso de usuários.

Log Retention tempo pelo qual as trilhas de auditoria são retidas e mantidas.

Logging processo de estocagem de informações sobre eventos que ocorreram num *firewall* ou numa rede.

Loophole um erro de omissão ou descuido no *software* ou hardware que permite passar pela política de segurança do sistema.

MAC 1. a mais baixa das duas subcamadas da camada de enlace de dados definidas pelo IEEE. A subcamada MAC manipula o acesso à mídia compartilhada. 2. um código de autenticação de mensagem: um algoritmo que usa um algoritmo seguro hash e uma chave secreta para garantir a integridade de uma mensagem. O remetente cria um MAC usando a mensagem e à chave secreta; o receptor verifica a mensagem com a mesma chave secreta. Ninguém que não conheça à chave secreta pode modificar a mensagem, porque não podem produzir um MAC que possa ser verificado com essa chave.

Malicious logic hardware ou *software* introduzido intencionalmente e maliciosamente num sistema com a finalidade de quebrar de alguma maneira a segurança do mesmo (ex.: cavalo de Tróia).

MAN rede que abrange uma área metropolitana. Em geral, uma MAN se estende por uma área geográfica maior que uma LAN, mas abrange uma área geográfica menor que a de uma WAN.

Mandatory access control restrição de acesso a objetos de acordo com a sensibilidade de uma informação. Os controles são mandatórios porque um objeto com certas permissões não pode passar estas permissões para outro objeto.

Máscara de sub-rede máscara de endereço de 32 *bits* usada no IP para indicar os *bits* de um endereço IP que estão sendo usados como endereço de sub-rede.

MD5 algoritmo seguro *hash* criado por Ron Rivest.

Mecanismo de controle de acesso características de *software* ou hardware, procedimentos operacionais, procedimentos de gerenciamento e várias outras combinações destes itens para detectar e prevenir acessos não autorizados e permitir acessos autorizados num sistema automatizado.

MIB banco de dados de informações de gerenciamento de rede, utilizado e mantido por um protocolo de gerenciamento de rede como o SNMP.

MOSPF protocolo de roteamento de *multicast* entre domínios usado em redes OSPF. São aplicadas extensões ao protocolo básico de *unicast* OSPF para admitir o roteamento de *multicast* IP. Definido pela RFC 1584.

Multicast mensagem que é enviada a um subconjunto de nós em uma rede.

Multicast IP técnica de roteamento que permite a propagação do tráfego IP de uma única origem para uma série de destinos. Em vez de enviar um pacote a cada destino, um pacote é enviado a um grupo de multicast identificado por um único endereço IP do grupo de destino.

NAT mecanismo para reduzir a necessidade de endereços IP globalmente exclusivos. A NAT permite que uma organização com endereços não globalmente exclusivos se conecte à Internet traduzindo esses endereços em endereços globalmente roteáveis.

Negação de Serviço o impedimento do acesso autorizado aos recursos ou o retardamento de operações críticas por tempo.

NetBIOS API usada por aplicativos em uma LAN para solicitar serviços de processos de rede de nível mais baixo. Esses serviços incluem o estabelecimento e o término de sessões e a transferência de informações. O NetBIOS é usado por sistemas operacionais de rede como *LAN Manager, LAN Server, Windows NT e Windows for Workgroups*.

NFS conjunto de protocolos de sistemas de arquivos distribuídos que permite acesso a arquivos remotos através de uma rede.

ODBC é usada comumente para conectar bancos de dados cliente/servidor.

OSI modelo de referência de arquitetura de rede que consiste em sete camadas cada uma das quais especifica determinadas funções de rede, como endereçamento, controle de fluxo, controle de erros, encapsulamento e transferência confiável de mensagens. O modelo de referência OSI é usado universalmente como um método para ensinar e compreender a funcionalidade de rede.

OSPF Algoritmo hierárquico de estado de link de roteamento interior proposto como sucessor do RIP na comunidade da Internet. As características do OSPF incluem o roteamento de custo mínimo, o roteamento de vários caminhos e o balanceamento da carga. Definido na RFC 2178.

Pacote agrupamento lógico de informações que inclui um cabeçalho contendo informações de controle e (normalmente) dados do usuário. Os pacotes são usados frequentemente como referência a unidades de dados da camada de rede.

Pacote de Criptografia (Cipher Suite) um método de criptografia SSL que inclui o algoritmo de troca de chave, o algoritmo de criptografia simétrica e o algoritmo *hash* de segurança, usados para proteger a integridade e confidencialidade das comunicações.

Payload número de *bytes* que seguem ao cabeçalho do IPv6. (tamanho do próximo pacote de dados).

PCT protocolo desenvolvido pela Microsoft como sucessor do SSL 2.0 para criptografar comunicações.

Penetration acesso bem sucedido, porém não autorizado, em um sistema de informação.

Penetration testing teste de segurança em que é avaliada a possibilidade de entrada não autorizada em um sistema.

Perimeter-based Security técnica de proteção de uma rede em que são controlados os acessos em todos os pontos de entrada e saída da rede.

Personal Security procedimentos estabelecidos para assegurar que todos que tenham acesso a qualquer informação sensível tenham realmente autorização para tanto.

Physical Security medidas usadas para garantir a proteção física dos recursos contra ameaças voluntárias e involuntárias (ex.: incêndio, invasões, acidentes, etc.).

PKCS The Public Key Cryptography Standards conjunto de especificações criadas pela RSA para padronizar os formatos e operações de criptografia.

PKCS#1 RSA Encryption Standard especificação de padrão de dados para o protocolo RSA, incluindo o padrão para criptografia e assinatura digital RSA e padrão para estocagem de chaves públicas e privadas.

PKCS#10 Certification Request Syntax Standard especificação de um padrão para codificar requisições de certificados, incluindo o nome da pessoa que requisita o certificado e sua chave pública.

PKCS#5: Password-Based Encryption Standard especificação de um padrão para proteção de dados para ser usar a criptografia baseada em senha com o DES.

PKCS#8: Private-Key Information Syntax Standard especificação de um padrão para estocagem de chaves privadas, incluindo a vantagem de criptografá-las com PKCS#5.

Plano de ação uma relação abrangente mas não detalhada dos objetivos, estratégias, ações, agendas e responsabilidades para um plano estratégico de tecnologia da informação.

Plano de contingência é um plano para situações de emergência, operações de backup, e recuperação após desastre, mantido por uma atividade que faz parte de uma programa de segurança que garanta a disponibilidade dos recursos críticos e facilite a continuidade de operações numa situação de emergência. Sinônimo de plano de desastre e plano de emergência.

Plano de recuperação de bancos de dados (*Database Recovery Plan*) plano para restaurar um banco de dados em sua operação normal depois da ocorrência de um erro.

Política de Segurança Corporativa conjunto de diretrizes, normas e procedimentos que regulam como os ativos, incluindo informação sensível, serão gerenciados, protegidos e distribuídos para os usuários de uma organização.

Porta 1. Interface em um dispositivo de interligação de redes (como um roteador). 2. Em terminologia IP, um processo de camada superior que recebe informações de camadas inferiores. As portas são numeradas, e cada porta numerada está associada a um processo específico.

PPP protocolo que oferece conexões de roteador para roteador e de *host* para redes sobre circuitos síncronos e assíncronos. O PPP foi projetado para funcionar com vários protocolos da camada de rede, como IP e IPv6.

PPTP protocolo que permite o encapsulamento de multiprocolos.

Privilégio de acesso é a configuração de direitos de acesso +permitidos (ex.: ler, escrever, anexar, executar, apagar, criar, modificar) a um recurso, programa ou arquivo.

Programa de contingência atividades e procedimentos diários (ex.: *backup* dos arquivos críticos) que cumpram as exigências de recuperabilidade dos sistemas.

Protocolo descrição do método pelo qual se comunicam os computadores ligados em rede.

Proxy ARP variação do protocolo ARP na qual um dispositivo intermediário (por exemplo, um roteador) envia uma resposta ARP em nome de um nó final ao *host* solicitante. Definido na RFC 1027.

Pull em aplicações de disseminação de informações cliente/servidor, a solicitação de dados de outro computador ou aplicativo. A WEB se baseia na tecnologia *pull* na qual um cliente usa um navegador para solicitar (puxar) uma página da WEB.

Push em aplicações de disseminação de informação cliente/servidor, envia dados a um cliente sem que este os solicite. Cada vez mais, as empresas estão utilizando tecnologias *push* para entregar dados personalizados aos usuários sem que os usuários solicitem esses dados explicitamente. Como exemplo a distribuição de notícias e informações sobre o mercado de capitais que são entregues diariamente.

RADIUS protocolo e banco de dados para autenticação de usuários, controle de tempos de conexão e autorização de serviços permitidos aos usuários. Um servidor de acesso remoto atua como um cliente de um servidor RADIUS.

RARP Protocolo em uso na pilha TCP/IP que fornece um método para uma estação sem disco determinar seu endereço IP quando seu endereço MAC é conhecido.

RC4 - Algoritmo simétrico desenvolvido por Ron Rivest que pode usar chaves de tamanho variável. Usualmente usado com 40 *bits* ou 128 *bits*

Recovery procedures ações necessárias para restaurar a capacidade de um sistema computacional e seus arquivos após uma falha do sistema.

Replay uma mensagem, ou parte dela, é interceptada, e posteriormente transmitida para produzir um efeito não autorizado.

Residual risk risco ainda existente depois de terem sido aplicadas medidas de segurança.

Retorno de ligação (*Call back*) procedimento para identificar um terminal remoto. No procedimento *call back*, o *host* desconecta a ligação logo após a chamada e a seguir liga para o número de telefone autorizado do terminal remoto para restabelecer a conecção.

RFC série de documentos escritos pela IETF como principal meio de comunicação de informações sobre a Internet e os protocolos TCP/IP. As RFCs estão disponíveis on-line a partir de numerosas fontes.

RIP protocolo de roteamento interior de vetor de distância fornecido com sistemas UNIX BSD e amplamente utilizado nos primeiros anos de Internet. Definido na RFC 1058 e na RFC 1723.

Rota caminha através de uma inter-rede.

Roteador dispositivo da camada de rede que usa uma ou mais métricas para determinar o caminho ótimo ao longo do qual o tráfego de rede deve ser encaminhado. Os roteadores encaminham pacotes de uma rede à outra com base nas informações da camada de rede. Também executam serviços na camada de rede.

RSA algoritmo assimétrico que permite criptografar dados, criar e verificar assinaturas digitais.

Screened *Host* um *host* em uma rede atrás de um *screening router* (filtra pacotes). O grau a que um *host* selecionado pode ser alcançado depende das regras da seleção no *router*.

Screened Subnet sub-rede atrás de *screening router* (filtra pacotes).

Screening *Router* um *router* configurado para permitir ou negar o tráfego baseado em um jogo de regras de permissão (filtro de pacotes) instaladas pelo administrador.

Secure Hash Algorithm algoritmo que cria a partir da mensagem original, uma assinatura digital que garante a autenticidade da mensagem.

Security evaluation uma avaliação do grau de confiança que pode ser colocado nos sistemas para a manipulação segura da informação sensível. Um tipo, uma avaliação do produto, é uma avaliação executada nas características do hardware e do *software* e em garantias de um produto de computador a partir de uma perspectiva que exclua o ambiente da aplicação. O outro tipo, uma avaliação de sistema, é feito com a finalidade de avaliar proteções de segurança de um sistema com respeito a uma missão operacional específica e é a etapa principal no processo da certificação.

Security measures elementos de *software*, hardware ou procedimentos que são incluídos num sistema para satisfazer as especificações de segurança.

Security perimeter o limite onde os controles da segurança devem de fato proteger recursos.

Security requirements baseline descrição dos requerimentos mínimos necessários para um sistema manter um nível aceitável de segurança.

Servidor de acesso remoto servidor de comunicações que conecta nós remotos ou LANs a uma inter-rede. Geralmente admite serviços de terminal padrão – como Telnet – bem como serviços de nós remotos, tradução de protocolos e roteamento assíncrono.

Servidor nó ou programa de *software* que fornece serviços a clientes.

Simulação processo de usar *software* e modelos matemáticos para analisar o comportamento de um sistema sem exigir que um sistema real seja construído.

Sinal de Autenticação (*Authentication Token*) um dispositivo portátil usado para autenticar um usuário. O símbolo do autenticação opera por desafio/resposta, por sequências de código baseado em tempo, ou por outras técnicas.

SMTP protocolo da Internet que fornece serviços de correio eletrônico.

SNA arquitetura de grandes redes, complexa, rica em recursos, desenvolvida nos anos 70 pela IBM para comunicação entre terminais e mainframes.

SNMP protocolo de gerenciamento de rede para redes TCP/IP. O SNMP oferece um meio para monitorar e controlar dispositivos de rede e para administrar configurações, coleta de estatísticas, desempenho e segurança.

Social Engineering um ataque baseado em enganar usuários ou administradores. Os ataques de engenharia social são realizados tipicamente fazendo-se passar por um usuário autorizado para tentar ganhar o acesso ilícito aos sistemas. Também são feitos iludindo os responsáveis por liberação de acesso baseados na confiança.

Spoofing 1. Esquema usado por roteadores para fazer um *host* tratar uma interface como se ela estivesse ativa e dando suporte a uma sessão. O roteador envia respostas de *spoofing* a mensagens *Keepalive* do *host* para convencer o *host* de que a sessão ainda existe. 2. Ação de um pacote que afirma ilegalmente ter vindo de um endereço a partir do qual na realidade não foi enviado. O *spoofing* é projetado para iludir mecanismos de segurança de rede como filtros e listas de acesso.

Spoofing tentativa de ganhar acesso ao sistema iludindo ser um usuário autorizado.

SQL linguagem padrão internacional para definir e obter acesso a bancos de dados relacionais.

SSL tecnologia de criptografia para a WEB usada para transações seguras como a transmissão de números de cartões de crédito para aplicativos de comércio eletrônico.

Sub-rede em redes IP, uma rede que compartilha um endereço de sub-rede em particular. As sub-redes são redes arbitrariamente segmentadas por um administrador de rede para fornecer uma estrutura de roteamento hierárquica de vários níveis, ao mesmo tempo que isola a sub-rede da complexidade de endereçamento de redes associadas.

System Security Administrator pessoa responsável pela segurança de um sistema de informação automatizado.

Tabela de roteamento é um banco de dados que resume os segmentos da rede e que fica armazenada no roteador. Funciona como um mapa da rede e instrui o roteador em como chegar a determinado nó nos diversos segmentos físicos que uma rede pode possuir.

Tabela de roteamento tabela armazenada em um roteador ou algum outro dispositivo de interligação e redes que controla as rotas até determinados destinos de rede e, em alguns casos, as métricas associadas com essas rotas.

TACACS protocolo de autenticação que oferece autenticação de acesso remoto e serviços relacionados, como o registro de *log* de eventos. As senhas de usuários são administradas em um banco de dados central, em vez de ficarem em roteadores individuais, proporcionando uma solução de segurança de rede escalonável.

Tampering modificação não autorizada que altera o funcionamento próprio de um equipamento ou sistema de modo que diminua a segurança ou a funcionalidade do mesmo.

TCP protocolo da camada de transporte orientado a conexões que oferece transmissão de dados *full-duplex* confiável. O TCP faz parte da pilha de protocolos TCP/IP.

Telnet protocolo padrão de emulação de terminal de pilha de protocolos TCP/IP. O Telnet é usado para conexão de terminal remoto, permitindo aos usuários efetuarem logon em sistemas remotos e usar recursos como se estivessem conectados a um sistema local. Definido pela RFC 854.

Teste de consentimento um tipo de teste de auditoria que é realizado para determinar se os procedimentos e os controles que são recomendados estão sendo seguidos plenamente.

Threat (Ameaça) violação da segurança. Uma ação ou evento que muito prejudica a segurança.

Topologia organização lógica de nós de rede e mídia dentro de uma estrutura de rede.

Trap door um mecanismo escondido do *software* ou do hardware. É ativado de alguma maneira aparentemente inocente; por exemplo, uma sequência chave "aleatória" especial em um terminal. Os desenvolvedores do *software* introduzem frequentemente portas de armadilha em seu código para permiti-los de reentrar no sistema e de executar determinadas funções. Sinônimo de "a porta de trás".

Trilha de auditoria histórico das transações de sistemas que estão disponíveis para a avaliação a fim a provar a correção de sua execução comparado com os procedimentos ditados pela política de segurança. Relacionam-se a uma chave ou transação que permite que as quebras na segurança sejam detectáveis.

TTL campo em um cabeçalho IP que indica por quanto tempo um pacote é considerado válido.

Tunneling ***Router*** um roteador ou um sistema capaz rotear o tráfego criptografando-o e encapsulando-o através da rede. Ou de outra maneira, fazendo o processo inverso.

UDP protocolo da camada de transporte sem conexões na pilha de protocolos TCP/IP. O UDP é um protocolo simples que troca datagramas sem reconhecimentos ou garantia de entrega, exigindo que o processamento de erros e a retransmissão sejam manipulados por outros protocolos. O UDP é definido na RFC 768.

Unicast mensagem enviada a um único nó de rede.

USENET

Vazamento relativo a dados sensíveis vazados para o público externo ou pessoas não autorizadas à receber tais informações.

Vírus uma classe do *software* malicioso que tem a habilidade de auto replicar e infectar partes do sistema operacional ou dos programas de aplicação, com o intuito de causar a perda ou dano nos dados.

VPN 1. um meio de estabelecer canais de comunicação seguros na Internet usando várias formas de criptografias. Implementa a interligação de redes particulares virtuais. 2. Conjunto de processos e protocolos que permitem a uma organização interconectar com segurança *sites* que fazem parte de uma rede particular por meio de rede pública, como a rede de um provedor de serviços Internet.

Vulnerabilidade probabilidade de uma ameaça transformar-se em realidade.

WAN rede de comunicação de dados que serve a usuários de uma ampla área geográfica e frequentemente usa dispositivos de transmissão fornecidos por operadoras comuns.

Web uma grande rede de servidores da Internet que oferece serviços de hipertextos e outros serviços a terminais que executam aplicativos clientes, como navegadores.

X.509 um padrão que especifica o formato dos certificados digitais, de tal maneira que se possa amarrar firmemente um nome a uma chave pública, permitindo autenticação forte.