FACULDADE DE TECNOLOGIA DE SÃO PAULO

Segurança de Redes

SÃO PAULO
2011

FACULDADE DE TECNOLOGIA DE SÃO PAULO

Segurança de Redes

Roberto Akio Mitshashi

Monografia apresentada à Faculdade de Tecnologia de São Paulo para a obtenção do Grau de Tecnólogo em Processamento de Dados

Prof. Orientador: Paulo Roberto Bernice

SÃO PAULO
2011

SUMÁRIO

LISTA DE FIGURAS

CURRICULUM VITAE

Roberto Akio Mitshashi, nascido em São Paulo em 27 de abril de 1963, cursando Processamento de Dados pela Faculdade de Tecnologia de São Paulo. Atualmente sou Administrador de Rede na Associação Auxiliadora das Classes Laboriosas onde cuido da infraestrutura (rede, servidores) e também suporte aos usuários.

RESUMO

Segurança em Redes é um assunto muito discutido no ambiente corporativo, porque o que está em jogo é o patrimônio tanto material quanto intelectual. Falhas na segurança podem até mesmo danificar a imagem da corporação.

Com a presença da Internet no dia a dia das empresas, aumentam as possibilidades de fraudes eletrônicas e ataques externos que exploram vulnerabilidades dos sistemas utilizados. Portanto o trabalho tem por objetivo mostrar conceitos e a implementação de técnicas para conseguir o máximo possível de segurança dentro de uma rede de computadores, para maximizar a integridade e segurança dos dados da empresa.

ABSTRACT

Network Security is a hot topic in the corporate environment, because what is at stake is both material and intellectual heritage. Security failures can even damage the image of the corporation.

With the presence of the Internet in daily business, the possibilities of electronic fraud and external attacks that exploit vulnerabilities in systems used. So the work is to show concepts and implementation techniques to achieve the maximum possible security within a computer network, to maximize the integrity and security of company data.

1 – INTRODUÇÃO

Ao falar de rede logo se pensa em algum tipo de compartilhamento. Quando trazido para o meio computacional, isto é entendido como compartilhamento de informações e recursos, que podem ser acessados facilmente a partir de vários pontos na rede.

Quando se disponibiliza dados e equipamentos ao alcance de várias pessoas, cria-se um ponto de atenção, a segurança destes dados. Com o aumento da utilização de componentes de redes e da internet, ocorreu um grande crescimento no número de invasões e infecções por vírus, surgiu então à necessidade primária de investimento na área de segurança de redes. Esse investimento tratou desde melhorias nos métodos de desenvolvimento e estruturação dos sistemas de segurança, como criptografia, por exemplo, quanto o investimento em estudo de técnicas de ataques para desenvolvimento de ferramentas específicas para determinados tipos de ataque.

O objetivo deste é abranger os principais conceitos de segurança utilizados atualmente no gerenciamento de redes, além disso, apresentar ferramentas de análise de vulnerabilidades e de varreduras, para alertar aos problemas de gerenciamento de segurança, expondo posteriormente uma forma de minimizar os problemas, oferecendo um modelo baseado em estudos e pesquisas que pode trazer muitos benefícios.

2 – SEGURANÇA DE REDES

Segundo a CERT.br, o número de incidentes reportados em 2010 foi de 142.844 e em 2011 (até junho) foi de 217.840, ou seja um aumento de 65% comparando somente com o primeiro semestre de 2011. Na Figura 1 abaixo podemos acompanhar a evolução dessas ameaças ao longo dos anos, a qual demonstra que as ameaças relacionadas a problemas de segurança estão cada dia mais presentes.

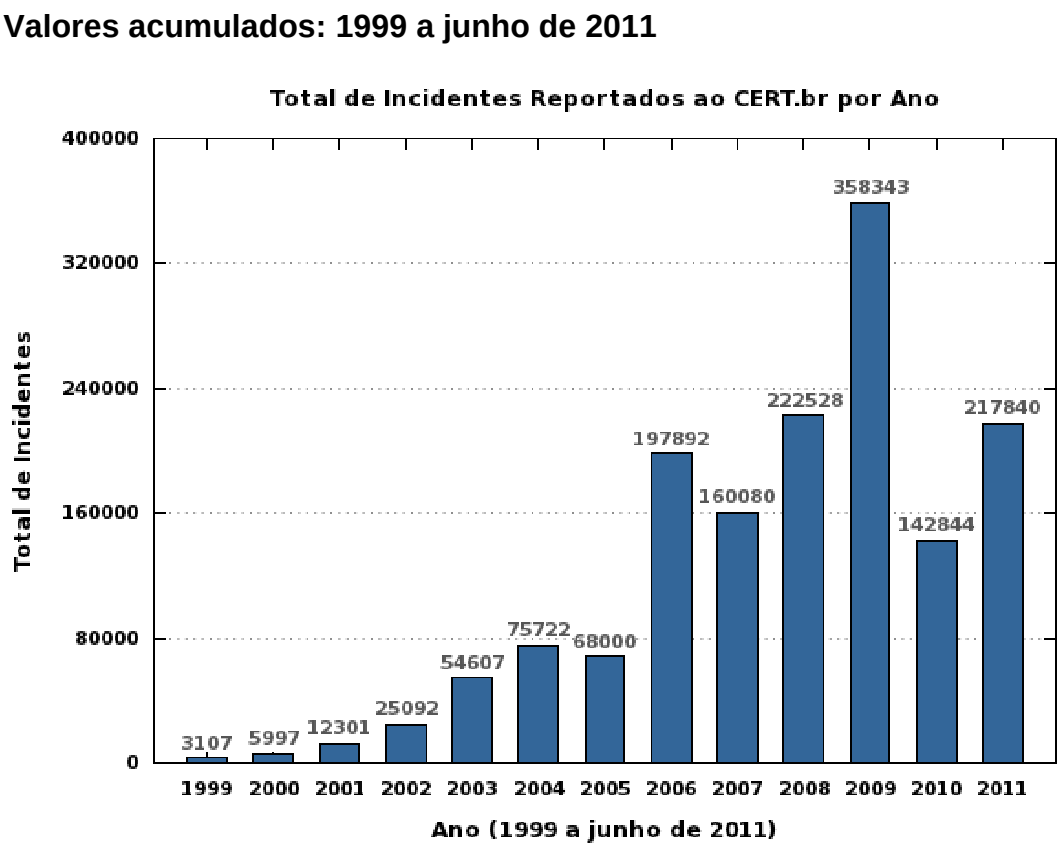

Figura **1 – Total de Incidentes reportados. (Extraído do site WWW.cert.br)**

Incidentes Reportados ao CERT.br -- Abril a Junho de 2011

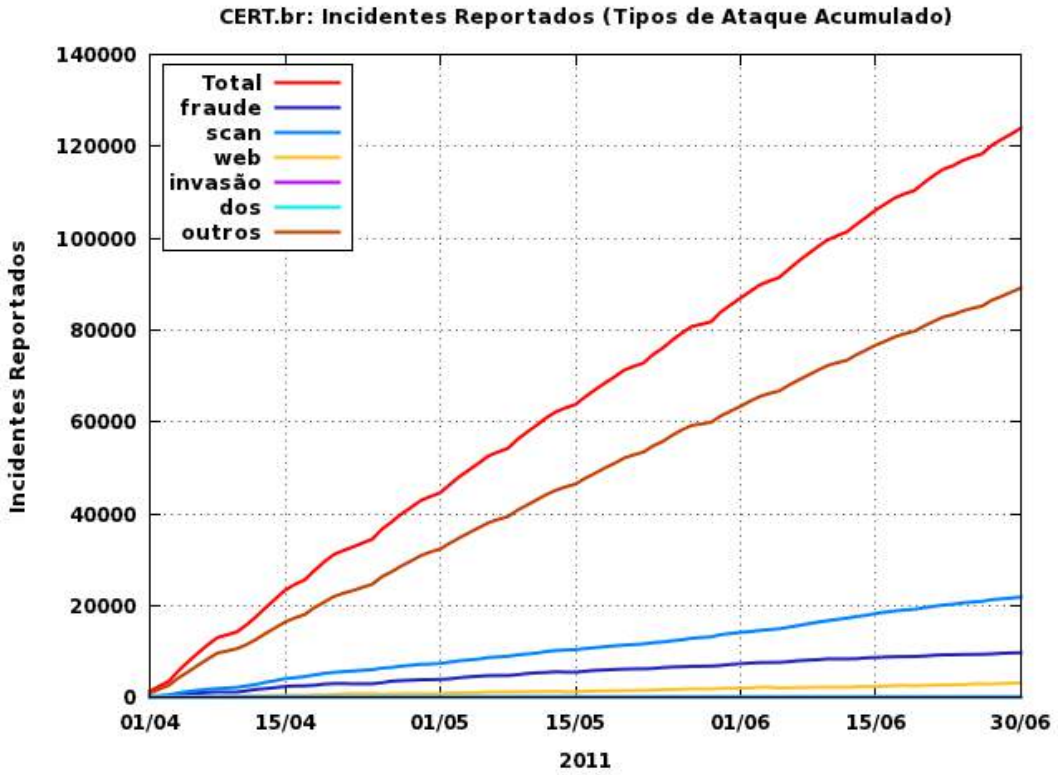

Figura 2 **– Incidentes reportados – Tipos de Ataque Acumulado.**

Este gráfico não inclui os dados referentes à worms.

Legenda:

- **dos** (DoS -- *Denial of Service*): notificações de ataques de negação de serviço, onde o atacante utiliza um computador ou um conjunto de computadores para tirar de operação um serviço, computador ou rede.
- **invasão**: um ataque bem sucedido que resulte no acesso não autorizado a um computador ou rede.
- **web**: um caso particular de ataque visando especificamente o comprometimento de servidores Web ou desfigurações de páginas na Internet.
- **scan**: notificações de varreduras em redes de computadores, com o intuito de identificar quais computadores estão ativos e quais serviços estão sendo

disponibilizados por eles. É amplamente utilizado por atacantes para identificar potenciais alvos, pois permite associar possíveis vulnerabilidades aos serviços habilitados em um computador.

- **fraude**: segundo Houaiss, é "qualquer ato ardiloso, enganoso, de má-fé, com intuito de lesar ou ludibriar outrem, ou de não cumprir determinado dever; logro". Esta categoria engloba as notificações de tentativas de fraudes, ou seja, de incidentes em que ocorre uma tentativa de obter vantagem.
- **outros**: notificações de incidentes que não se enquadram nas categorias anteriores.

Obs.: Vale lembrar que **não se deve confundir scan com scam**. Scams (com "m") são quaisquer esquemas para enganar um usuário, geralmente, com finalidade de obter vantagens financeiras. Ataques deste tipo são enquadrados na categoria fraude. Quando se fala de segurança, leva-se em consideração atingir a menor taxa de risco possível, quando se trata de segurança de dados de uma rede computacional, fala-se da segurança das suas informações e equipamentos compartilhados, ou seja, da menor taxa de risco possível para com estes.

Com o crescimento da utilização de tecnologias computacionais nas corporações e no meio doméstico aumentou-se a necessidade de garantir a confidencialidade, integridade e disponibilidade dos recursos dispostos em rede.

- Confidencialidade — Parte do pressuposto que somente as pessoas autorizadas terão acesso aos dados ou recursos;
- Integridade — Baseia-se no fato de que somente pessoas devidamente autorizadas terão acesso à alteração destes dados;

- Disponibilidade — É a garantia de que as pessoas devidamente autorizadas terão acesso aos dados quando desejarem.

Estas medidas de garantia fazem-se necessárias devido ao grande número de informações confidenciais que trafegam em uma rede.

Devido à amplitude atingida por um sistema computacional em rede e ao aumento do número de serviços operantes, tornou-se a cada dia mais complexo a manutenção da segurança.

2.1 – Por que é necessária a segurança?

Na medida em que os ambientes crescem e se diversificam, e na medida em que a segurança *host a host* se torna mais complexa, cresce o uso do modelo de segurança de rede. Com um modelo de segurança de rede, o controle do acesso é concentrado em seus vários *hosts* e os serviços que eles oferecem, no lugar de fazê-lo um por um. A abordagem de segurança de rede inclui *firewalls* para proteger os sistemas internos e redes, usando um sistema de autenticação forte e encriptação para proteger dados particularmente importantes que transitam na rede. Assim, um local pode obter um tremendo reforço de segurança usando um modelo de segurança de rede.

2.2 – Aplicação em ambiente corporativo

Hoje em dia as empresas procuram manter um nível de segurança que lhes proporcione certa garantia sobre as suas informações confidenciais que trafegam pela rede, isso devido aos prejuízos que estas corporações podem ter numa eventual ocorrência de ataques *hackers* ou de infestações de vírus em suas redes. Com isso as corporações tornaram-se as maiores investidoras e usuárias em ferramentas de segurança de rede.

2.3 – Criptografia

Criptografia é junção das palavras gregas *kriptos* (escondido) e *grapho* (grafia), consiste na codificação de uma mensagem onde somente o remetente e o destinatário conheçam a forma de traduzi-la. Basicamente, utiliza-se de uma chave, a qual contém os parâmetros de conversão de valores de um texto para um modo criptografado e, em contrapartida, existe uma chave correspondente que contém os parâmetros para decriptar essa mensagem, podendo ela ser a mesma do remetente ou até mesmo uma versão inversa.

Por ser uma ciência matemática e não computacional, a criptografia é muito anterior aos computadores. Assim sendo, sua aplicação já existe há séculos na história da humanidade.

Um exemplo foi o modelo criptográfico chamado Júlio Cypher, criado por Júlio César, na época do Império Romano. Como os meios de comunicação eram muito limitados, nesta época se usava um cavalo rápido e um mensageiro para o envio de mensagens no campo de batalha, entretanto sempre se corria o risco de o mensageiro ser interceptado, e com isso o conteúdo das mensagens, crucial para a batalha, cair nas mãos do inimigo.

Júlio Cesar criou um algoritmo que consistia em deslocar as letras da mensagem três posições para a direita de forma a torná-la sem sentido para quem fosse ler. Assim a mensagem ATACAR ficaria transformada em DWDFDU. A decriptação da mensagem consistia em deslocar novamente para a esquerda as três posições.

Esse mecanismo, embora muito simples, foi efetivo por algumas décadas nas comunicações do império, principalmente porque poucas pessoas na época sabiam ler.

Com a criação do computador e devido às pesquisas para melhoria dos métodos, houve no século passado uma grande evolução nas técnicas de criptografia, baseando-se agora em algoritmos matemáticos bastante complexos, até então não utilizados devido ao grande tempo perdido na codificação de pequenas mensagens. Os métodos de criptografia atuais se apóiam no fato da dificuldade encontrada ao tentar se resolver os problemas matemáticos usados na criptografia, mesmo utilizando as ferramentas mais modernas. Isso se deve exatamente ao ponto citado anteriormente: a complexidade dos algoritmos.

Hoje em dia existem basicamente duas formas de utilização e distribuição de chaves de criptografia. Uma delas utiliza a mesma chave para encriptação e decriptação das mensagens, a outra utiliza chaves diferentes. Devido a isso, além de algumas outras características inclusas na criptografia moderna, deve-se subdividi-la em dois grandes Grupos: Sistemas de Criptografia de Chave Simétrica e Sistemas de Criptografia de Chave Assimétrica.

2.3.1 – Sistemas de Criptografia

2.3.1.1 – Sistema simétrico ou de chave privada

O Sistema Simétrico ou de Chave Privada consiste dos conceitos mais tradicionais de criptografia, partindo do pressuposto que a mensagem é encriptada e decriptada utilizando uma chave única, de conhecimento somente das pessoas envolvidas na transmissão, daí vem à denominação de Chave Privada. Essa chave secreta idealmente teria mais segurança se não fosse transmitida eletronicamente, visto que, se houvesse uma forma de transmitir com segurança essa chave, por que não enviar já os dados da mensagem? Uma segurança maior seria garantida se a troca das chaves

fosse feita pessoalmente. Exemplos de estratégia de encriptação de chave privada são os padrões DES e Triple DES, que serão descritos posteriormente.

Uma característica positiva da criptografia simétrica, ou criptografia de chave privada, é a velocidade com a qual são encriptadas e decriptadas as mensagens, sendo constatada sua eficiência em coisas de uso no dia a dia, como, por exemplo, transações seguras na Internet, que em muitas transmissões requerem trocas de senhas temporárias.

Uma desvantagem da aplicação de criptográfica simétrica é o numero de chaves requeridas quando mais que duas pessoas estão envolvidas. Por exemplo, quando duas pessoas estão envolvidas, é necessária apenas uma chave para elas trocarem mensagens secretas, porém, se dez pessoas estão envolvidas, serão necessárias 45 chaves privadas. Chega-se a esse número partindo da função que: n pessoas requerem n* ((n-1) /2) chaves privadas.

2.3.1.1.1 – Algoritmo DES

O Algoritmo DES (*Data Encryption Standard*) é um algoritmo que implementa um sistema simétrico ou de chave privada. Foi criado pela IBM em 1976, atendendo aos pedidos públicos feitos pelo departamento nacional de padrões dos Estados Unidos, NBS(agora o *National Institute of Standards and Technology - NIST*), que buscava um novo padrão de algoritmo de criptografia para ser utilizado na proteção de dados em transações por meios computacionais, ex: transações financeiras do governo dos Estados Unidos. Trata-se de um sistema de codificação simétrico por blocos de 64 bits, dos quais 8 bits (um octeto) servem de teste de paridade (para verificar a integridade da chave). Cada bit de paridade da chave (1 em cada 8 bits) serve para testar um dos bytes da chave por paridade ímpar, quer dizer que cada um dos bits de paridade é ajustado de forma a ter um número ímpar “de 1” no byte a que pertence.

A chave possui então um comprimento “útil” de 56 bits, o que significa que só 56 bits servem realmente no algoritmo.

O algoritmo consiste em efetuar combinações, substituições e permutações entre o texto a codificar e a chave, fazendo de modo a que as operações possam fazer-se nos dois sentidos (para a descodificação). A combinação entre substituições e permutações chama-se **código produzido.**

A chave é codificada em 64 bits e formada por 16 blocos de 4 bits, geralmente notados k_1 à k_{16}. Já que “apenas” 56 bits servem efetivamente para calcular, podem existir 2^{56} (quer dizer $7.2*10^{16}$) chaves diferentes.

O Esquema pode ser visualizado na figura 3, logo abaixo:

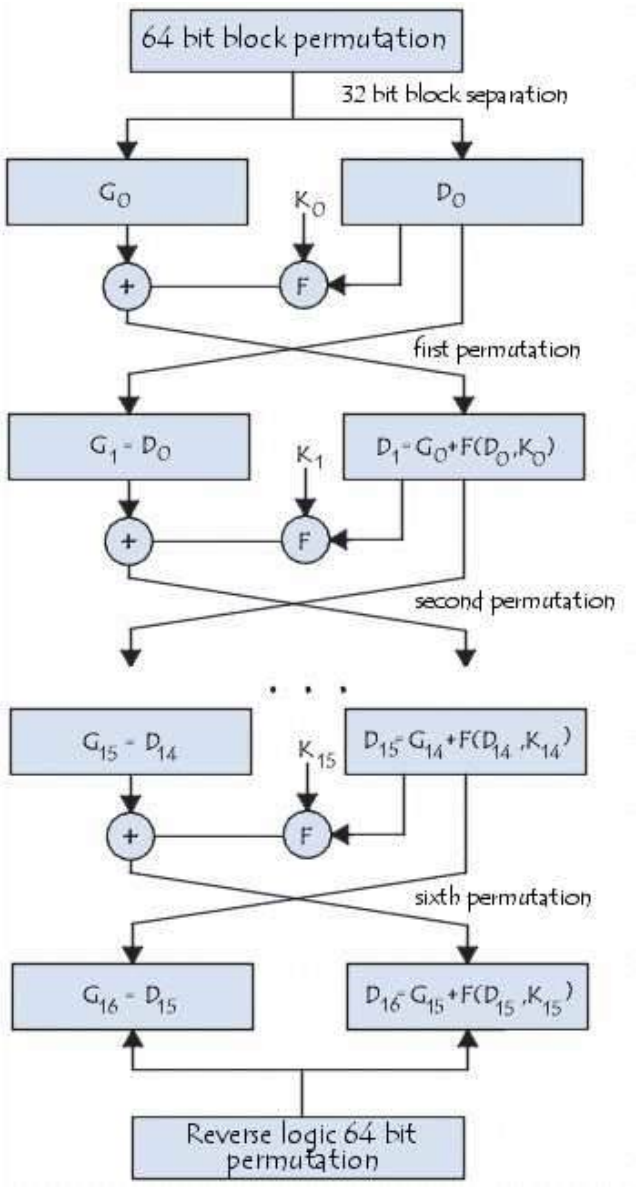

Figura 3: DES Encriptação

As grandes linhas do algoritmo são as seguintes:

- Fraccionamento do texto em blocos de 64 bits (8 octetos);
- Permuta inicial dos blocos;
- Corte dos blocos em duas partes: esquerda e direita, nomeadas G e D;
- Etapas de permuta e de substituição repetidas 16 vezes (chamadas **voltas**);

- Recolamento das partes esquerda e direita, seguidamente permutação inicial inversa.

Durante mais de 20 anos o DES foi o algoritmo mais inteiramente testado, nunca quebrado, porém em 1998 em projeto provido pela *Eletronic Freedom Foundation* (EFF) e patrocinado pelo governo dos Estados Unidos, *John Gilmore e Paul Kocher,* munidos de um super computador de centenas de processadores conseguiram quebrar o código em 56 horas. Esse tempo até hoje foi reduzido para 22 horas. Após todas essas evidências de que o DES havia ficado obsoleto, surgiu à necessidade da criação de novos algoritmos que garantissem mais segurança. Nisso foi sugerido como forma mais rápida e de fácil implementação o uso do DES Triplo, que nada mais é que uma tripla implementação do DES, conforme texto seguinte.

2.3.1.1.2 – DES Triplo

O DES triplo é simplesmente outra modalidade da operação do DES, este faz uso de três chaves de *64-bits*, para um comprimento de chave total de *192 bits*. O código do DES triplo quebra então a chave fornecida pelo usuário em três subchaves, o procedimento para encriptação é exatamente o mesmo que o DES regular, mas repete-se três vezes. Desse passo surge o nome Triplo DES, que consiste em encriptar os dados com a primeira chave, decriptar com a segunda chave e encriptar novamente com a terceira chave. Conseqüentemente, o DES Triplo funciona três vezes mais lento do que o padrão, porém muito mais seguro se usado corretamente.

2.3.2.1.3 – AES

Em Janeiro de 1997 o *NIST* emitiu um pedido público para substituição do DES que estava se tornando obsoleto, seria um projeto para a criação do AES – Algoritmo Padrão de Encriptação Avançada (*Advanced Encryption Standard).* Diversas sugestões foram recebidas, sendo cinco classificadas para a semifinal do processo. Em outubro 2000 o *NIST* anunciou que *Rijndael,* um algoritmo criado por dois criptógrafos belgas, *Joan Daemen e Vincent Rijmen*, foi selecionado como a proposta do *AES*.

2.3.1.2 – Sistema assimétrico ou de chave pública

O Sistema Assimétrico surgiu em 1976, proposto por *Whitfield Diffie* e *Martin E. Hellman.* Esse Sistema foge do esquema tradicional de criptografia descrito anteriormente pelo texto *Sistema Simétrico* ou de *Chave Privada* o qual utiliza a mesma chave para encriptação e decriptação das mensagens. O Sistema Assimétrico por sua vez utiliza uma chave dupla, sendo uma delas de conhecimento publico e outra de conhecimento restrito. Essas chaves trabalham em conjunto e são completamente relacionadas, caso seja feita uma encriptação utilizando a chave pública ela só poderá ser decriptada com sua correspondente privada, caso seja encriptada pela chave privada só poderá ser decriptada pela sua correspondente pública, no exemplo abaixo será demonstrada sua aplicação prática.

Simulando uma comunicação entre dois usuários, denominados *remetente* e *receptor,* para garantir a segurança dos dados no envio da mensagem utilizando o sistema assimétrico de criptografia, o *remetente* deve encriptar a mensagem a ser enviada utilizando a chave publica do *receptor*, assim garantindo que a mensagem será somente decriptada pela chave privada correspondente, em poder do *receptor*. É

possível que o remetente disponibilize uma forma do receptor validar a origem da mensagem, para isto, o remetente após encriptar a mensagem utilizando a chave publica do receptor deverá encriptá-la novamente utilizando a chave privada do remetente, por sua vez ao receber a mensagem, o receptor deverá primeiro validar a origem, buscando decriptar a mensagem através da chave publica do suposto remetente, caso consiga a origem da mensagem esta validada. O próximo passo é decriptar a mensagem utilizando sua chave privada, daí tendo acesso aos dados da mensagem.

Algoritmos que implementam encriptação de chave pública são os padrões RSA e o de Curvas Elípticas.

Uma desvantagem desse sistema em relação ao sistema simétrico é que, mesmo diminuindo o número de chaves privadas a ser trocadas, este apresenta maior lentidão.

2.3.1.2.1 – Algoritmo RSA

O Algoritmo RSA é um algoritmo que implementa sistema assimétrico ou de chave pública. Foi criado pelos pesquisadores do MIT, *Ronald Rivest, Adi Shamir* e *Leonard Adleman* em 1977, RSA corresponde à junção das iniciais do sobrenome dos três. Seu método de implementação utiliza conceitos matemáticos baseados na teoria dos números, com a aplicação de números primos, Máximo Divisor Comum (MDC) e o conceito de relativamente primo, além de sistemas modulares.

A força do RSA consiste no fato da facilidade deste gerar números primos extremamente grandes, o que torna muito difícil a fatoração, que é a característica de segurança do algoritmo. O algoritmo é formado por três estágios, sendo eles, geração de duas chaves (pública e privada), encriptação da mensagem e decriptação da

mensagem, sempre lembrando que a utilização do conceito de chave pública consiste na paridade entre as chaves pública e privada, que devem ser relacionadas.

2.3.1.2.1.1 – Primeiro Estágio – Geração das Chaves

- Gerar dois números primos extremamente grandes, que receberão a denominação de p e q, respectivamente (escolhido, secreto);
- Calcular n a partir do produto de p e q(publico, calculado);
- Calcular e, sendo MDC (F(n), e) = 1; 1 < e < F(n) (público, escolhido);
- Calcular d=e – 1 mod F(n) (escolhido, secreto).

Após esses passos obtém-se a chave pública e a chave privada da seguinte sendo: Chave pública (e, n) e Chave privada (d, n).

2.3.1.2.1.2 – Segundo Estágio - Encriptação

Para encriptar a mensagem aplica-se a chave publica na função, *E(s) = s^e (mod n),* onde *s* seria o valor da mensagem e *e* e *n* representam um par de inteiros que determina a chave pública.

2.3.1.2.1.3 – Terceiro Estágio – Decriptação

Para decriptar a mensagem, aplica-se a chave privada na função, s = *[E(s)]^d (mod n),* onde E(s) é a mensagem encriptada e *d* e *n* representam um par de inteiros que representam a chave privada.

3 – A ARQUITETURA TCP/IP E SUAS FRAGILIDADES

O conjunto TCP/IP foi um dos primeiros contribuidores na consolidação da implementação de níveis de camadas. O TCP/IP é formado por um conjunto de protocolos nos quais os principais são o TCP (*Transfer Control Protocol* que trabalha no nível de transporte) e o IP (*Internet Protocol*, que trabalha no nível de rede).

Na época do desenvolvimento do protocolo TCP/IP não havia preocupação com ataques *hackers*, e a segurança dos dados que trafegavam pela rede era inexistente, podendo-se facilmente descobrir senhas de usuários, pois os pacotes que trafegavam pela rede não eram encriptados, ou seja, podendo facilmente ser descobertos.

Com o desenvolvimento de técnicas e ferramentas para ataques *hacker*, essa vulnerabilidade do protocolo TCP/IP foi facilmente explorada, deixando sistemas vulneráveis a vários tipos de ataques, como os mencionados abaixo:

- IP SPOOFING
- DOS e DDoS
- TROJAN HORSES
- TCP HIJACKING
- SNIFFING
- FLOODING

Nas empresas os trabalhos realizados pelos usuários são oferecidos através de um servidor, com o grande desenvolvimento tecnológico há um grande numero de serviços que podem ser usados e são esses serviços que se tornam alvos de ataques nas redes corporativas. Serviços como:

Exemplo de Serviços

- FTP
- Email
- EDI
- Multimídia
- Teleconferência
- Videoconferência
- WWW
- VPN

3.1 – Algumas Definições Importantes

3.1.1 – Portas

É um numero de 16 bits que identifica processos (também chamados de serviços de rede). O número da porta de origem e o número da porta de destino estão incluídos no cabeçalho de cada segmento TCP ou pacote UDP. Os números de portas de 0 a 1023 são chamados de portas conhecidas (*well-know ports*) e são reservados para as aplicações mais comuns, como *telnet* e *ftp*.

3.1.2 – Sockets

Socket é a combinação de um endereço *IP* com um número de porta, que identifica o processo como único na rede e através deles os programas se comunicam.

Para que haja distinção entre requisições duplicadas tendo como destino o mesmo *host*, a informação referente ao *socket* origem é incluída na requisição.

Estrutura da Requisição:

- Protocolo;
- IP de origem;
- Porta de origem;
- IP de destino;
- Porta de destino.

Os sockets podem ser classificados de várias formas, dentre elas duas são mais conhecidas e mais utilizadas, são os stream_sockets e os datagram_sockets.

Os *Stream Sockets* trabalham sob o protocolo de controle de transmissão, o qual garante que as informações chegarão livres de erros e na mesma ordem que foram enviadas.

Os *Datagram Sockets* não utilizam protocolo de controle de transmissão para o envio de informações, o que não garante que os pacotes chequem na mesma ordem que foram enviadas, porém garante que os pacotes que chegarem estarão livres de erros, pois utilizam o IP.

Em resumo, os *Stream Sockets* utilizam protocolo TCP, que é orientado a conexão, enquanto os *Datagram Sockets* utilizam o protocolo UDP, não orientado a conexão.

3.2 – O ICMP – Internet Control Message Protocol

O ICMP é um protocolo que trabalha na camada de rede e é sempre implementado em conjunto com o protocolo TCP/IP. Tem por função gerar mensagens de testes efetuados através da comunicação entre *hosts*.

Vale lembrar que o objetivo do ICMP não é corrigir eventuais erros de comunicação que ocorram entre os componentes da rede e sim de informá-los, cabe assim ao administrador da rede solucionar a eventual falha apontada.

As mensagens ICMP trafegam pela interligação de redes, não se restringindo somente a roteadores, e diferentemente do que se possa pensar, este não faz interface com programas ou usuários e sim com o IP de destino.

Como exemplos de serviços realizados através de mensagens ICMP, temos os conhecidos PING e TRACEROUTE, que se baseiam em pacotes ICMP para informar os status da comunicação entre *hosts*.

3.2.1 – Formato de mensagens ICMP

O ICMP adotou uma padronização das mensagens de erro de acordo com o tipo de falha ou status que ocasionou o lançamento do pacote. Essas mensagens por si próprias não são capazes de determinar a ação a ser tomada, pois tratam-se de mensagens genéricas, que na maioria das vezes auxiliam o administrador na busca pela causa do erro.

Mensagens ICMP trafegam da mesma forma que outros pacotes de dados, podendo então ser perdidos ou descartados, pois não possuem nenhuma prioridade adicional. Além disso, quando trafegam em uma rede já congestionada, podem causar ainda mais congestionamento**.**

As mensagens ICMP são apontadas de acordo com o modelo abaixo:

Campo Tipo	Tipo de Mensagem ICMP
0	Resposta *ao Echo Reply*
3	Destino Não-acessível
4	Dissipação da Origem
5	Redirecionamento
8	Solicitação de *Echo Request*
11	Tempo Excedido para um datagrama
12	Problema de Parâmetro num Datagrama
13	Solicitação de Indicação de Hora
14	Resposta de Indicação de Hora
15	Solicitação de Informação (Obsoleto)
16	Resposta de Informação (Obsoleto)
17	Solicitação de Máscara de Endereço
18	Resposta de Máscara de Endereço

Formato de Mensagens ICMP

3.3 – Arquitetura IP Security (IPSec)

A tecnologia *IP Security* foi desenvolvida com objetivo de corrigir as vulnerabilidades do protocolo IP, que foi desenvolvido em uma época em que não se pensava ou pouco se desenvolvia em relação à segurança de redes.

Com o crescimento da Internet, problemas relacionados com segurança ficaram cada vez mais críticos, pois com esse crescimento, aumentou-se a exploração das vulnerabilidades apresentadas pelo protocolo IP.

A Arquitetura *IP Security* é na verdade uma estrutura desenvolvida para prover serviços de segurança e não simplesmente um protocolo

O IPSec é altamente flexível e configurável, permitindo que o administrador selecione para a parametrização desejável opções como algoritmos de criptografia e protocolos de segurança especializados. São configuráveis também opções de serviços de segurança, tais como, controle de acessos, integridade, autenticação, confidencialidade, etc. Além disso, pode-se controlar também a amplitude de aplicação dos serviços do IPSec, baseando-se no fluxo o qual ele irá proteger, podendo ser tanto em ambiente reduzido, como em uma conexão entre dois *hosts,* ou mesmo em um ambiente macro, controlando, por exemplo, um fluxo completo de pacotes que trafegam entre os *gateways*.

A plataforma *IP Security* através da utilização dos protocolos AH *(Authentication Header*) e ESP (*Encapsulating Security Payload)* e do protocolos de gerenciamento de chaves ISAKMP (*Internet Security Association and Key Management Protocol*), foi desenvolvido para efetuar melhorias nas ferramentas de segurança. A união entre essas duas chaves gera uma Associação de Segurança(SA), que ao ser criada, recebe um índice de parâmetros de segurança(SPI) pela máquina receptora.

A função do ISAKMP é definir formatos de pacotes para troca de dados de criação e autenticação de chaves. A forma exata das chaves e dos dados de autenticação depende da técnica de criação da chave, do algoritmo de criptografia e do mecanismo de autenticação utilizado.

3.3.1 – Authentication Header

O *Autentication Header*, ou cabeçalho de autenticação, visa prover uma proteção contra reproduções, pois, através de seu uso, pode-se garantir a integridade sem conexões e autenticação da origem de dados para datagramas IP.

Ele é composto por um campo chamado *numseq*, o qual contém um contador que é incrementado automaticamente e é usado para proteção contra *replays*. Além disso, possui um campo chamado *DadosAutenticacao*, campo de tamanho variável que contém o código de integridade da mensagem para esse pacote. O campo precisa ser múltiplo inteiro de 32 bits. Abaixo, pode-se visualizar a figura 4, que ilustra o *Authentication Header* do IPSec.

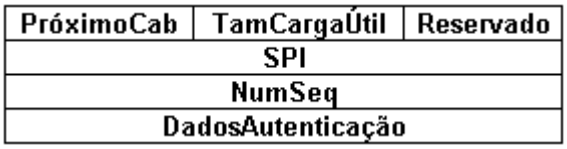

Figura 4 – AH do IPSec36]

Onde: PróximoCab: Tipo da próxima carga útil após o AH;

TamCargaÚtil: Tamanho do AH em palavras de 32bits menos 2^5;

Reservado: Reservado para uso futuro (vem com zero)

SPI: Índice de parâmetros de segurança

3.3.2 – Encapsulating Security Payload

O *Encapsulating Security Payload*, ou carga útil de segurança do encapsulamento, atua no *Ipv4* e no *Ipv6*, projetando uma mistura de serviços de segurança, combinando os cabeçalhos AH e ESP. O cabeçalho ESP oferece

confidencialidade, autenticação da origem de dados e integridade sem conexões e um serviço *anti-replay.*

O ESP, a exemplo do AH, é composto pelo campo *numseq*, que tem a mesma funcionalidade do AH, a proteção *anti-replay*, e o campo *DadosAutenticacao* contendo o código de integridade da mensagem36]. Abaixo, pode-se visualizar a figura 5, que ilustra o cabeçalho ESP do IPSec.

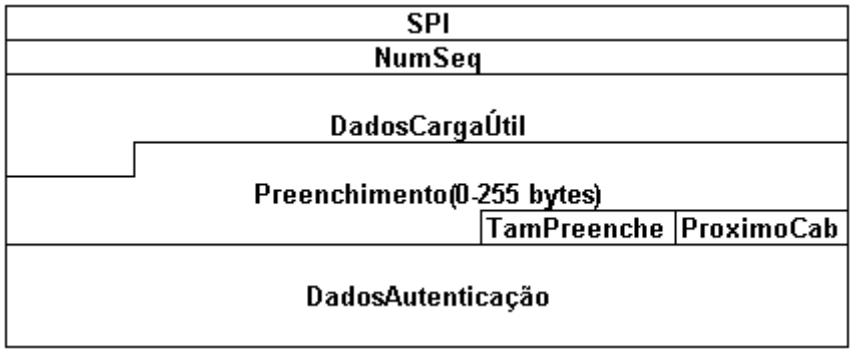

Figura 5–ESP do IPSec36]

Onde: SPI: Índice de parâmetros de segurança;

DadosCargaUtil: Contém dados descritos pelo campo PróximoCab;

Preenchimento: Valida limitação quanto ao numero de caracteres

TamPreenche: Registra quanto preenchimento foi acrescentado aos dados.

Uma dúvida que pode surgir seria: “O ESP e o AH tem praticamente a mesma funcionalidade?”. A resposta seria sim. Eles têm as mesmas funcionalidades, porém aplicadas de formas diferentes, o que justifica até mesmo sua utilização em conjunto,

técnica a qual inclusive é amplamente utilizada para aumento da segurança nos itens citados acima.

Atualmente pode-se encontrar filtros de pacotes que utilizam o Ipv6, tecnologia baseada no IPSec, como o ip6tables, que pode ser dado como um possível trabalho futuro à esse.

4 – DISPOSITIVOS DE SEGURANÇA

4.1 – Firewall

Um *firewall* é um componente ou um conjunto de componentes que tem como função analisar pacotes que trafegam entre redes, sendo tanto de uma rede interna para a Internet, quanto entre redes internas.

Ao contrário do que muitos pensam, um *firewall* não é um dispositivo único, mas sim um conjunto de ferramentas de segurança instaladas e configuradas de modo a trabalharem em conjunto, garantindo assim a aplicação dos parâmetros implementados pelo administrador de segurança para tratamento dos pacotes que trafegam pelas redes.

Entre as principais arquiteturas para uma implementação de um *firewall*, tem-se a *DUAL-HOMED HOST*, *SCREENED HOST* e *SCREENED SUBNET*.

A arquitetura implementada foi a *DUAL-HOMED HOST*, Figura 6, que é a arquitetura mais conhecida e mais utilizada. Consiste de uma máquina que intercala e divide duas redes de forma centralizada. Essa máquina é composta por duas placas de redes e age como roteador, interligando as redes.

Abaixo Ilustração da Topologia DUAL-HOMED HOST.

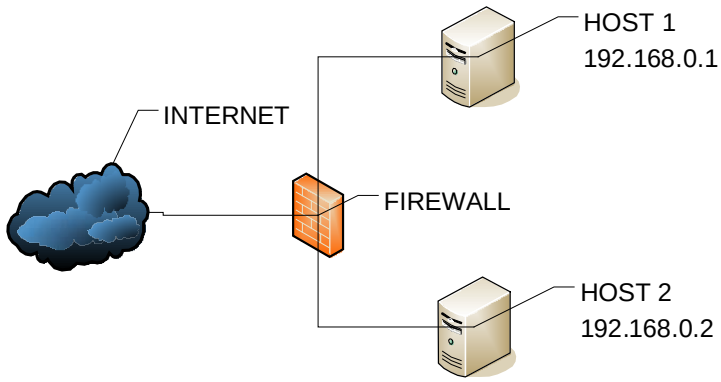

Figura 6 - Topologia de Firewall.

4.2 – BASTION HOST

Tem seu nome originado das arquiteturas fortemente construídas no período Medieval. O *Bastion Host* é encontrado em todas as topologias estudadas nesse projeto e pode ser definido como o *Gateway* da rede, tornando-se o único *host* da rede interna por onde os *hosts* da rede externa podem abrir conexões com os servidores da rede interna, como *email* e *http*.

Por ser o servidor que está localizado entre as redes interna e externa, o *Bastion Host* torna-se alvo preferencial para ataques e tentativas de *login* não permitidos.

Para aumento da segurança algumas topologias fazem uso te uma técnica denominada DMZ, *De-Militarized zone*, ou zona desmilitarizada, nome dado à região que separa as Coréias do Norte e do Sul. Se algum invasor conseguir burlar a segurança do *Bastion Host*, terá ainda que passar pela DMZ, ou rede periférica, não conseguindo acesso direto a rede interna.

4.3 – IDS - Instrusion Detection System

O IDS ou Sistema de Detecção de Intrusão tem como função a descoberta e análise de possíveis ataques que possam ser direcionados a rede por ele vigiada. Ele é geralmente um sensor ou um conjunto deles, que ao analisar qualquer vulnerabilidade informa sobre as atitudes consideradas suspeitas ao *firewall*, que poderá bloquear a ação.

Existem muitos programas de IDS e eles agem em conjunto com o sistema de gerenciamento da rede, complementando a segurança da mesma. Um exemplo de Sistema detector de Intrusão é o *Snort*, implementado neste trabalho e mais especificamente definido no capitulo 6, referente à implementação prática.

4.4 – Honeypots

Honeypots ou potes de mel são ambientes criados para a analisar invasões. Trata-se de uma estrutura especificamente montada para ser invadida, e no ato desta invasão analisar e estudar as ações do atacante, podendo até mesmo contra atacá-lo. Em resumo um *Honeypot* é uma estrutura configurada com falhas de segurança banais, as quais facilitam a ação de invasores que, ao pensar que estão invadindo um servidor, na verdade estão sendo estudados. Quanto ao contra ataque trata-se de uma ação não recomendável, pois o atacante pode estar utilizando de um IP *spoofing* que pode ser desde um IP interno, de algum site conhecido, ou de qualquer outra máquina na *Web*.

4.4.1 – Tipos de Honeypots

Existem dois tipos de *Honeypots*, os de baixa interatividade e os de alta interatividade, também conhecidos como *Honeynets*.

Os *Honeypots* de baixa interatividade usam *softwares* para simular um ambiente a ser invadido. O ambiente real onde este é instalado deve ser configurado para ser seguro a fim de que o invasor não ultrapasse o perímetro que foi definido para invasões. Este tipo de *Honeypot* não tem muita capacidade para adquirir informações do invasor.

Os *Honeypots* de alta interatividade ou *Honeynets* são capazes de adquirir muito mais informações dos invasores, podem ser divididos entre dois tipos, os *Honeynets* reais e os *Honeynets* Virtuais.

Honeynets reais são mecanismos de contenção maiores, dispostos como uma rede normal, com vários micros, sistemas operacionais, *hubs* e *switches*, quando necessários. Esses *Honeypots* têm aplicativos de contenção como *firewalls.* Nesses aplicativos existem meios de monitorar as ações dos invasores. Os *Honeynets* reais tem

de ser muito bem configurados para que não possam ser usados como pontes para invasão de outros computadores.

Honeynets Virtuais são *honeynets* simulados por *software*, estes necessitam de máquinas mais parrudas para execução, pois simularão vários sistemas operacionais numa mesma máquina. Exigem um ambiente real para instalação muito bem configurado e sua capacidade de adquirir informações do invasor depende da capacidade do emulador.

5 – TÉCNICAS UTILIZADAS PARA INVADIR SISTEMAS

Técnicas utilizadas para invadir sistemas computacionais não autorizados, os invasores testam as possibilidades de falhas de segurança (vulnerabilidades) nos sistemas, para isso utilizam algumas técnicas, como por exemplo, tentativas de invasão para conhecer os meios de segurança dos sistemas a serem invadidos e nisto suas possíveis falhas para uma invasão efetiva.

De acordo com o tipo de invasão, classificam-se os invasores, na maioria das vezes como Hackers, Crackers, Carders ou Phreacking. Pessoas que realizam este tipo de atividade detém na maioria das vezes um amplo conhecimento nas áreas computacionais, principalmente as voltadas à segurança de redes e de sistemas operacionais.

5.1 – Verificando a Integridade do Sistema

Em algumas ocasiões uma atitude comumente utilizada por administradores de redes é verificação da integridade, que consiste basicamente na montagem de um ambiente considerado seguro, servindo de comparação caso o servidor tenha sido invadido e exista a necessidade da verificação exata de todos os danos causados pela invasão.

Em resumo, o administrador grava uma base com informações sobre a estrutura de diretórios, inodes e permissões de grupos e de usuários para comparações futuras.

No capitulo 6, há mais informações sobre a ferramenta AIDE, que é responsável pela criação da base que será utilizada futuramente pelo administrador.

5.2 – Conhecendo sobre Hackers, Crackers, Carders e Phreackings

Hackers, atribuição mais conhecida quando se fala de invasor de sistemas. Estes têm geralmente intenções menos destrutivas, eles furtam arquivos e informações de seu interesse, mas porém não destroem os arquivos e informações dos sistemas invadidos. Atualmente muitos têm sido convocados por empresas para trabalhar na área de segurança destas para que testem seus sistemas e possibilitem que estes se tornem mais seguros.

Crackers, são basicamente Hackers com instinto destrutivo. Estes roubam informações, dinheiro, e tudo que conseguirem dos sistemas invadidos, e não se contentando somente com isso geralmente ainda causam danos ao ambiente invadido ou a informação original, estes também gostam de se identificar através de nicks (pseudônimos), a fim de serem conhecidos na mídia.

Carders, estes furtam números e senhas de cartões de crédito alheios além de gerarem números para fazer compras via Internet indevidamente. Ou seja, os valores de suas compras acabam indo para indivíduos normais e causam grandes prejuízos as operadoras de cartão de crédito.

Phreacking, pessoas com vasto conhecimento na área de telecomunicações a qual é paralela à área computacional. Estes indivíduos realizam atividades tais como grampos telefônicos, clonagem de linhas celulares, ligações clandestinas, mudanças de programação do sistema de telefonia, etc; tudo isso buscando benefício próprio.

5.3 – Ferramentas e métodos de invasão conhecidos

5.3.1 – Spoofing

Técnica que consiste em enganar o sistema de segurança passando-se por uma identidade falsa(IP). Geralmente trata-se de informar ao sistema de segurança de autenticação um IP conhecido, que dificultará ou inviabilizará o rastreamento de quem está fazendo o ataque, ou um IP interno da rede atacada, o qual possibilitará fácil acesso a pastas e arquivos confidenciais.

5.3.2 – Port Scans

Realiza uma varredura em todas as portas do IP desejado, buscando estabelecer conexão entre estas. Caso seja encontrado algum serviço em execução nestas portas, o mesmo será retornado.

5.3.3 – Sniffers

São ferramentas que rodam em Background e em modo promiscuo, ou seja, analisam todos os pacotes que estão trafegando pela rede no momento. Existem, no entanto, sniffers com atitudes maliciosas, cujo objetivo é analisar pacotes em busca de senhas ou informações confidenciais e sniffers sem atitudes maliciosas, que analisam os pacotes em busca de algum tipo de ação que venha ser prejudicial à rede, restringindo-as. Vale ressaltar que o sniffer, para atingir seu objetivo precisa estar instalado em algum ponto da rede interna por onde trafegue os dados por ele desejado.

5.3.4 – DoS (Denial of Service)

Ataque que tem como objetivo a indisponibilização de um serviço, ou seja, tira-lo do ar. Esse tipo de ataque baseia-se na simulação de muitas conexões ao mesmo serviço, de forma a ultrapassar a capacidade de acessos por ele permitido. Devido ao congestionamento criado, o objetivo do DOS é atingido e o serviço fica indisponível. A principal característica DOS é o ataque provindo de um único ponto.

Como danos causados pelo DOS, apresenta-se não necessariamente perdas de dados, mas a indisponibilização de serviços importantes para as empresas.

Um exemplo desse tipo de ataque é o SYN Flood, que, conforme seu próprio nome sugere, causa uma inundação de SYNs. Como esse abre muitas conexões e não as fecha, o serviço acaba sendo indisponibilizado.

5.3.5 – DdoS – Distributed Denial of Service

Ataques que agem da mesma forma que o DOS, ou seja, visam indisponibilizar serviços porém, ao contrario do DOS, em que o ataque parte de um único ponto, no DDOS o ataque é coordenado e simultâneo a partir de diversos pontos, dificultando qualquer ação de defesa, exatamente por ser distribuído.

Uma ferramenta conhecida que utiliza as técnicas do DDOS para coordenar os ataques é o TRIN00. Alem de utilizar conexão UDP para comunicação entre os atacantes, estabelece uma conexão TCP com as vitimas, além de ser ocultado por um trojan, o qual o torna transparente. DDOS é um ataque criado por um programador brasileiro.

5.4 – Vírus

5.4.1 – Introdução

Atualmente, com o crescimento constante do tráfego de informações via internet e a grande utilização de ferramentas de email, encontra-se cada vez mais exposto ao risco sempre constante dos vírus, que tem por objetivo impedir o bom funcionamento de nossos sistemas, causando percas financeiras, percas de dados, além do aumento de vulnerabilidades a ataques externos.

5.4.2 – Vírus

Vírus nada mais são do que programas especificamente desenvolvidos para trazer algum mal ao sistema, tendo como propriedade a fácil proliferação, ou seja, facilidade em se alastrar para outros computadores ou redes, causando uma infecção em massa. As características apresentadas pelos programas de vírus são as mais diversas, podendo gerar desde causas de lentidão no processamento, até a destruição completa do seu sistema de arquivos.

5.4.3 – O surgimento dos Vírus

Os vírus praticamente surgiram junto com os computadores pessoais, os primeiros infectavam disquetes de computadores Apple, posteriormente com o surgimento dos PC da IBM e com a propagação das redes de computadores os vírus começaram a ser melhorados e causar mais estragos aos computadores. Sempre considerados como uma perturbação à utilização dos computadores, eles foram passando desse estágio e ganhando um ar maior de praga digital. Nos anos 90, com o surgimento da internet e com o seu crescimento os vírus ganharam capacidade de se

replicar via email e outros tipos de comunicação via internet. Hoje a cada dia estão mais avançados tecnologicamente e causando mais estragos.

5.4.4 – Tipos de Vírus

5.4.4.1 – Worm/Vermes

O Worm é uma espécie de vírus que tem como propriedade o fato de se copiar automaticamente para outros computadores, proliferando-se de maneira bastante rápida dentro de redes internas e na Web. Geralmente os Worms se utilizam de emails para proliferação, podendo se auto-enviar para toda a sua lista de emails com facilidade. Quando estes se copiam, a cópia passa a operar realizando a mesma função. Assim esse tipo de vírus causa muito transtorno, lentidão e prejuízos na utilização das redes de computadores.

5.4.4.2 – Malwares

Malware nada mais é que uma denominação para as inúmeras classificações de vírus, tais como cavalos de tróia (trojans), vermes, etc. O termo Malware vem da junção do termo em inglês “malicious software”.

5.4.4.3 – Spywares

Spywares são programas mal intencionados que têm por característica espionar as atividades do computador onde se alojou. Estes programas entram no computador de uma forma difícil de se perceber, pois geralmente são baixados da internet em transmições P2P, por via sites de jogos, além de muitos softwares não confiáveis também instalarem algum tipo de programa spyware no computador. Os Spywares

podem copiar desde informações banais até informações confidenciais do computador, ou seja, representam perigo para o computador.

5.4.4.4 – Backdoors

Os Backdoors são programas que têm por característica facilitar o acesso externo ao computador no qual se instala. Ele realiza esta tarefa abrindo portas de acesso neste computador e se comportando como um server de acesso, possibilitando que outros se conectem diretamente utilizando as portas conhecidas pelo programa.

Os Crackers utilizam-se de backdoors como ferramenta facilitadora. Após invadir um computador, eles costumam deixar algum backdoor para lhe garantir maior facilidade em acessos futuros.

Dois dos backdoors mais conhecidos são o Back Oriffice e o NetBus. O Back Oriffice é um backdoor que trabalha em todas as redes que se utilizam do protocolo TCP/IP e possibilita que, conforme o tópico anterior, o computador seja acessado através de portas conhecidas, que no caso do Back Oriffice são as portas 31337 e 31338.

O Net Bus é também um backdoor, a exemplo do Back Oriffice, porém mais velho e de maior tamanho. Utiliza os conceitos citados acima e, assim como o Back Oriffice, utiliza-se do protocolo TCP/IP para aguardar por conexões externas.[24] Os backdoors são considerados Trojans, classificação a qual vem a seguir.

5.4.4.5 – Trojans (Cavalos de Tróia)

Trojans ou Cavalos de Tróia são arquivos com aparência inofensiva, o qual se fazem passar por ferramentas interessantes ou úteis ao dia a dia, porém que carregam consigo códigos maliciosos, que podem desde transmitir informações confidenciais até mesmo disponibilizar serviços de backdoors, como visto no tópico anterior. Por

característica os trojans não se auto replicavam, porém com a evolução dos vírus essa característica mudou, existindo assim hoje em dia trojans com capacidade de auto replicação.

6 – IMPLEMENTAÇÃO DE FERRAMENTAS DE SEGURANÇAS EM REDES CORPORATIVAS

6.1 – Motivação

Aumentar o conhecimento na Área de Segurança de Redes e de conhecer novos ambientes, novas ferramentas e métodos amplamente utilizados pelas empresas foi um fator importante também, visto que havia apenas uma visão superficial sobre a área de redes.

6.2 – Cenário

6.2.1 – Lousano Fios e Cabos Elétricos

A Lousano Fios e Cabos Elétricos era uma empresa de médio porte na área de fios e condutores elétricos, localizada em Itaquera, onde foram implantados os conceitos apresentados neste trabalho de conclusão. Ao analisar o cenário, verificou-se a seguinte situação:

- A empresa dispunha de um *firewall*, porém o mesmo era desatualizado, o que não garantia integridade dos dados nem da rede interna;

- Não havia confiabilidade nos *logs* devido a outros problemas, que seriam a ausência de configuração eficaz das ferramentas de segurança, além da não atualização do sistema operacional do servidor, tornando-o obsoleto;

- Surgiu a necessidade de divisão do corpo da Lousano em 3 pontos: um Data Center, localizado em Alphaville, a matriz, localizada em Itaquera e uma filial, localizada em Santana. Isso causou uma mudança na topologia da rede da Lousano, que precisou sofrer adaptações, as quais serão apresentadas em nosso trabalho.

6.2.1.1 – Proposta

Reinstalação e configuração do sistema operacional do *firewall*, além da implementação de ferramentas de segurança, visando obter a integridade e disponibilidade desejada aos dados.

Para resolução do problema da mudança de topologia, foi implementado o algoritmo OSPF na interligação entre as redes, garantindo no máximo possível, o tempo de disponibilidade dos serviços e o fornecimento de balanceamento de carga, que permite ao administrador especificar múltiplas rotas para um mesmo destino. O OSPF distribui o trafego igualmente por todas as rotas. Durante a implementação na Lousano foram encontradas algumas dificuldades relacionadas com a complexidade deste algoritmo, que foram sanadas através de pesquisas e consultores contratados.

Em nossos estudos foram encontradas informações sobre vulnerabilidades deste protocolo, que ocorre no processamento de pacotes OSPF, um protocolo de roteamento intra-AS. Trata-se de uma vulnerabilidade a ser explorada remotamente, na qual um pacote mal-formado enviado para um dispositivo vulnerável, resulta num *reload* e numa condição de DoS consequentemente. Porém há uma ressalva a ser feita sobre essa vulnerabilidade, que é a necessidade do atacante conhecer algumas configurações do sistema atacado, para obter o resultado esperado.

6.2.1.2 – Implementação – Lousano

A empresa possuia 4 servidores em um *Datacenter* localizado em Alphaville, que são acessados pela matriz em Itaquera e por uma filial em Santana. Entre os 4 servidores está o firewall implementado por nós, o qual disponibiliza acesso a internet para a matriz e a filial. Tanto o *Datacenter* quanto a matriz e sua filial estão ligados por *Frame-Relay*, utilizando o protocolo de roteamento OSPF. Abaixo a figura 7, ilustrando a topologia utilizada pela Lousano:

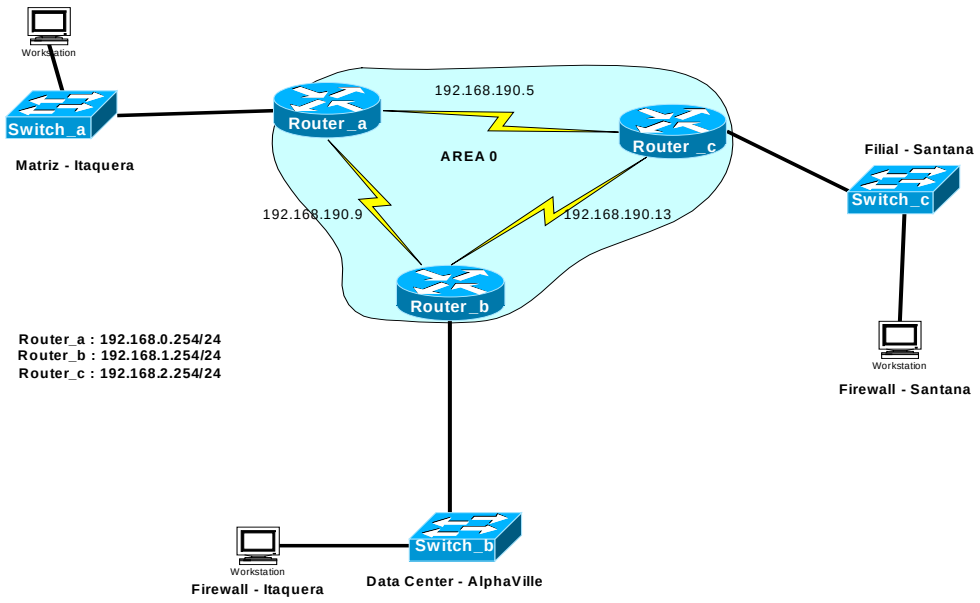

Figura 7 - Topologia Lousano

6.3 – Ferramentas Implementadas

Para a instalação das ferramentas citadas neste projeto, utilizou-se comandos presentes na maioria das distribuições *Linux*, A explicação conceitual e métodos de implementação vêm a seguir:

- Iptables 8
- Squid-2.5.stable7
- Snort-2.4.3
- Swatch-3.1.1
- Syslog
- Aide-1.11
- Bind 9
- Sarg-2.0.9

6.3.1 – IPTABLES

Site official http://netfilter.samba.org

Iptables – é um filtro de pacotes que decide que tipos de datagramas podem ser processados normalmente e quais serão descartados, e é de fundamental importância na implementação de um *firewall* baseado em filtro de pacotes, que são os mais utilizados e mais simples para implementação. Para configuração, foi utilizada uma tabela de endereços que contém informações sobre a origem e o destino dos pacotes e as portas que serão utilizadas, onde através do *prompt* de comando, pode-se inserir novas linhas ou remove-las.

Para que os dados nesta tabela não sejam perdidos quando o servidor for desligado, foi criado um *script* com regras básicas de roteamento que é executado do arquivo rc.local no diretório /etc/rc.d/.

O filtro de pacotes *IPTABLES* é parte integrante das distribuições *Linux* a partir do *Kernel-2.4*, e através de critérios podem-se determinar quais datagramas serão filtrados, como:

- Tipo de protocolo: TCP, UDP e ICMP;
- Tipo de datagramas: SYN/ACK, data, ICMP *Echo Request*;
- Endereço de destino e de origem do datagrama;

Neste projeto, foi criada uma cadeia de regras com o objetivo de proteger nossa rede interna de ataques como:

- Trinoo
- Trojans
- Spoofings
- Worms
- Syn-flood
- NetBus
- Back Orifice
- Multicast
- Port Scanners

No capitulo cinco “Técnicas utilizadas para invadir sistema”, pode-se encontrar mais informações sobre os tipos de ataques citados acima.

6.3.2 – SQUID

Site official http://www.squid-cache.org/

Squid – É uma ferramenta utilizada para instalação e configuração de um *firewall* baseado em *proxy cache*, onde as solicitações à páginas efetuadas pelos clientes são pesquisadas, caso a solicitação não seja encontrada no cache a mesma é enviada ao servidor, que, por sua vez, tem a função de resolver a pesquisa e armazenar a página no *cache* para futuros acessos.

Um *proxy* também pode ser utilizado para controlar o acesso a porta 80 baseando-se em vários tipos de regras, como será demonstrado mais adiante. O *proxy* tem como seu principal objetivo gerenciar o acesso à redes públicas, como a internet, a partir de estações da rede interna. Em resumo, uma estação faz uma requisição de alguma página da internet, por exemplo, e o *proxy* é o responsável por buscar essa página, carrega-la no *cache* e provê-la para a estação da rede interna. Ou seja, o *proxy* liga as máquinas da rede interna a redes externas sem acesso direto.

6.3.2.1 – CONTROLE DE ACESSO

No controle de acesso do *Squid* há recursos suficientes pelos quais pode-se definir quais tipos de serviços podem ser acessados, por quais máquinas ou por quais usuários e em horários estipulados pelo administrador. Foram configuradas duas regras para acesso:

ACESS_FULL: Os usuários têm acesso integral, com restrições a alguns sites.

RESTRITO: Os usuários têm acesso somente a sites estipulados pelo administrador.

Foi utilizado o *htpasswd* que é uma aplicação do *Apache* para criação de senhas dos usuários no arquivo /etc/squid/squid_passwd. Deste modo, somente quem possui *login* e senha terá permissão de acesso à internet e em horário estipulado.

Foi configurado também o bloqueio de extensões como *mpeg, avi, exe, mp3, wma*. Controle do uso da banda através das diretivas *delay_spool*.

Para instalação do s*quid*:

. configure –prefix=/etc/squid

Make all

Make install

Para que o s*quid* seja ativado automaticamente durante a inicialização do servidor, deve-se incluir a linha abaixo no arquivo rc.local:

/etc/squid/sbin/squid –d1

Os arquivos de configuração do s*quid* se encontram no caminho abaixo:

/etc/squid/squid.conf

/etc/squid/restrito

/etc/squid/access_full

6.3.3 – SNORT

Site oficial http://www.snort.org

Ferramenta de alerta em tempo real que reconhece as tentativas de ataque como também as formas utilizadas para esses ataques através de regras.

O *Snort* trabalha de forma análoga a um *Sniffer*, como o *TCPDump*, sendo o que o diferencia é sua forma de análise dos dados encontrados, sendo muito eficaz na detecção de diversas vulnerabilidades.

As atualizações de regras estão disponíveis no site oficial do *Snort*, onde podem ser encontradas regras para proteção contra port scanner, DoS, DDoS e outros tipos de ataques.

Neste projeto o *snort* está instalado e configurado no *firewall* e monitora as *interfaces* de rede *eth0* e *eth1*, por se tratar da máquina que está entre a rede interna e a internet.

Para instalação do *snort*:

. configure –prefix=/etc/snort

Make all

Make install

Para que o *snort* seja iniciado e comece a monitorar automaticamente as *interfaces* de rede, toda vez que o servidor for reiniciado, deve-se incluir as linhas abaixo no arquivo rc.local:

/usr/local/bin/snort –c /etc/snort/snort.conf -d – o –s –A full eth0

/usr/local/bin/snort –c /etc/snort/snort.conf -d – o –s –A full eth1

Os arquivos de configuração do *snort* encontram-se disponíveis nos caminhos abaixo:

/etc/snort/snort.conf

/etc/snort/rules

6.3.3.1 Topologia do Snort

Na figura 8 ilustra-se a Topologia do Snort.

Figura 8 - Topologia Snort

6.3.4 – SYSLOG

O *Kernel* do sistema, bem como a maioria dos programas e aplicativos para *Linux*, gera mensagens de inicialização, status ou erro de operação, e é importante que os administradores possam ter acesso a estas mensagens, sempre que necessário, para que se possa verificar o que ocorre no sistema.

No sistema operacional, o *syslog* recebe mensagens do sistema para arquivos apropriados. Neste projeto, por *default*, as mensagens são escritas no arquivo */var/log/syslog.* O arquivo de configuração do syslog encontra-se no caminho: /etc/syslog/syslog.conf

6.3.5 – SWATCH

Site para download http://sourceforge.net/projects/swatch/files/swatch/3.2.3/

Um dos grandes problemas na administração de redes é que a informação fica localizada em registros de *logs*, gerado por ferramentas como o *syslog*, e nem sempre se tem tempo para monitorá-los.

O *swatch* é uma aplicação em *perl* que utiliza os módulos *perl*::*Date* e *perl*::*Parse*, que são encontrados no site www.cpan.org e devem ser instalados para garantir o seu perfeito funcionamento. O seu arquivo de configuração encontra-se no diretório de *root* com o nome de *swatchrc*.

Para a instalação do *swatch*, como pré-requisito faz-se necessário constar no sistema as aplicações *perl::parse* e *perl::date*. Depois de atendidos esses pré-requisitos, deve-se digitar o comando abaixo no diretório de instalação do *swatch*:

```
Perl Makefile.pl
```

O *swatch* é uma ferramenta utilizada para analisar registros de *logs* e permite o envio de alertas em tempo real. Em seu arquivo de configuração foi incluída a linha "mailaddress=suporte@lousano.com.br", que envia mensagens para o email estipulado, contendo informações sobre tentativas de acessos, listando *ips* e portas acessadas.

No presente projeto o *swatch* está analisando os arquivos de registro continuamente, procurando por tentativas de *login* e falhas no sistema.

6.3.6 – AIDE - Advanced Intrusion Detection Environment

Site oficial http://www.cs.tut.fi/~rammer/aide.html

O AIDE é uma ferramenta utilizada para analisar as mudanças decorrentes no sistema operacional, mudanças tais como:

- Arquivos recém criados;
- Inodes alterados;
- Mudanças nos parâmetros dos programas;
- O *hash* guardado é comparado com o gerado pela cópia atual de cada arquivo.

É um programa criado para ser uma versão gratuita do *Tripwire.* Seus arquivos de configuração encontram-se no caminho:

/etc/aide/aide.conf

/etc/aide/aide.db

Para garantir a integridade utilizando o AIDE, o administrador deve criar uma base de dados de um sistema seguro, do qual se há certeza da integridade. Esta base criada deve ser mantida em local seguro e comparado com as próximas bases geradas. No arquivo de configuração do AIDE encontrado no /etc/aide.conf, define-se os diretórios que serão verificados e de que maneira será feita esta verificação, lembrando que nos diretórios, os quais são mantidos os *logs* do sistema, não é feita a verificação, pois seus arquivos mudam frequentemente.

Para criar uma base nova no diretório /etc/aide/ade.bb.new basta utilizar o comando:

Aide –i

Periodicamente, deve-se comparar as bases para que se saiba com precisão as alterações que o sistema operacional sofreu. Para isso basta executar o comando:

Aide –C

No projeto de implementação, gerou-se um hash da estrutura de diretórios, conforme demonstração na figura 9, logo abaixo, na qual mostra uma comparação feita com o AIDE.

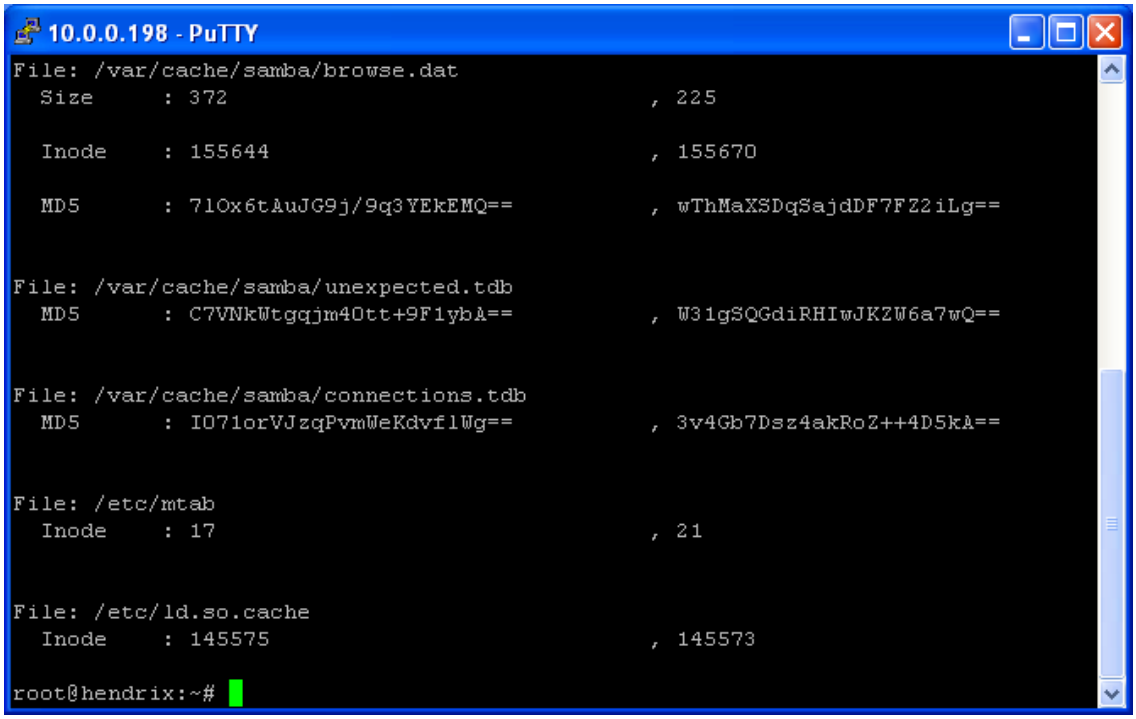

Figura 9 - Verificação feita com o AIDE

6.3.7 – BIND

Site oficial http://www.bind.com

- Fluxograma para proteção do DNS

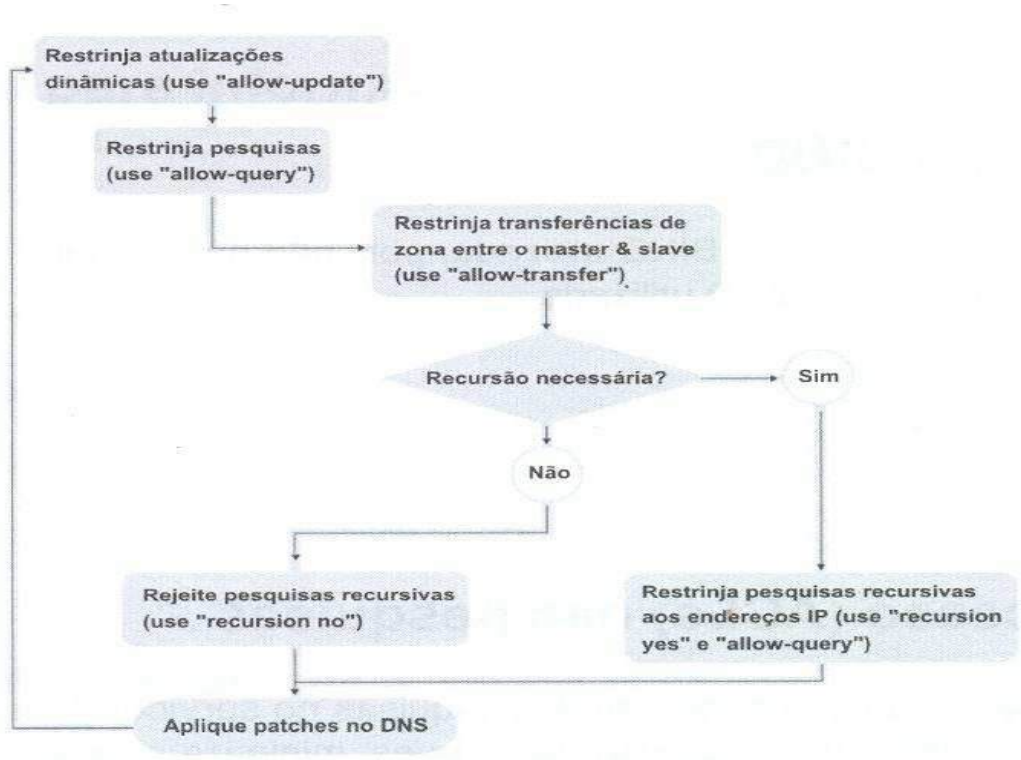

Figura 10 - Fluxograma para proteção do DNS

Na implementação efetuada na Lousano, havia um servidor de DNS primário, configurado em uma plataforma *Windows* 2000, que efetua atualizações dinâmicas no arquivo lousano.com.br.db, encontrado no diretório /var/named/caching-example. Dessa forma, o *firewall* torna-se um servidor de DNS secundário para qualquer eventualidade, assim evitando a indisponibilização do serviço.

6.3.7.1 – Restrição para transferência de zonas

Allow-transfer – Opção utilizada para restringir transferências de zonas. Por padrão, a transferência de zona é habilitada a quem quiser fazer uma transferência de zonas no servidor de nomes.

6.3.7.2 – Especificação de Interfaces a escutar

Listen-on – utilizado para especificar quais interfaces de rede deverão ser escutadas para respostas de requisições e pesquisas de zonas.

6.3.7.3 – Informação da versão da zona

Version – utilizada para mascarar a versão do *bind*, pois não há qualquer razão para que outros usuários, além do administrador da rede, saibam a versão do BIND utilizada.

6.3.7.4 – Uso de controle de acessos nas pesquisas

Allow-query – utilizado para restringir os *hosts* que podem fazer pesquisas no servidor de nomes, restringir *queries* e minimizar a exposição a vulnerabilidades de *cachê-poisoning*.

6.3.8 – SARG - Squid Analysis Report Generator

Site para download http://sarg.sourceforge.net/sarg.php

SARG – é uma ferramenta utilizada para gerar relatórios baseados nos *logs* criados pelo *squid*, no arquivo *"/var/log/squid/access.log"*. Através do SARG é possível visualizar de forma transparente e com interface gráfica as seguintes informações:

Informações disponibilizadas pelo SARG:

- Acessos a sites, por usuários;
- Tempo de permanência;
- Consumo em bytes;
- Quantidade de conexões;
- Sites mais acessados;
- Sites negados;
- Falha de autenticação.

Como pré-requisito para a instalação do *Sarg*, faz-se necessário a instalação e configuração do *Apache*, criando um arquivo sarg.conf no diretório /etc/apache com o conteúdo:

Alias /sarg /var/www/squid-reports/dia

Directory Index index.html

No arquivo http.conf deve-se incluir a linha abaixo:

Include sarg.conf

Abaixo seguem telas demonstrando a implementação do *SARG* no projeto. A primeira tem uma visão sintética, figura 11, a segunda com uma visão analítica, figura 12:

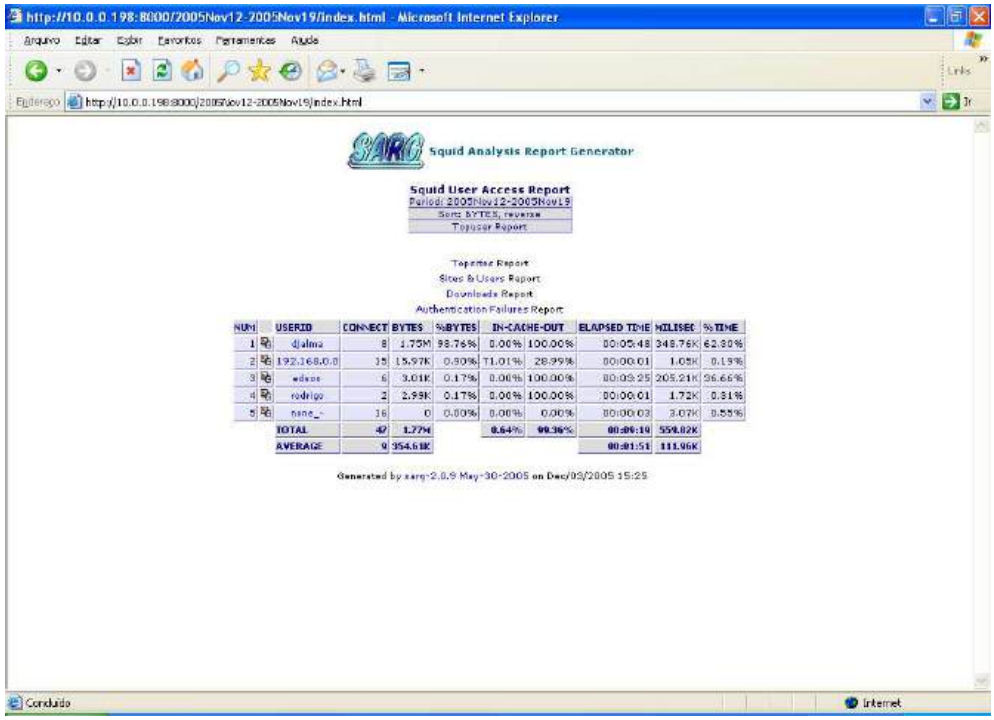

Figura 11 - Implementação do SARG

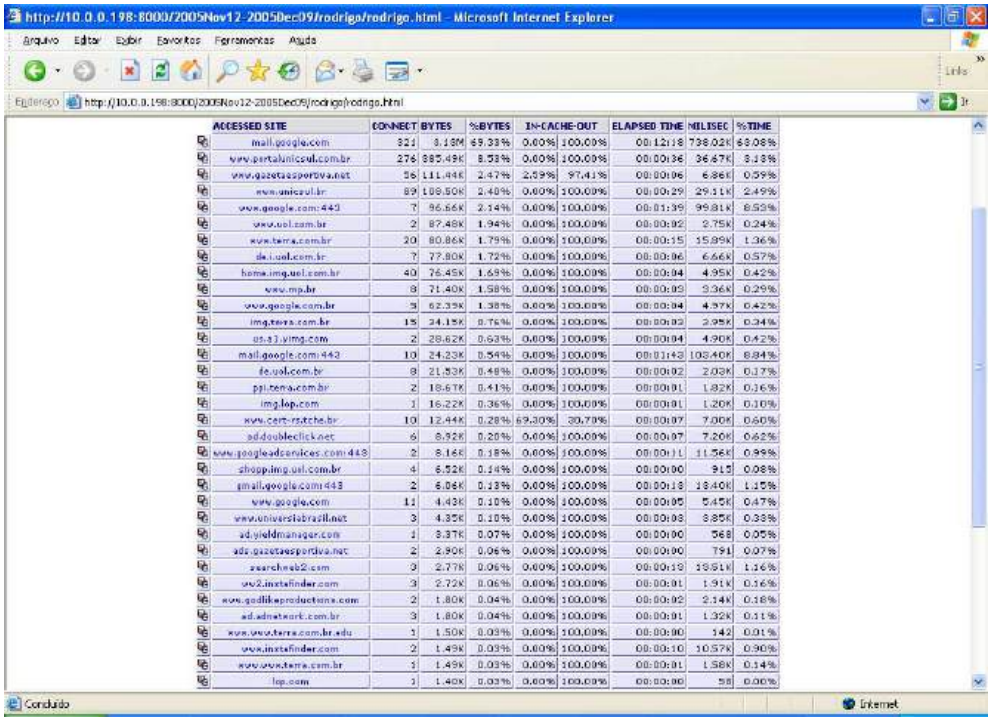

Figura 12 - Implementação do SARG (ANALÍTICA)

6.4 – Resultados Obtidos

Na implementação realizada na Lousano, foi obtido o sucesso esperado e o ambiente com configuração do OSPF.

Com as regras do *iptables*, pode-se bloquear alguns serviços, os restringindo somente a rede interna, como é o caso do FTP, e efetuando redirecionamento para os servidores *Web* e email, restringindo o acesso a porta 25 somente a usuários do domínio lousano.com.br.

Com o analisador de *logs Swatch*, há monitoramento diário e, através dos emails enviados, é possível saber o que ocorre com o sistema em tempo real. Além disso, pode-se ficar informado sobre alterações em arquivos de sistemas com o *Aide* e/ou impedir que sejam executados no servidor comandos de *port scanner*, impedindo dessa forma que se obtenham informações sobre os programas e serviços executados pelo *firewall*, protegendo-o contra diversas modalidades de ataques.

Com as ferramentas implementadas o índice de ataques por vírus ou *hackers* diminuiu bruscamente, pois foi aumentada a segurança contra invasores, a segurança na comunicação entre a rede interna e as redes externas foram fortificadas, além do aumento da rastreabilidade, no caso de mesmo assim existir algum eventual ataque.

6.5 – Trabalhos Futuros

Como trabalho futuro ao realizado, estudava-se na incrementação da Topologia da Lousano, com a implementação de um *firewall* para cada uma das redes, visando o melhor controle da Banda através do *Software* CBQ. Isso faz-se necessário devido ao numero de serviços concorrentes utilizados na empresa provindos do mesmo Data Center, como arquivos, Banco de Dados e Servidor de *emails.*

7 – CONCLUSÃO

Ao término desse trabalho, pode-se concluir como é importante manter um nível de risco reduzido, que garanta a integridade dos dados em ambientes corporativos e a importância de uma boa implementação dos métodos mais eficazes, para proteção estrutural de uma companhia, devido ao crescimento da tecnologia e ao avanço das técnicas para “ataques” aos sistemas de segurança.

Este trabalho teve como objetivo demonstrar de maneira concisa, uma implementação de segurança em redes, através de softwares livres, ou de domínio público, que demonstram sua eficácia em muitos ambientes corporativos e através do tempo vêm conquistando um espaço até então dominado por conhecidos softwares proprietários. Espera-se que o uso de ferramentas livres continue crescente e que um número maior de empresas se convença das vantagens que o uso do software gratuito proporciona, devido ao baixo custo e a alta qualidade.

A implementação deste projeto envolveu o estudo de programação de redes de computadores, configuração de servidores Firewall e Proxy e das estações de trabalho, técnicas de intrusão e intercomunicação de processos, entre outros. Isso possibilitou aos autores deste projeto, um grande aprendizado, em especial na área de Redes de Computadores.

8 – BIBLIOGRAFIA

- BRUGESS, Burgess Mark, **Princípios de Administração de Redes e Sistemas**, Segunda Edição, Rio de Janeiro, LTC Editora, 2006.
- STALLINGS, Stallings William, **Criptografia e segurança de redes**, Quarta Edição, São Paulo, Pearson Editora do Brasil, 2008.
- MORAES, Moraes Alexandre Fernandes de, **Segurança em Redes – Fundamentos**, Primeira Edição, São Paulo, Editora Érica 2010.
- NAKAMURA, Nakamura Emilio Tissato & Geus Paulo Lício de, **Segurança de Redes – em ambientes cooperativos**, Segunda Edição, São Paulo, Novatec Editora, 2010.
- TERADA, Terada Routo, **Segurança de dados – criptografia em redes de computador** - , Primeira Edição, São Paulo, Editora Edgard Blüchier Ltda, 2000.
- Revista Linux Magazine, números 62 ao 67(2010), IVC.
- BEZERRA, Dinarde Almeida Bezerra & Gustavo Magno de Sousa, **Protocolos Criptográficos,** Disponível em : http://www.projetoderedes.com.br, Acesso em: 15 nov. 2011.
- Cert.br, **Cartilha cert.br**, Disponível em: http://cartilha.cert.br, Acesso em: 15 nov. 2011.
- **Introdução à codificação DES** , Disponível em : http://pt.kioskea.net/contents/crypto/des.php3 , Acesso em: 15 nov. 2011.
- Equipe Conectiva - Treinamento, Apostila **Segurança de Redes: Firewall**, Primeira Edição, 2001, Conectiva S.A..
- Equipe Conectiva - Treinamento, Apostila **Fundamentos de Adm. de Sistemas**, Primeira Edição, 2001, Conectiva S.A..
- Equipe Conectiva - Treinamento, Apostila **Administração de redes Linux**, Primeira Edição, 2001, Conectiva S.A..