UNIVERSIDADE FEDERAL DE PELOTAS
INSTITUTO DE FÍSICA E MATEMÁTICA
DEPARTAMENTO DE MATEMÁTICA, ESTATÍSTICA E COMPUTAÇÃO
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

ESTUDO SOBRE AS PRINCIPAIS AMEAÇAS E TÉCNICAS PARA OBTENÇÃO DE FALHAS DE SEGURANÇA EM SISTEMAS COOPERATIVOS

Débora Lopes Ramos

Pelotas, 2005

Débora Lopes Ramos

ESTUDO SOBRE AS PRINCIPAIS AMEAÇAS E TÉCNICAS PARA OBTENÇÃO DE FALHAS DE SEGURANÇA EM SISTEMAS COOPERATIVOS

Proposta para Projeto de Conclusão apresentada ao Curso de Bacharelado em Ciência da Computação, Instituto de Física e Matemática, Universidade Federal de Pelotas.

Orientador : Prof. M.Sc. Carlos Jorge Ribeiro
Co-orientador(a): Profª. M.Sc. Flávia Azambuja

Pelotas, 2005

BANCA EXAMINADORA:

.....
Prof. Msc. Carlos Jorge Ribeiro

.....
Profa. Msc. Ana Marilza Pernas Fleischmann

.....
Prof. Msc. Leonardo Lemos Ribeiro

Dedico meu projeto a todas as pessoas que,
por algum motivo particular, ou por falta de oportunidades,
não puderam chegar até aqui.

LISTA DE ABREVIATURAS E SIGLAS

ASP	**Active Server Pages**
ACL	**Access List**
BD	**Banco de Dados**
CGI	**Common Gateway Interface**
DdoS	**Distributed Denial of Service**
DNS	**Domain Name System**
DoS	**Denial of Service**
FTP	**File Transfer Protocol**
ICMP	**Internet Control Message Protocol**
IDS	**Intrusion Detection System**
IP	**Internet Protocol**
ISS	**Internet Security Systems**
LDAP	**Lightweight Directory Access Protocol**
MAC	**Media Access Control**
NetBIOS	**Network Basic Input/Output System**
NIST	**National Institute of Standards and Technology**
NMAP	**Network Mapper**
SNMP	**Simple Network Management Protocol**
PBX	**Private Branch of eXchange**
SO	**Sistema Operacional**
TCP	**Transmission Control Protocol**
TCP/IP	**Transmission Control Protocol/ Internet Protocol**

UDP	**User Datagram Protocol**
VPN	**Virtual Private Network**

SUMÁRIO

LISTA DE FIGURAS

LISTA DE ANEXOS

RESUMO

A segurança da informação deve ser contínua e evolutiva. Sua importância pode ser reforçada quando novas oportunidades de negócios, no mundo digital, condicionam seu sucesso à eficiência da estratégia de segurança adotada pela empresa.

Considerando-se que as novas tecnologias trazem consigo novas vulnerabilidades e novas formas de ataque, que o número de crimes digitais cresce a cada ano e que a defesa é mais complexa que o ataque, entender a natureza destes é fundamental para preveni-los.

Portanto, esta monografia objetiva salientar a importância da segurança das informações, destacar as principais ameaças aos ambientes cooperativos e as estratégias empregadas pelos *hackers* e *crackers* para obtenção de informação confidencial das empresas, mas, principalmente, relatar as principais técnicas existentes para levantamento de vulnerabilidades de segurança.

Palavras-chave: segurança da informação, teste de vulnerabilidades, teste de invasão.

ABSTRACT

The safety of information must be continuous and evolutive. Its importance must be strengthened when new oportunities of business, in digital world, condition its sucess to the efficiency of the safety strategy adapted by the company.

Considering that the new technologies bring new vulnerabilities and new ways of attacking, that the number of digital crimes increases every year and that defense is more complex than the attack, to understand the nature of these is essential to prevent them.

Consequently, this monography aims to highlight the importance of security of information, to detach the main threats posed to the cooperative environments and the main techniques applied by hackers and crackers to obtain confidential information of companies, but, mainly, to report the most important techniques existing for checking vulnerabilities of security.

Key Words: security of information, vulnerability test, intrusion test.

1 INTRODUÇÃO

Um sistema cooperativo é um novo ambiente onde múltiplas organizações trocam informações por meio de uma rede integrada. Estas informações trafegam por esta rede interligando matrizes e filiais, parceiros comerciais e também clientes e fornecedores.

Com o surgimento da rede mundial de computadores e dos sistemas cooperativos, que tornaram as relações comerciais mais estreitas e também mais perigosas, a informação e os negócios tornaram-se suscetíveis a novas ameaças. Porém, a utilização destas novas tecnologias é primordial para as empresas buscarem vantagem competitiva. Neste cenário, resta às empresas assumir uma postura defensiva contra estas ameaças, pois seus dados precisam ser protegidos, de maneira que os três pontos-chave para um sistema ser considerado seguro sejam mantidos: **integridade, confidencialidade e disponibilidade da informação. (**NAKAMURA, 2003)

A **integridade** consiste em assegurar que os dados não possam ser modificados por pessoas não autorizadas. O conceito de dados neste caso é bem amplo, incluindo dados, programas, documentação, registros, fitas magnéticas entre outros. A **confidencialidade** é a segurança de que estes mesmos dados não possam ser acessados por usuários sem permissão e a **disponibilidade** é a garantia de que os serviços prestados por um sistema sejam acessíveis a processos e usuários autorizados. (DIAS, 2000)

Este trabalho tem como objetivo principal, além do estudo das principais ameaças aos sistemas cooperativos, o estudo das principais técnicas utilizadas por administradores de redes e *hackers* para levantamento de *bugs* de sistemas. Estas técnicas, quando usadas por administradores de rede, servem como recurso para a análise da rede; para o auditor, servem para validar ou não a política de segurança da empresa e para os *hackers*, servem como auxílio para o acesso não autorizado ao sistema, apesar de não terem sido desenvolvidas para este propósito.

Este documento está dividido em 6 capítulos:

- O capítulo 2, intitulado Tipos de Ataque, descreve os principais pontos explorados pelos *hackers* nas suas tentativas de invasão aos sistemas, os tipos

de ataques praticados pelos invasores para obtenção de informações e para causar negação de serviços, entre outros malefícios;

- No capítulo 3, intitulado Tipos de Testes de Segurança, encontram-se descritos os tipos de Teste de Segurança existentes para testar a segurança dos sistemas;
- O capítulo 4 contém uma explicação sobre o Teste de Invasão, que foi a técnica de teste de segurança utilizada no exemplo prático;
- O capítulo 5 apresenta o exemplo da aplicação de uma ferramenta de detecção de vulnerabilidades na Prefeitura Municipal do Rio Grande e os resultados obtidos;
- O capítulo 6 apresenta a conclusão e os trabalhos futuros;
- No anexo A encontra-se o relatório gerado pela aplicação da ferramenta *PortScan* nos IPs (Internet Protocol) analisados;
- No anexo B encontra-se o relatório gerado pela aplicação da ferramenta *NBScan* na mesma seqüência de IPs analisados anteriormente;
- O anexo C contém a autorização concedida pela Prefeitura Municipal do Rio Grande para a divulgação dos dados obtidos no teste.

2 TIPOS DE ATAQUE

As invasões aos sistemas podem ser executadas por meio da exploração de técnicas que podem ter como base a engenharia social ou invasões técnicas. Essas invasões exploram deficiências na concepção, implementação, configuração e/ou no gerenciamento dos serviços e sistemas, e continuarão existindo na medida em que o mercado é centrado nas características dos produtos e não na segurança.

Partindo da idéia de que um ataque *hacker* explora brechas, e estas brechas podem estar tanto no sistema operacional, passando por serviços e protocolos, rede e telecomunicações, basta que o intruso invada o sistema através de um destes níveis para colocar toda a segurança do sistema em jogo. Por esta razão é mais fácil um *hacker* penetrar no sistema do que um perito em segurança consertar suas falhas. (NAKAMURA, 2003)

Algumas condições exploradas nos ataques técnicos são descritas a seguir (NAKAMURA, 2003):

- Exploração de vulnerabilidades decorrentes de *bugs* na implementação ou no design do sistema operacional, serviços, aplicativos e protocolos. Vários protocolos possuem ferramentas de ataque específicas;
- Senhas fracas, que podem ser capturadas (*packet sniffing*), decifradas mesmo quando criptografadas (*cracking*) e exploradas em ataques *replay* (*replay attack*);
- O emprego indevido de ferramentas que são utilizadas para o fim de invadir ao invés de obter informações do sistema para o seu gerenciamento e administração. Ex: *port scanning*, que é utilizado para identificar as portas ativas do sistema e os serviços providos por cada porta e o *packet sniffing*, que normalmente diagnostica problemas na rede, mas pode também ser usado para capturar pacotes que trafegam pela rede;
- Configuração, administração ou manutenção imprópria de sistemas;

- Projeto do sistema ou capacidade de detecção ineficiente, como um sistema de detecção de intrusão - *Intrusion Detection System* (*IDS*) que forneça informações falsas, erradas ou exageradas.

Sabendo-se que para se atacar um sistema, deve-se levantar o maior número de informações a respeito deste, algumas técnicas são empregadas para a obtenção destas informações, o que consiste no primeiro passo para um ataque de sucesso. Estas técnicas estão descritas a seguir.

2.1 Ataques para Obtenção de Informações

2.1.1 *Dumpster diving* ou t*rashing*

Basicamente consiste em revirar o lixo das empresas à procura de informações pessoais e confidenciais. (REZENDE, 2005). Geralmente em documentos de bancos, por exemplo, diversas informações preciosas podem ser obtidas em papéis que vão para o lixo, como a matrícula de um funcionário, números de contas-correntes de clientes, suas informações pessoais como endereços e telefones, que podem vir a ser utilizados em um outro tipo de técnica: a engenharia social. Portanto, um fragmentador de papel deve fazer parte da política de segurança destas organizações. (CERQUEIRA, 2005)

2.1.2 Engenharia social

Esta técnica tem por objetivo enganar e ludibriar pessoas, atacando o elo mais fraco da segurança: o usuário. Estes atacantes se fazem passar por funcionários da empresa ou por responsáveis técnicos do sistema, a fim de obter senhas ou outras informações que comprometam a segurança da organização. (REZENDE, 2005)

2.1.3 Informações Livres

Abrange as informações que são obtidas principalmente da internet, através da consulta a Servidores de Nome de Domínio (DNS), análise de cabeçalhos de *e-mail* e busca de informações em listas de discussão. Por meio delas, detalhes sobre sistemas, topologia e usuários podem ser obtidos facilmente. (CERQUEIRA, 2005)

2.1.4 Packet sniffing

Também conhecida como "*passive eavesdropping*", esta técnica consiste na captura de informações valiosas diretamente pelo fluxo de pacotes na rede. (NAKAMURA, 2003)

Estes pacotes são exibidos na tela ou armazenados em disco para análise posterior.

Esta ferramenta, por ser transparente aos dispositivos de rede e quase impossível de ser detectada, tornou-se uma das principais utilizadas por invasores para fins não éticos, embora tenha sido criada com o objetivo de ajudar na administração de rede. (NUNES, 2004)

Um programa de *sniffing* altera o funcionamento do adaptador. Este alterado passa a funcionar em "modo promíscuo", isto é, independente do endereço de destino do *frame* ser igual ou não ao endereço da placa, ele é lido como se fosse dele. Todos os computadores conectados a uma rede com *sniffing* ficarão vulneráveis, pois bastam alguns minutos para garantir uma boa quantidade de informação, desde senhas de acesso, números de cartões de crédito até trechos de documentos sigilosos da empresa armazenados, por segurança, no servidor de arquivos. (NUNES, 2004)

Os protocolos mais vulneráveis a um ataque de *sniffing* são aqueles que transportam dados sem qualquer tipo de codificação. Inclui-se nesta categoria: Telnet, rlogin, HTTP, SNMP, NNTP, POP, FTP, IMAP, além de serviços com *talk*, *finger*, *whois*, etc. (NUNES, 2004)

2.1.5 Port scanning

São ferramentas que fazem a varredura de portas TCP (*Transmission Control Protocol*) e UDP (*User Datagram Protocol*), podendo identificar quais estão abertas, os serviços que estão rodando nelas, permitindo assim sua exploração e o aproveitamento de suas vulnerabilidades. (MARTINS, 2003)

2.1.6 Scanning de vulnerabilidades

Após a utilização do *port scanning*, que identifica os sistemas que podem ser atacados e os serviços que são executados, as vulnerabilidades de cada um destes serviços serão procuradas por meio da “varredura” de vulnerabilidades.

Os *scanners* de vulnerabilidades realizam diversos tipos de testes na rede à procura de falhas de segurança, seja em protocolos, serviços, aplicativos ou sistemas operacionais. (NAKAMURA, 2003)

Esta varredura é executada tanto para a infra-estrutura como para os processos. Na infra-estrutura geralmente a varredura é executada de forma automática. Estes tipos de *software*, dado um endereço IP, procuram identificar todos os meios de acesso aos elementos da rede (roteadores, servidores, estações e outros): (MARTINS, 2003)

- Faz uma varredura de todas as portas abertas do TCP/IP;
- Para cada porta aberta, o *software* tenta conectar-se ao serviço e avaliar sua parametrização. Por exemplo:
    - Analisa se o serviço FTP está disponível e se aceita conexões com o usuário anônimo;
    - Verifica se o TELNET está acessível, pois a partir deste serviço é possível acessar o sistema operacional;
    - Se o serviço de compartilhamento estiver ativo, verifica se os diretórios estão protegidos com senha ou não e, em caso afirmativo, tenta quebrar a senha (para algumas versões de sistema operacional isto é relativamente simples);
    - Coleta informações sobre o sistema operacional, com tipo e versão, assim como os *services packs* instalados;
    - Verifica se o serviço http está disponível e qual o tipo de serviço utilizado (ISS, Apache, etc);
    - Verifica se o serviço SNMP está ativo e qual o seu nível de segurança.
- O *software* de varredura vem programado com todos os *bugs* conhecidos nas aplicações mais populares da rede, como servidores de *web* e FTP, aplicativos de *e-mail*, ferramentas de desenvolvimento e outros. No processo de varredura,

todos esses *bugs* são testados, para verificar se foram corrigidos ou não. (MARTINS, 2003)

Alguns exemplos de vulnerabilidades identificadas pelos *softwares* de varredura: (MARTINS, 2003)

- O *software* tenta se conectar nos serviços usando contas padrão com senha em branco, senha igual ao nome do usuário, e senhas fracas como, por exemplo, seqüência de caracteres como “123456”, “abc”, etc;
- Verificação do privilégio de acesso aos diretórios e serviços;
- Sistemas com opção de “*autologon*”, onde o computador é configurado para se “logar” automaticamente numa conta sempre que for reiniciado.

Concluída a varredura, o *software* gera um relatório com tudo que foi identificado nos dispositivos associados aos IP’s analisados.

Um importante ponto a ser considerado é que o conteúdo reportado pelo *scanner* deve ser conferido individualmente, porque podem ocorrer casos de falso positivos e falso negativos, por isso a ferramenta deve estar sempre atualizada, pois funciona por meio de uma base de dados de ataques conhecidos. Outro ponto importante é que assim como estes *scanners* servem para auxiliar os administradores na proteção da rede, podem ser utilizados por *hackers* para detecção de suas falhas. (NAKAMURA, 2003)

2.1.7 Firewalking

Firewalking é uma ação que emprega técnicas como as do *traceroute* para analisar respostas a pacotes IP, para determinar filtros ACL de *gateway* e mapear redes protegidas por um firewall. A ferramenta *Firewalk* emprega tais técnicas para determinar as regras de filtragem em um dispositivo de *forwarding* de pacotes. Essa técnica permite que pacotes passem por portas em um gateway, além de determinar se um pacote com várias informações de controle pode passar pelo *gateway*.

Pode-se ainda mapear roteadores encontrados antes do *firewall*. Isso é possível devido à possibilidade de mudar o campo *Time to Live* (TTL) do pacote e as portas utilizadas, que permitem que as portas abertas pelo firewall sejam utilizadas para o mapeamento da rede. (NAKAMURA, 2003)

2.1.8 IP spoofing

Esta técnica consiste na falsificação de endereços IP, permitindo que uma máquina utilize-se do endereço IP de outra máquina, escondendo sua verdadeira identidade. Apesar do *Spoofing* poder ocorrer com diversos protocolos específicos, o *Spoofing* do IP é o mais conhecido dentre todos os ataques deste gênero. (NETSEC, 2004)

O processo de um ataque de *Spoofing* envolve diversas etapas: primeiro é necessário identificar duas máquinas de destino A e B, que possuam um relacionamento confiável uma com a outra (ex: duas máquinas em uma rede interna). É esse relacionamento que o ataque tenta explorar. (Boletins de Segurança, 2004)

Logo após, o farsante tentará estabelecer uma conexão com a máquina B fazendo-se passar por A, quando na verdade, é a máquina do farsante. Para isso, é gerada uma mensagem falsa (uma mensagem criada na máquina do farsante, mas que contém o endereço de origem de A), solicitando uma conexão com B. Ao receber a mensagem, B responderá com uma outra mensagem que reconhece a solicitação e estabelece números de seqüência. Porém, esta mensagem será direcionada a máquina A, com a qual B acredita estar se conectando. Neste caso, o farsante terá que tentar adivinhar os números de sequência que B utilizará. Além disso, também deverá impedir que a mensagem de B chegue até A. Se a mensagem chegasse até A, A negaria ter solicitado uma conexão e o ataque falharia. Para alcançar este objetivo, normalmente o intruso envia diversos pacotes a máquina A para esgotar sua capacidade e impedir que ela responda à mensagem de B. Essa técnica é conhecida como “violação de portas”. Uma vez que essa operação tenha chegado ao fim, o violador poderá concluir a falsa conexão.

Esta técnica também pode ser utilizada para ocasionar um *Denial of Service,* pois um servidor que receba milhares de pacotes *IP Spoofing* terá uma sobrecarga no *host* quando do tratamento a estes pacotes. (NETSEC, 2004)

2.2 Ataques de Negação de Serviços

Ataques de Negação de Serviços tem como objetivo paralisar (derrubar) um serviço em um servidor, ou então tornar os serviços tão lentos que o usuário legítimo não consegue acessá-los. (AUTOSCAN, 2004).

2.2.1 *Bugs* em serviços, aplicativos e sistemas operacionais

Bugs são falhas na programação dos softwares encontradas em quase todos os programas. (LIMA JR, 2004) Estas falhas são exploradas por *hackers* em tentativas de invasão.

A concorrência do mercado é a responsável por estas falhas, pois a qualidade do software não é mais prioridade e sim a velocidade com que é lançado no mercado. Desta forma, estes *softwares* são comercializados sem o tempo necessário de exposição a testes, sendo liberado com muitos erros que, futuramente, serão explorados por *hackers* em tentativas de invasão.

2.2.2 *SYN flooding*

A característica dos pacotes de *Syn Flooding* é a geração de um grande número de requisições de conexões (pacotes *SYN*) enviadas a um sistema, de tal maneira que o servidor não é capaz de responder a todas estas requisições. A pilha de memória sofre um *overflow*, e as requisições de conexões de usuários legítimos acabam sendo desprezadas, ocasionando negação de serviços.(DANDREA,1999)

A identificação do invasor é difícil de ser detectada, pois os ataques são realizados forjando o endereço IP de origem, com endereços IP's de *hosts* que não estão *online.* (SILVA, 2004)

2.2.3 Fragmentação de pacotes de IP

Este tipo de ataque explora um erro na implementação do protocolo TCP/IP que trata da fragmentação de pacotes. Quando os pacotes são divididos – por possuírem tamanho maior que o permitido pela rede – quando chegam ao seu destino estes devem ser reagrupados. Neste momento o sistema operacional trava, somente voltando a funcionar após ser reinicializado. Os sistemas operacionais (SO) mais vulneráveis a este tipo de ataque são o *Windows* e o *Linux.* (MARTINS, 2003)

2.2.4 *Smurf* e *fraggle*

São ataques em nível de camada de rede e de transporte, no qual um grande tráfego de pacotes *ping* (ICMP *echo*) é enviado para o endereço IP de *broadcast* da rede, tendo como origem o endereço IP de uma vítima (*IP Spoofing*). Assim o *host*, atacado, acaba recebendo uma sobrecarga de pacotes, acarretando DoS (*Denial of Service). O Fraggle* é semelhante ao *Smurf,* a diferença é que utiliza pacotes UDP *e o Smurf* TCP. As redes são afetadas, pois todos respondem às solicitações. Isto acaba prejudicando as funções normais que deveriam ser executadas, pois estão ocupadas com o *broadcast* amplificado. (WEBER, 2000) Estas formas de ataque são explicadas através da figura1.

Figura 1 - Forma de Ataque *Smurf* e *Fraggle*

2.2.5 *Teardrop*, *nuke e land*

O *teardrop* é uma ferramenta utilizada para explorar os problemas de fragmentação IP nas implementações do TCP/IP. O *Land* é um ataque que tem como objetivo travar o computador da vítima. O ataque é efetuado enviando-se um pacote TCP para qualquer porta do computador destino com a flag *SYN* habilitada. O pacote é montado de tal forma que os endereços de origem e de destino são os mesmos (*spoofing*). Alguns sistemas operacionais não conseguem processar este tipo de pacote, fazendo com que o computador pare de funcionar. Geralmente este ataque é direcionado à porta 139 (*NetBios*) de computadores com sistema operacional *Windows*. O *Nuke* é semelhante ao *Land*, a diferença é que a *flag* habilitada, neste caso, é a URG (urgente). Após receber o pacote, o SO fica aguardando um *stream* de dados da chamada *Out-of-band* e, como estes dados têm prioridade em relação ao tráfego normal, o computador ficará travado enquanto estes dados não vierem. (MARTINS, 2003)

2.3 Ataque Ativo contra o TCP

2.3.1 Seqüestro de conexões

Uma conexão de TCP entre dois pontos é realizada de modo *full-duplex*, sendo definida por quatro informações: endereço IP do cliente, porta de TCP do cliente, endereço IP do servidor, porta de TCP do servidor. Todo o *byte* enviado por um *host* é identificado com um número de seqüência de 32 bits, que é reconhecido pelo receptor utilizando esse número de seqüência. O número de seqüência do primeiro *byte* é computado durante a abertura da conexão e é diferente para cada uma delas, de acordo com regras designadas para evitar sua reutilização em várias conexões.

O ataque tem como base a exploração do estado de dessincronização nos dois lados da conexão de TCP, que assim não podem trocar dados entre si pois, embora ambos os *hosts* mantenham uma conexão estabelecida, os pacotes não são aceitos devido a números de seqüência inválidos. Desse modo, um terceiro *host*, o do atacante, cria os pacotes com números de seqüência válidos, colocando-se entre os dois *hosts*. (NAKAMURA, 2003)

Este tipo de ataque pode se apresentar através de duas formas: (REZENDE, 2005)

O atacante intercepta os dados e responde pelo cliente, podendo alterar os dados (figura 2) ou,

figura 2 - ***man-in-the-middle***

O atacante derruba o cliente e mantém a conexão em andamento. (figura 3)

Figura 3 - ***session hijacking***

2.3.2 Prognóstico de número de seqüência do TCP

Por apresentarem em alguns sistemas comportamento-padrão, como incremento de 128 ou 125 mil a cada segundo, é relativamente fácil descobrir os números de seqüência dos pacotes TCP em cada pacote. Isto possibilita que o *hacker* utilize esta informação para se inserir em uma conexão. Atualmente, alguns sistemas implementam padrões de incremento do número de seqüência mais eficientes, que dificultem seu diagnóstico e os ataques. (NAKAMURA, 2003)

2.3.3 Ataque de Mitnick

Um dos incidentes de segurança mais famosos que já ocorreram é, sem dúvida, o bem sucedido ataque de Kevin Mitnick ao sistema do pesquisador Tsutomu Shimomura, em 1994. Para tanto, Mitnick explorou vulnerabilidades bem conhecidas do protocolo TCP, ainda presentes em muitas implementações.

O ataque usou duas técnicas: *SYN flooding* e o seqüestro de conexões TCP. Enquanto a primeira técnica causa uma negação de serviço no sistema alvo, silenciando sua atividade de rede pela incapacidade de tratar tantos pedidos de novas conexões, a segunda passa a agir no seu lugar através do seqüestro de conexões TCP, explorando relações de confiança existentes entre a máquina alvo e outros computadores da rede interna (arquivo *rhosts*).(CAMPELLO, 2005)

2.3.4 Source routing

Source routing é a habilidade de lidar com um pacote de modo que este seja direcionado a certos roteadores sem que passe pelos roteadores convencionais. Tipicamente, utiliza-se *source routing* quando um roteador executa o bloqueio de algum tipo de tráfego que o invasor deseja explorar, onde o roteamento é alterado na tentativa de burlar o dispositivo de conectividade. (FREIRE, 2004)

2.4 Ataques Coordenados

Também chamados de DDos (*Distribuited Denial of Service*) este tipo de ataque faz com que vários *hosts* sejam atacados e manipulados para que eles realizem um ataque simultâneo a uma determinada vítima. (ANÔNIMO, 2001) A sobrecarga é inevitável e, como o acesso vêm de muitas origens diferentes (como mostra a figura 4), fica difícil filtrar o que são conexões provenientes de ataques ou não. (LINUX, 2004)

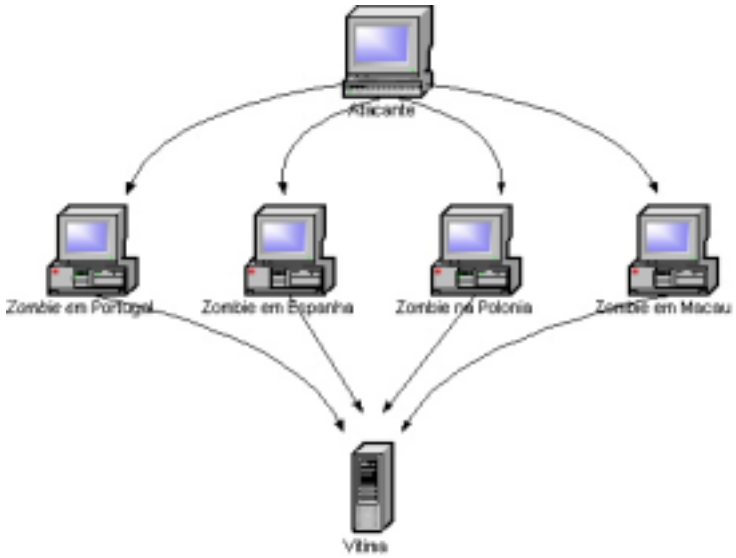

figura 4 - Estrutura de um ataque DdoS

2.5 Ataques no Nível da Aplicação

2.5.1 *Buffer overflow*

O *Buffer Overflow* é decorrente de uma falha de programação advinda da não validação dos dados de entrada recebidos por uma variável, isto é, quando não há a verificação do limite máximo de *bytes* que a variável em questão pode comportar. (NETSEC, 2004)

Entre as conseqüências da exploração deste tipo de falha de implementação, estão o fato de poderem acarretar inoperância do sistema e servirem para causar *Denial of Service. (*AUTOSCAN, 2004) Também podem ser usadas para executar códigos arbitrários nos sistemas sendo, portanto, de grande risco.

2.5.2 Ataques na web

Bugs em servidores *web*, navegadores de internet, *scripts Common Gateway Interface* (CGI) e scripts *Active Server Pages* (ASP), são as vulnerabilidades mais exploradas, mais simples e mais comuns de serem vistas. Pesquisas demonstram que os ataques à Internet na América Latina aumentaram mais de 20% de julho a outubro em relação ao mesmo período de 2003. Os ataques mais comuns foram os de vários

vírus, como Sasser e Korgo, destinados a explorar a vulnerabilidade dentro do LSASS, um componente de segurança do sistema operacional *Windows.* Estes ataques têm como característica principal a busca por pontos fracos no *software* do servidor *web* (por exemplo, Microsoft IIS, Apache http Server e Netscape iPlanet). (Fonte: IBM)

2.5.3 Problemas com o SNMP

O protocolo SNMP (*Simple Network Management Protocol*) é muito usado pelos administradores de rede para monitorar e administrar todos os tipos de equipamentos conectados à rede, desde roteadores e impressoras, até servidores e estações de trabalho. O tráfego SNMP, quando interceptado ("*sniffed*"), pode revelar muitas informações sobre a estrutura de sua rede, bem como dos sistemas e os equipamentos conectados a ela. Os invasores usam tais informações para escolher alvos e planejar os ataques. (ROCHA, 2005)

Algumas falhas deste protocolo é que ele não possui travamento de senhas, permitindo ataques de força bruta e sua configuração padrão pode anular alguns esforços de segurança pretendidos com a utilização de certas ferramentas para tal fim, como, por exemplo, o *RestrictAnonimusKey* para o SNMP do windows NT, que bloqueia informações como os nomes de usuários, os serviços que estão sendo executados e os compartilhamentos dos sistemas. (MCCLURE, 1999)

2.5.4 Vírus, w*orms* e cavalos de tróia

Os vírus são pequenos programas desenvolvidos com o intuito de se espalharem, infectando diversas máquinas. Atacam geralmente programas ou o setor de *boot* do disco rígido. Necessitam de outro código executável para serem acionados. Possuem um mecanismo de ativação (evento ou data) e uma missão (apagar arquivos, enviar dados, etc) que se propagam (anexando-se a arquivos e programas). Normalmente são encontrados em microcomputadores.

Os *worms* são programas que trafegam através da rede e podem rodar independentemente. Geralmente não modificam outros programas, embora possam carregar outros códigos que o façam (vírus, por exemplo).

O cavalo de tróia é um programa que parece ter uma função, quando na realidade, executa outras funções. Normalmente se parecem com programas comuns, amplamente utilizados (como um editor de textos), mas por trás da interface adulterada estão executando operações menos nobres, como apagando arquivos, reformatando discos ou alterando dados. São utilizados como veículo para vírus, *worms* e outras ameaças programadas. (DIAS, 2000)

2.5.5 War dialing

Método utilizado por *hackers* e *crackers* para encontrar redes desprovidas de proteção contra invasão. Baseia-se no uso de um modem, que disca seqüencialmente para diversos números de telefone, numa determinada região, armazenando informações sempre que um outro modem for identificado na ponta oposta da linha. (PELEGRINA, 2004)

3 TIPOS DE TESTES DE SEGURANÇA

Há vários tipos de testes de segurança. Alguns métodos são empregados predominantemente por humanos, e outros tipos são automatizados. Freqüentemente algumas destas técnicas de teste são usadas em conjunto para ganhar maior foco de avaliação total da segurança da rede global. Um exemplo é o teste de intrusão que se utiliza do mapeamento de rede e escaneamento de vulnerabilidades para identificar os computadores servidores vulneráveis e serviços que podem ser alvo para posterior penetração. É um dos testes mais abrangentes, pois além de mapear a rede e escanear em busca de suas vulnerabilidades, seu objetivo principal é a utilização destas informações para auxiliar na invasão do sistema alvo, provando assim que estas vulnerabilidades realmente são uma ameaça e estão permitindo o acesso ao sistema. Alguns tipos de testes de segurança serão descritos a seguir. (WACK, 2002)

3.1 Mapeamento de Rede

O mapeamento de rede é feito através da utilização de um *port scanner* que identifica computadores servidores ativos em uma rede em uma organização, os serviços de rede operando nestes computadores e a aplicação específica que executa o serviço identificado.

É de grande importância a escolha de uma ferramenta de *scanner* apropriada, pois certos *scanners* são mais adequados para o trabalho com *firewalls* enquanto outros são melhores para escaneamento interno ao *firewall*.

Alguns *scanners*, além de relacionarem as portas abertas e os servidores ativos, fornecem informações adicionais, levando geralmente a descoberta do sistema operacional alvo. Por exemplo, um servidor com portas de TCP 135 e 139 abertas, é bem provável que seja um servidor *windows* NT ou 2000. Porém, alguns *firewalls* bloqueiam certas portas e administradores de sistema podem camuflar o sistema operacional através do uso de configuração não - padrão.

Uma importante limitação do *scanner* de portas é que não identifica vulnerabilidades. Os dados obtidos com o escaneamento precisam da interpretação humana para serem desvendados.

Este tipo de teste pode ser empregado com o objetivo de:

- Checar computadores servidores não autorizados conectados a rede da organização;
- Identificar serviços vulneráveis;
- Identificar desvios dos serviços permitidos definidos na política de segurança da organização;
- Preparar para o teste de penetração.

Alguns problemas que podem ser gerados pela aplicação dos *port scanners* são o comprometimento das operações de redes pelo consumo de largura de banda e diminuição do tempo de resposta. (WACK, 2002)

3.2 Escaneamento de Vulnerabilidades

Os *scanners* de vulnerabilidades, além de identificarem portas abertas e servidores, geram automaticamente respostas as vulnerabilidades relacionadas com o resultado obtido pelo escaneamento, dispensando a análise humana neste processo. Também identificam aplicações, sistemas operacionais, captura de *banner* e vulnerabilidades associadas com sistemas operacionais descobertos e aplicações. É uma forma relativamente rápida e fácil de quantificar a exposição de uma organização à superfície de vulnerabilidade.

O emprego deste teste isoladamente pode gerar uma análise de risco falha, pois a ferramenta não possui a capacidade de analisar o verdadeiro nível de risco de pequenas vulnerabilidades associadas, somente o risco isoladamente (falso positivo). Outra limitação é a constante necessidade de atualização do banco de dados de vulnerabilidades. Estas ferramentas são capazes de identificar versões de programas antigas, vulnerabilidades, caminhos aplicáveis ou melhorias no sistema e a validar conformidades ou desvios da política de segurança da organização. Para isso, empregam grandes bancos de dados de vulnerabilidades para identificar

vulnerabilidades associadas com sistemas e aplicações de operações comumente usadas. Podem fazer certas correções automaticamente. Por requererem mais informação, geram mais tráfego de rede do que o *scanner* de porta, gerando impacto negativo nos servidores alvo ou nas redes nas quais estão cruzando. (WACK, 2002)

3.3 Teste de Segurança e Avaliação

O Teste de Segurança e Avaliação (ST & E) é um exame ou análise de medidas de proteção que são colocadas em um sistema de informação uma vez que é completamente integrado e operacional. Os objetivos do ST & E são:

- Revelar erros de operações, implementação e projeto que poderiam permitir a violação da política de segurança;
- Determinar a adequação dos mecanismos de segurança, certificações e outras propriedades que reforcem a política de segurança;
- Avaliar o grau de consistência entre a documentação do sistema e sua implementação.

O alcance de um plano ST & E tipicamente enfoca segurança do computador, segurança das comunicações, segurança das emanações, segurança física, segurança pessoal, administrativa e de operações. Segurança do computador é compreendida entre as medidas e controles que protegem o sistema contra DoS e descobertas não autorizadas, ou a destruição dos dados do sistema. A segurança do computador pode ser testada através de teste de configuração e operacional para validar aqueles mecanismos de segurança do sistema que foram implementados e estão funcionando apropriadamente. O teste de configuração é desempenhado pela comparação da configuração instalada contra a configuração aprovada encontrada nas exigências de segurança, conceito de segurança e operações, ou um outro documento similar. O teste operacional oferece uma avaliação dos mecanismos de sistema de segurança em um ambiente operacional para determinar se os mecanismos estão reforçando a política de segurança do *site*. (WACK, 2002)

O teste operacional é desempenhado através da execução de testes predefinidos. Estes testes estabelecem uma linha base para gerenciamento de configuração e teste de sistema.

A segurança de comunicação compreende as medidas e controles tomados para evitar acesso não autorizado através das telecomunicações. O teste é realizado para assegurar que os *links* de comunicações sejam protegidos a um nível comensurado com o nível de sensibilidade dos dados a serem transferidos. Adicionalmente, os testes de comunicação deveriam determinar que a conexão do sistema não introduzisse novas vulnerabilidades na rede.

Segurança de emanações analisa sinais que são relacionados a dados que, se interceptados e analisados, revelam a transmissão da informação recebida, manipulada, ou do contrário, processada por qualquer equipamento de processamento de informação. Teste de segurança de emanações é desempenhado através de interceptação e análise de sinais eletrônicos emitidos do sistema.

A porção de segurança física de ST & E é feita para determinar se o ambiente físico onde o sistema reside é adequado para a proteção e operação do sistema. Esta parte do teste é feita através da análise das características de segurança do produto, sua adequação a proteção do sistema, e as condições ambientais para assegurar que um ambiente operacional apropriado possa ser mantido.

Segurança pessoal é o processo pelo qual a confiabilidade e adequação do pessoal são verificadas. Para o ST & E, isto inclui assegurar que o acesso ao equipamento é limitado somente ao pessoal que dispõe do acesso (possui permissão).

Segurança administrativa compreende os limites de gerenciamento e controles suplementares estabelecidos para prover um nível aceitável de proteção para os dados.

Segurança administrativa é também conhecida como segurança procedural. A testagem para a seção administrativa do ST & E deveria incluir análise do projeto e adoção de procedimentos adotados diariamente para a operação do sistema. Esta parte do teste deveria também determinar a adequação do plano de contingência do site.

Segurança de operações é um processo analítico pelo qual os adversários potenciais têm informações negadas sobre as capacidades e intenções pela identificação, controle e proteção de evidência do planejamento e execução de atividades sensíveis e operações. Segurança de operação em ST & E é feita através da análise da habilidade destes sistemas em limitar o acesso a esta informação. O benefício derivado das operações de teste de segurança é que ela verifica a informação que não esta sendo dada aos adversários que os ajudaria a tirar vantagem da segurança da rede. (WACK, 2002)

3.4 Quebra de Senha

Programas de quebra de senha podem ser usados para identificar senhas fracas. Estes programas verificam se os usuários estão empregando senhas suficientemente fortes. As senhas são geralmente guardadas e transmitidas de uma forma criptografada chamada de *hash* (algoritmo usado para produzir um código *hash* para uma entrada e assegurar que este código é único para cada entrada).

Quando um usuário entra em um computador ou sistema e entra com a sua senha, um *hash* é gerado e comparado ao *hash* guardado. Se os *hashes* guardado e colocado combinam, o usuário é autorizado.

Durante o teste de penetração ou em um ataque real, a quebra de senha usa *hashes* de senha capturados. Senhas de *hashes* podem ser interceptadas quando elas são transmitidas ao longo da rede (usando um “sniffer” de rede) ou eles podem ser encontrados a partir do sistema alvo. O último geralmente requer acesso *root* ou administrativo ao sistema alvo.

Uma vez que os *hashes* são obtidos, um decodificador de senha automatizado rapidamente gera *hashes* até que a combinação seja encontrada. O método mais rápido de gerar *hashes* é um dicionário de ataque que usa todas as palavras de um dicionário ou arquivo de texto. Há muitos dicionários disponíveis na internet que cobrem a maior parte das linguagens, nomes, *shows* favoritos de televisão mais e menos importantes e etc. Então qualquer palavra de dicionário, não importa o quão vaga ela seja, é fraca.

Um outro método de decodificar é um ataque híbrido, que complementa o método do dicionário por adição de caracteres numéricos e simbólicos ao dicionário de palavras. Dependendo do decodificador de senhas a ser usado, este tipo de ataque tentará um número de variações. Ele testará o substituto comum de caracteres e números para letras (ex. p@assword e h$ckme). Alguns também tentarão adicionar caracteres e números ao início e fim de um dicionário de palavras (ex. Password99, password$%, etc).

O método mais poderoso de decodificar senhas é chamado de método da força bruta. Embora força bruta leve um longo tempo, ele geralmente leva menos tempo que a maior parte das políticas de senhas especificam para uma troca de senha. Conseqüentemente, as senhas encontradas durante os ataques de força bruta ainda são muito fracas. A força bruta gera aleatoriamente senhas e seus *hashes* associados. Entretanto, uma vez que há tantas possibilidades, ele pode levar meses para quebrar uma senha. Teoricamente, todas as senhas são quebráveis para um ataque de força bruta, dado tempo suficiente e poder de processamento.

Avaliadores de penetração e *hackers,* com freqüência, tem múltiplas máquinas por onde eles podem espalhar o ataque de quebrar senhas. Isto pode encurtar significativamente o tempo exigido para a tarefa de checar senhas. Uma senha forte é aquela que é longa (maior do que dez caracteres, no mínimo) e complexa (contém tanto letras maiúsculas quanto minúsculas, caracteres e números).

Decodificadores de senha deveriam ser executados no sistema em uma base mensal ou até mesmo continuamente para assegurar composição correta de senha ao longo da organização. As seguintes ações podem ser feitas, se um número inaceitavelmente alto de senhas forem quebradas:

- Se as senhas decodificadas foram selecionadas conforme a política, a política deveria ser modificada para reduzir a porcentagem de senhas decodificáveis. Se tal modificação de política levar os usuários a anotar suas senhas porque elas são difíceis de memorizar, uma organização deveria considerar substituir a autenticação de senha por uma outra forma de autenticação;
- Se as senhas quebradas não foram selecionadas conforme a política, os usuários deveriam ser educados sobre os possíveis impactos de seleção de senhas fracas.

Se tal violação pelos mesmos usuários for persistente, a gerência deve considerar uma ação disciplinar contra tais usuários. Muitas plataformas de servidores também permitem ao administrador do sistema estabelecer comprimento mínimo de senha e complexidade. (WACK, 2002)

3.5 Revisões de Arquivos de Transações (*log*)

Vários sistemas de *logs* podem ser usados para identificar desvios da política de segurança da organização, incluindo arquivos de transação de *firewall*, *logs* IDS, *logs* de servidor e qualquer outro *log* que junte dados do sistema ou rede.

Enquanto não é considerada tradicionalmente uma atividade de teste, a revisão de *log* e a análise podem oferecer uma idéia dinâmica das atividades em andamento do sistema que podem ser comparadas com o conteúdo e intenção da política de segurança. Essencialmente, auditoria de *logs* pode ser usada para validar se o sistema esta operando conforme as políticas.

Por exemplo, se um sensor de IDS for colocado atrás de um *firewall* (dentro do enclave) seus *logs* podem ser usados para examinar os pedidos de serviços e comunicações que estão sendo permitidas dentro da rede através do *firewall*. Se este sensor registra atividades não autorizadas além do *firewall*, ele indica que o *firewall* não esta mais configurado corretamente.

Revisão de *log* com auditoria manual é extremamente lenta e enfadonha. Ferramentas de auditoria automatizada oferecem uma forma de reduzir significativamente o tempo de revisão exigido e impressão dos relatórios (predefinidos ou adaptados) que resumiriam os conteúdos do *log* para um conjunto de atividades específicas. É difícil dizer que quaisquer filtros aplicados ao *log* filtrem o que não é preciso e passem qualquer coisa mais. Se for filtrado apenas o que é necessário então qualquer evento de exceção também será filtrado.

Revisões de *log* deveriam ser conduzidas pelo menos semanalmente, não considerando como os resultados serão usados. Para a específica finalidade de testar a implementação de configurações exigidas de segurança, uma freqüência mensal pode

ser suficiente com a exceção de uma revisão de demanda resultando de uma importante melhoria do sistema que requer validação.

As seguintes ações podem ser tomadas se um sistema não esta configurado conforme as políticas:

- Re-configurar o sistema como exigido para reduzir a chance de comprometimento, ou mudar a política de firewall para limitar o acesso ao sistema, ou serviço vulnerável;
- Mudar a política de firewall para limitar os acessos da sub-rede IP que é a fonte de comprometimento. (WACK, 2002)

3.6 Checagem de Integridade de Arquivo

Uma checagem de integridade de arquivo computa e estoca uma soma de checagens para todo o arquivo guardado e estabelece um banco de dados de somas de checagens de arquivo. Ele oferece uma ferramenta para o administrador do sistema reconhecer mudanças nos arquivos, particularmente mudanças não autorizadas. A soma de verificação estocada deveria ser re-computada regularmente para testar o valor atual contra o valor estocado para identificar qualquer modificação dos arquivos. A capacidade de um verificador de integridade de arquivo é geralmente incluída com qualquer sistema comercial de detecção de intrusão baseada em computador servidor.

Enquanto um verificador de integridade é uma ferramenta útil que não requer um alto grau de interação humana, ele precisa ser usado cuidadosamente para assegurar que ele é efetivo. Um verificador de integridade de arquivo requer um sistema que é conhecido como seguro para criar o primeiro banco de dados de referência. Do contrário, *hashes* criptográficos de um sistema comprometido podem ser criados e, portanto, criar um falso senso de segurança para o avaliador. O banco de dados de referência deveria ser estocado não conectado a uma rede de forma que um transgressor não pode comprometer o sistema e esconder seus ataques ao modificar o banco de dados. Um verificador de integridade de arquivo pode também gerar alarmes falsos positivos. Cada atualização de sistema e implementação de conserto de arquivo muda o arquivo e irá, portanto, requerer uma atualização do banco de dados do total da

verificação. Portanto, manter um bando de dados atualizado pode ser difícil. Entretanto, ate mesmo se o verificador de integridade é executado somente uma vez (quanto o sistema é instalado pela primeira vez) ele pode ainda ser uma atividade útil para determinar quais arquivos tem sido modificados, em caso de uma suspeita de comprometimento. Finalmente, os transgressores têm demonstrado uma habilidade para modificar um arquivo de forma que um verificador de redundância cíclica de 32-*bits* (CRC) de total de verificação não poderia detectar. Portanto, totais de verificação mais fortes são recomendados para assegurar a integridade dos dados que são estocados no banco de dados do total da verificação.

Verificadores de integridade poderiam ser conduzidos diariamente em uma seleção de arquivos de sistema que seriam atacados por um comprometimento. Verificadores de integridade deveriam também ser usados quando um comprometimento é suspeito, ao determinar a extensão do dano possível. Se um verificador de integridade detecta modificações de arquivo de sistema não autorizado, a possibilidade de um incidente de segurança deveria ser considerada e investigada conforme a resposta e relato de incidente da empresa e procedimento de política. (WACK, 2002)

3.7 Detector de Vírus

Todas as organizações estão em risco de ”contrair” vírus de computador. *Trojans* e *worms* (vermes), se conectados à *internet*, usam mídia removível (ex. disquetes e cd-roms), permitem acesso não supervisionado aos usuários ou usam programas compartilhados.

Um vírus de computador é uma série de códigos que se anexa a um outro programa de computador ou documento. Uma vez que ele está anexado, ele se multiplica usando alguns dos recursos do programa cooptado ou documento para replicar e se anexar a outros programas, *hosts* e documentos. Códigos maliciosos não estão limitados a vírus: há vários tipos de códigos maliciosos que são geralmente detectados pelos programas de antivírus até mesmo se o código não é, estritamente falando, um vírus. As outras categorias de códigos malígnos incluem *worms*, *trojans* e códigos móveis malignos.

O impacto de um vírus, *trojan*, *worm* ou código móvel maligno pode ser sem danos, como uma mensagem instantânea na tela do computador, ou altamente destrutivo, como apagar todos os arquivos do disco rígido. Como qualquer código maligno, há também o risco de expor ou destruir informação confidencial ou sensível.

Há dois tipos principais de programas antivírus disponíveis: aqueles que são instalados na infraestrutura da rede e aqueles que são instalados nas máquinas do usuário final. Cada um tem vantagens e desvantagens, mas ambos, usados em conjunto, são geralmente requeridos para o mais alto nível de segurança.

O detector de vírus instalado na infraestrutura da rede é baseado no servidor e é geralmente instalado em servidores de correspondência e etc, ou em conjunção com *firewalls* na fronteira da rede de uma organização. A vantagem dos programas de detecção de vírus baseados no servidor, é que eles podem detectar vírus antes que eles entrem na rede ou que um usuário baixe o seu *e-mail*. A outra vantagem da detecção de vírus baseada no servidor é que todos os detectores de vírus requerem freqüente atualização para permanecerem efetivos. Isto é muito mais fácil de realizar nos programas baseados no servidor, devido ao número limitado relativo aos clientes de computadores servidores. Infelizmente, programas baseados no usuário podem ter um efeito negativo no desempenho da rede.

O outro tipo de programa de detecção de vírus é instalado nas máquinas do usuário final. Este programa detecta códigos malignos em *e-mails*, discos rígidos e disquetes, documentos e etc, mas somente para o computador servidor local. Ele também detecta, às vezes, código maligno de sites da *web*. Este tipo de programa de detecção de vírus tem menos impacto no desempenho da rede, mas geralmente depende que os usuários finais atualizem sua assinatura, o que não é sempre confiável. Também a proteção contra vírus no usuário final não pode proteger a rede de todas as ameaças de vírus.

Não importa qual tipo de programa de detecção de vírus é usado, ele não pode oferecer sua completa proteção, a menos que ele tenha um banco de dados de identificação de vírus atualizado (às vezes chamado de assinaturas de vírus), que permita o reconhecimento de todos os vírus. Se o programa de detecção de vírus não esta atualizado, ele geralmente não detecta um novo vírus. Para detectar vírus, os programas de antivírus comparam conteúdos dos arquivos com as assinaturas de vírus

conhecidas do computador, identificam arquivos infectados e os reparam quando possível ou os apagam, quando não é possível restauração. Programas mais sofisticados também buscam atividades semelhantes a vírus, em uma tentativa de identificar vírus novos e modificados que não seriam reconhecidos pelo banco de dados de detecção de vírus atual. Enquanto não é perfeito, este sistema pode prover uma camada adicional de proteção, com o custo de alguns falsos positivos.

Vírus e outros códigos malignos, tais como *worms* e *trojans,* podem ser enormemente destrutivos para um sistema de computador e a informação da qual ele depende para o sucesso da organização. O mais importante aspecto do programa de detecção de vírus é a freqüente atualização de arquivos de definição de vírus e atualizações necessárias, quando se sabe que um vírus importante esta se espalhando na internet. Quanto mais freqüente for a atualização do banco de dados, mais vírus o programa de detecção de vírus estará equipado a detectar. Se estes passos preliminares forem dados, a chance de uma infecção séria por vírus será minimizada. Os arquivos de detecção de vírus deveriam ser atualizados pelo menos a cada dois meses e sempre que um importante surto de um novo vírus ocorrer. (WACK, 2002)

3.8 Guerra de Discagem *War Dialing*

Em uma rede bem configurada, uma das áreas mais vulneráveis com freqüência negligenciada é a presença de modems não autorizados. Estes modems não autorizados oferecem uma forma de ignorar todas as medidas de segurança designadas para evitar que usuários não autorizados acessem uma rede. Há vários pacotes de programas disponíveis que permitem aos *hackers* e administradores de rede discar grandes blocos de números telefônicos na busca de algum modem disponível. Este processo é chamado de Guerra de Discagem. Um computador com quatro modems pode discar 10.000 números, em questão de dias. Certas guerras de discadores irão até mesmo tentar algum tipo de exploração quando um modem for descoberto. O resultado da aplicação é um relatório sobre os números descobertos com os modems.

A Guerra de Discagem deve ser conduzida pelo menos anualmente e feita depois de horas, para limitar um rompimento potencial para o sistema de telefones dos empregados e da organização (isto é claro tem que ser contrabalanceado com o perigo de que os modems possam ser desligados depois de horas e, portanto, não serão detectados). Isto deve incluir todos os números que pertencem a uma organização, exceto aqueles que poderiam ser de impacto negativo por receberem um grande número de ligações (ex. Centros de Operações 24 horas, números de emergência, etc. A maior parte dos programas de Guerra de Discagem permite que o avaliador isente alguns números da lista).

Se qualquer modem não autorizado for identificado, ele deve ser investigado e removido, se for apropriado. Geralmente, o administrador da troca de ramal privado (PBX - Private Branch of eXchange) deve estar apto a identificar o usuário para quem o número foi atribuído. Se a remoção não é possível, o PBX deve ser configurado para bloquear o numero que chega ao modem. Se as ligações que chegam ao modem são necessárias, deve-se assegurar que um forte método de autenticação seja adotado.

Embora os ataques via internet tenham muita publicidade, muitos ataques de sucesso são feitos através de modems não autorizados. O aumento dos computadores portáteis tem exacerbado este problema, uma vez que a maioria tem um modem. Um simples comprometimento via um modem não autorizado poderia permitir a um transgressor um acesso direto a uma rede, e porque ele evita a segurança de perímetro, é mais provável de passar sem ser detectado. (WACK, 2002)

4 O TESTE DE INTRUSÃO (*PENETRATION TEST*)

De acordo com (CHAN TUCK WAI, 2002): Um "*Penetration Test*" ou seja, um Teste de Intrusão, serve como um recurso para detectar vulnerabilidades de segurança de sistemas ou redes que não possuam uma boa política de segurança.

Um Teste de Intrusão é basicamente uma tentativa de romper a segurança de uma rede ou um sistema. Estes testes são feitos freqüentemente por duas razões: (WACK, 2002).

- Aumentar a consciência da gerência superior acerca das falhas dos seus sistemas e para detecção de intrusão e capacidade de resposta;
- Ajuda também a gerência mais elevada em processos de tomada de decisão. A gerência de uma organização não pode querer dirigir-se a todas as vulnerabilidades encontradas, mas pode querer dirigir-se às fraquezas do sistema que são encontradas através de um teste deste tipo. Isto pode acontecer, pois corrigir todas as fraquezas que são encontradas em uma avaliação pode ser caro, e a maioria das organizações não possui recursos para fazê-lo.

É uma atividade de trabalho intenso e requer grande conhecimento para minimizar o risco dos sistemas alvejados. Geralmente estes testes empregam técnicas normalmente utilizadas pelos *hackers* para a obtenção de informações confidenciais de empresas. Muitas vezes estes "intrusos" são contratados exclusivamente para invadir sistemas de organizações concorrentes, buscando informações sobre novos produtos que estão sendo desenvolvidos, ou mesmo para roubar dados de potenciais clientes.

Dependendo do tipo de teste empregado, pode consistir, simplesmente, de uma varredura de endereços IP para identificar as máquinas que estão oferecendo serviços com vulnerabilidades facilmente reconhecidas.

Após a verificação, os levantamentos obtidos são documentados e apresentados na forma de um relatório ao proprietário do sistema para, posteriormente, as brechas na segurança serem resolvidas.

Estes testes devem ser cuidadosamente empregados, pois se mal aplicados, podem causar parada ou congestionamento do sistema ou exatamente o que tentam impedir. É

vital ter o consentimento da gerência de uma organização antes de conduzir um Teste de Penetração em seus sistemas ou rede. Esta “regra de compromisso” deve incluir: (WACK, 2002)

- Especificação dos endereços IP a serem testados;
- Qualquer computador servidor restringido (ex. computadores servidores, sistemas, subredes, a não ser testadas);
- Uma lista de técnicas de teste aceitáveis (ex.engenharia social, Dos, etc) e ferramentas (decodificadores de senha, “*sniffers*” de rede, etc);
- Momentos que o escaneamento será conduzido (ex. durante hora de trabalho, depois de horário de trabalho, etc);
- Endereço de IP das maquinas nas quais o teste de penetração será conduzido, de forma que os administradores possam diferenciar o legítimo ataque de teste de penetração dos ataques verdadeiros de hackers;
- Pontos de contato, tanto para a equipe de teste de penetração, como para os sistemas alvejados e para as redes;
- Medidas para evitar sanções de leis serem chamadas a termo, com alarmes falsos;
- Lidar com a informação coletada pela equipe do Teste de Penetração.

Pelo fato deste tipo de teste poder ser empregado com o conhecimento ou não da equipe de segurança da organização, ele se torna útil para testar, além da segurança da rede, a resposta da equipe de tecnologia da informação a incidentes de segurança percebidos e seu conhecimento da política de segurança da organização.

Um Teste de Invasão pode ser projetado para simular um ataque interno ou externo. Se ambos os teste internos e externos forem feitos, a testagem externa geralmente ocorre primeiro. Com a testagem de penetração externa, *firewalls* geralmente limitam a quantia, tipo e tráfego que é permitido dentro da rede interna a partir de fontes externas. Dependendo de quais protocolos são permitidos, ataques iniciais são geralmente focados em protocolos de aplicação comumente usados e permitidos tais como FTP e HTTP. Com a testagem externa, as barreiras entre as redes

externas e internas são o que pode aumentar o tempo, dificuldade e custo de realizar um teste externo.

Para simular um verdadeiro ataque externo, os avaliadores não são providos com qualquer informação real sobre o ambiente alvo do que os endereços IP/ variações e devem coletar informações secretamente, antes do ataque. Eles coletam informações em questão de páginas publicadas na *web*, grupos de notícias e afins. Eles então usam portas de *scanners* e *scanners* de vulnerabilidade para identificar computadores servidores alvejados. Uma vez que eles estão, muito provavelmente, se submetendo a um *firewall*, a quantia de informação é muito menor do que eles obteriam se operassem internamente.

Depois de identificar computadores servidores na rede que podem ser alcançados de fora, eles tentam comprometer um. Se tiverem sucesso, eles então alavancam este acesso para comprometer outros computadores servidores, não geralmente acessíveis de fora. Devido a isso, o teste de penetração é um processo iterativo que alavanca acesso mínimo para praticamente ganhar acesso total.

Um teste de penetração interna é similar ao teste externo, exceto devido ao fato de que os avaliadores agora estão na rede interna (e atrás do *firewall*) e lhes é garantido algum nível de acesso para a rede (geralmente como um usuário, mas às vezes, em um nível superior). Os avaliadores de penetração tentarão então ganhar um nível maior de acesso para a rede através de escalação de privilégio.

O teste de penetração consiste de quatro fases, como mostra a figura 5: (WACK, 2002)

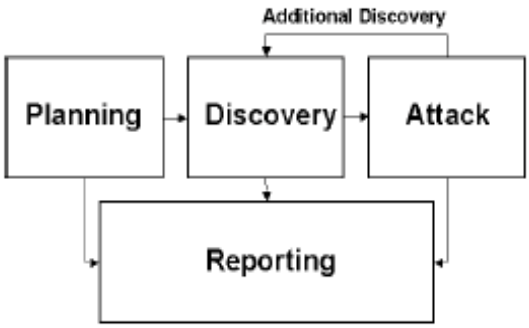

Figura 5 - Fluxograma das fases do teste de penetração

- **Fase de Planejamento (*Planning*):** Nesta fase é obtida a aprovação da gerência, são identificadas as regras e definidos os objetivos;
- **Fase de descoberta (*Discovery*):** Nesta fase é que se inicia o teste real. É feito o mapeamento da rede (escaneamento de portas) e podem ser usadas outras técnicas para reunir informações tais como: (WACK, 2002)
  - Consulta de sistema de nome de domínio;
  - Questionário *InterNIC* (*whois*);
  - Buscar o servidor (es) de web da organização alvo para informação;
  - Buscar o servidor (es) de protocolo leve de acesso a diretório - *Lightweight Directory Access Protocol* (LDAP) para informação;
  - Captura de pacote (geralmente somente durante testes internos);
  - Enumeração de *NetBIOS* (geralmente somente durante testes internos);
  - Captura de *banner.*

  A segunda parte da fase de descoberta consiste na análise das vulnerabilidades levantadas. Durante esta fase, serviços, aplicações e o sistema operacional do servidor alvo são comparados como o BD de vulnerabilidade. Esta etapa é automatizada (no caso dos scanners de vulnerabilidade).

  A desvantagem da análise automatizada é que, para vulnerabilidades novas, é mais eficiente a análise manual, porém este processo é bem mais veloz quando automatizado.
- **Fase de Ataque (*Attack*):** É nesta fase que, se a invasão obtiver êxito, confirma-se à vulnerabilidade do sistema e procedimentos corretivos podem ser empregados.
- **Fase de Relato (*Reporting*):** Fase final do teste. É entregue um relatório à empresa com as vulnerabilidades descobertas.

A maior parte das vulnerabilidades exploradas por testes de penetração e invasores maliciosos encaixa-se nas seguintes categorias: (WACK, 2002)

- Falhas de *kernel:* código de *Kernel* é o núcleo de um sistema operacional. O código de *kernel* reforça o modelo de segurança de todo o sistema. Qualquer falha de segurança que ocorre no *kernel* coloca o sistema inteiro em perigo;

- Estouro de memória intermediaria (*Buffer Overflow*): um estouro de memória intermediaria ocorre quando os programas não checam a entrada de informações adequadamente, para apropriado comprimento, e é geralmente o resultado de prática pobre de programação. Quando isto corre, um código arbitrário pode ser introduzido no sistema e executado com os privilégios da execução do programa. Este código pode, com freqüência, ser executado como *root* em sistemas Unix e SYSTEM (administrador equivalente) em sistemas *windows*;
- Ligação simbólica: uma ligação simbólica ou *symlink* é um arquivo que leva a outro arquivo. Com freqüência há programas que mudarão a permissão de um arquivo. Se estes programas forem executados com permissões privilegiadas, um usuário poderia estrategicamente criar *symlinks* para ludibriar tais programas, ao modificar ou listar arquivos de sistema críticos;
- Ataque de descritor de arquivo: descritores de arquivo são números inteiros não negativos que o sistema usa para rastrear arquivo mais do que para usar nomes específicos de arquivos. Certos descritores de arquivos têm usos subentendidos. Quando um programa privilegiado designa um descritor de arquivo não apropriado ele expõe esse arquivo a se comprometer;
- Condições de disputa (*Race Condition*): condições de disputa podem ocorrer quando um programa ou processo entra em um modo privilegiado. Antes do programa ou processo ter desistido de seu modo privilegiado, um usuário pode marcar um ataque para tomar vantagem deste programa ou processo, enquanto ele ainda esta no modo privilegiado. Se um transgressor consegue comprometer o programa ou processo com sucesso durante seu estado privilegiado, então o transgressor vence a “disputa”. Condições comuns de disputa incluem manejo do sinal e manipulação de arquivo central;
- Permissões de arquivo e diretório: é o controle de permissões de arquivos e diretórios aos quais os usuários e processos têm acesso. Permissões apropriadas são criticas para a segurança de qualquer sistema. Permissões pobres poderiam permitir qualquer número de ataques, incluindo a leitura e

escrita de arquivos de senhas ou adicionar computadores servidores lícitos para conectar ao arquivo de computador servidor;

- *Trojans* – programas de *trojans* podem ser construídos sob encomenda ou podem incluir programas tal como *BackOrifice*, *Netbus* e *Subseven*. *Kits* de *root kernel* poderiam também ser empregados uma vez que o sucesso é obtido de forma a permitir uma porta dos fundos para o sistema, a qualquer momento;
- Engenharia social (como visto anteriormente).

Aconselha-se às organizações praticarem atividades de teste menos abrangentes (como mapeamento de rede e varredura de vulnerabilidades) constantemente, corrigindo deficiências descobertas para, quando forem submetidas a testes de invasão ou ataques reais estarem menos expostas. (WACK, 2002)

5 EXEMPLO PRÁTICO

Aplicação da ferramenta *Essential Net Tools* para levantamento de vulnerabilidades de segurança na Prefeitura Municipal do Rio Grande

De acordo com o descrito anteriormente, um Teste de Invasão possui diversas etapas que envolvem sua aplicação. A primeira etapa envolve a obtenção da permissão do administrador ou responsável pela rede a qual iremos testar, bem como é nesta etapa que serão definidos os objetivos do teste. Geralmente, o objetivo desta aplicação é validar a política de segurança da organização ou contestá-la, quando a invasão é consumada.

A segunda etapa (a qual focaremos neste exemplo) consiste na fase de descoberta, primordial para que este tipo de teste apresente um resultado satisfatório. É nesta fase que se obtêm um conjunto de informações que permitirão ou não que este sistema seja invadido, através da exploração das vulnerabilidades encontradas.

Para a obtenção destas vulnerabilidades, diversas técnicas podem ser utilizadas, entre elas as técnicas de escaneamento de portas e escaneamento de vulnerabilidades, descritas em capítulos anteriores.

Neste exemplo foi utilizada a ferramenta *Essencial Net Tools* para a obtenção de falhas de segurança da rede da Prefeitura Municipal do Rio Grande. Escolheu-se esta ferramenta, pois é de fácil utilização e encontrada facilmente na internet (versão *trial*). Esta ferramenta identifica computadores *windows* com má configuração de *NetBIOS* e compartilhamento de arquivos e impressoras.

Já a interface *NetBIOS* permite que computadores com compartilhamento de arquivos e impressoras ativados, ou outros itens ativados, possam ser alvejados. Este compartilhamento de arquivos e impressoras é umas das vulnerabilidades mais exploradas por *hackers* para acessar documentos particulares de outros computadores por IP (*Internet Protocol*), bastando que estes estejam conectados à internet.

5.1 Utilização do *Essential Net Tools*

O *Essential Net Tools* é um conjunto de ferramentas de rede que serve para diagnosticar e monitorar conexões de rede do computador. A utilização desta ferramenta é bem simples e intuitiva. Sua tela principal é mostrada na figura 6.

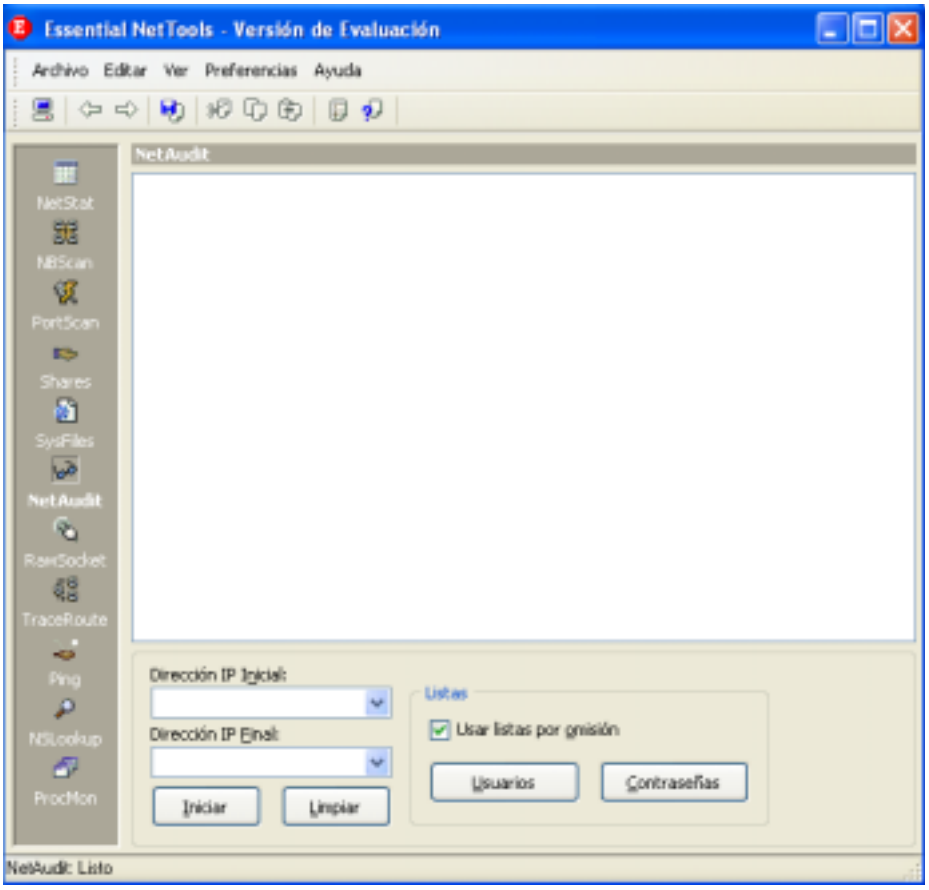

Figura 6- Tela principal do *Essential Net Tools*

Nesta aplicação utilizou-se duas ferramentas do conjunto, o *NBScan* e o *PortScan*. O *NBScan* é um software utilizado para fazer uma varredura de uma série de IPs pré-definidos identificando nestes má configuração de *NetBIOS*, serviços, recursos compartilhados, assim como suas tabelas e endereço do MAC - *Media Access Control* (endereço da placa de rede). O *PortScan* é um escaneador de portas TCP que permite a varredura da rede em busca de portas ativas.

Tentou-se, a partir da listagem dos computadores identificados com má configuração de *NetBIOS* e compartilhamento de arquivos, invadir estas máquinas interna e externamente, isto é, fazer com que a máquina em uso pudesse invadir outra máquina da intranet e esta máquina utilizada, ou qualquer outra da rede, pudesse ser invadida por uma outra estação fora do domínio da rede, através da internet.

5.2 Descrição da Rede da Prefeitura Municipal do Rio Grande

A Prefeitura Municipal do Rio Grande possui cerca de 400 computadores conectados através de fibra ótica (cerca de 10 km de cabo), através de rádio e através de VPN. A conexão da rede que liga a FURG, (Fundação Universidade Federal do Rio Grande) com todos os outros pontos que compõem a rede da Prefeitura, é através de rádio. A ligação da rede do Cassino com a rede do centro da cidade é feita através de VPN.

A transmissão via rádio foi adotada, pois a área onde se encontra situada a FURG é uma região muito pobre, composta de vilas, onde há muita criminalidade, não justificando o uso de fibra ótica, onde cada m^2 de cabo custa cerca de R$ 6,00.

Cada servidor da Prefeitura abastece cerca de 20 computadores. Estes computadores são máquinas obsoletas (a partir de 486 100 MHz com 16 MB de RAM) que são aproveitadas através deste recurso. Os 14 servidores são computadores Athlon 2400 com 1GB (*GigaByte*) de RAM.

Para a segurança da rede contra acesso externo é utilizado um *firewall* no *proxy*.

5.3 Resultados Obtidos no Teste

O domínio da aplicação da ferramenta *PortScan* foi definido com a relação de IPs abrangidos na seqüência de 192.168.0.1 a 192.168.0.255, que são os IPs que as máquinas da Prefeitura possuem, entre outros (a rede possui em torno de 400 computadores). O resultado do escaneamento gerou a relação de portas abertas e serviços ativos nestas máquinas, conforme Anexo A deste documento.

O resultado da utilização da ferramenta *NBScan* acusou, neste intervalo de IPs, as máquinas com má configuração de *NetBIOS* e compartilhamento de arquivos (conforme

Anexo B). O relatório gerado faz referência a todas as máquinas analisadas, porém, para ilustrar, na figura 7, somente será referenciado os dois IPs testados que são 192.168.1.182 e 192.168.1.221.

Reporte: NBScan

Generado en 6/1/2005 a 17:17:56 por Essential NetTools.

MAQUINA182	SMEC	Si	192.168.0.182	00:E0:7D:C3:CE:40	MAQUINA182 <00> UNIQUE SMEC <00> GROUP MAQUINA182 <03> UNIQUE MAQUINA182 <20> UNIQUE SMEC <1E> GROUP
MAQUINA221	PJ	Si	192.168.0.221	00:0D:87:9C:7E:1A	MAQUINA221 <00> UNIQUE MAQUINA221 <20> UNIQUE PJ <00> GROUP PJ <1E> GROUP MAQUINA221 <03> UNIQUE PJ <1D> UNIQUE ..__MSBROWSE__ <01> GROUP

Figura 7 - Relatório gerado no escaneamento dos IPs pelo NBScan

Foram escolhidos para o teste dois IPs com compartilhamento, objetivando invadi-los através de outra máquina da rede (IP 192.168.0.223). O resultado obtido aparece na figura 8.

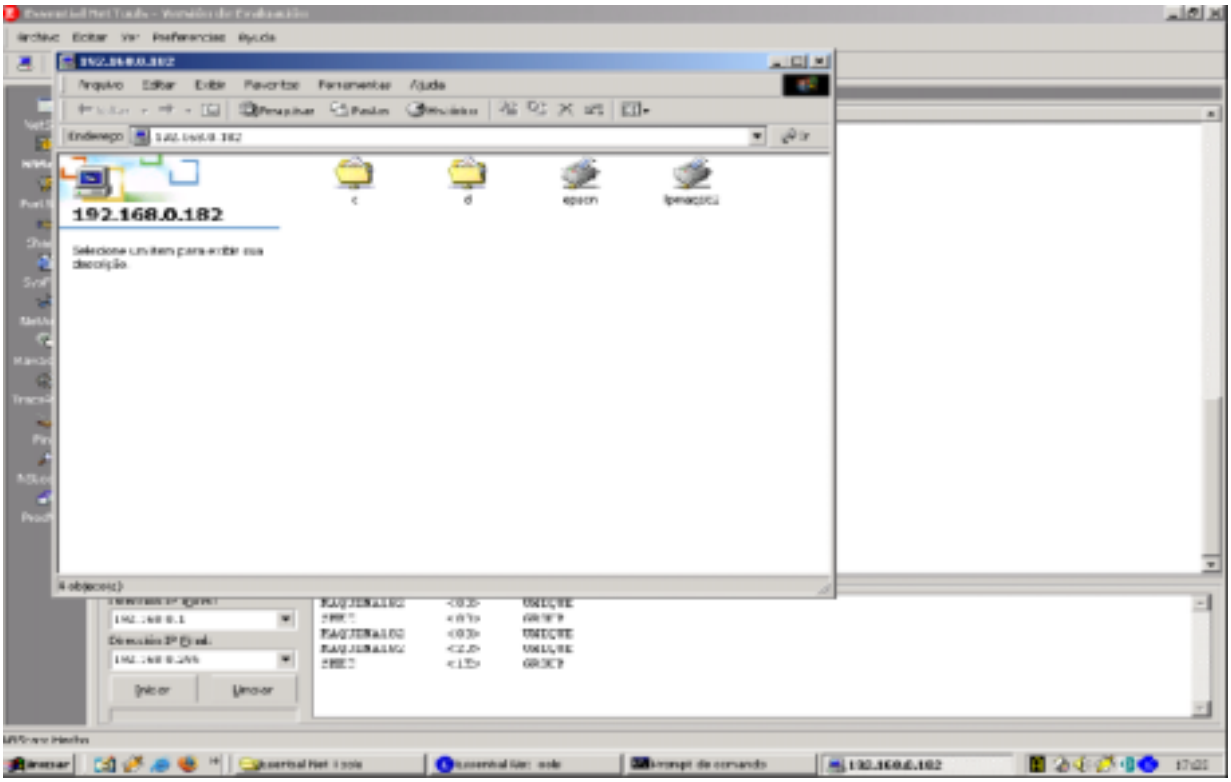

Figura 8 – Invasão a máquina de IP 192.168.0.182 através da Intranet

Esta máquina (IP 192.168.0.182) foi facilmente invadida através da ferramenta *NBScan*, que lista os IPs e as portas abertas destas máquinas. Com a invasão à máquina confirmou-se, como mostra na figura, que esta possui compartilhamento de impressoras, e dos diretórios C e D. Esta falha permite que qualquer computador conectado a rede possa acessar todos os arquivos destes diretórios e utilizar as impressoras compartilhadas.

Outro exemplo foi a exploração do IP 192.168.0.221 que também possui compartilhamento de arquivos e impressoras, porém, ao contrário do IP anterior, estes arquivos só podem ser acessados através de senha. Com isso a invasão a este IP não foi possível naquele momento (conforme mostra a figura 9), pois para que a invasão obtivesse êxito seria necessária a obtenção de acesso através de *ftp* ou *Telnet* para que fosse possível acessar os arquivos *pwl* (arquivos onde ficam armazenadas as senhas no *windows*) para que esta senha pudesse ser quebrada através de um software de quebra de senha ou a aplicação de alguma outra técnica, como engenharia social para descobri-la.

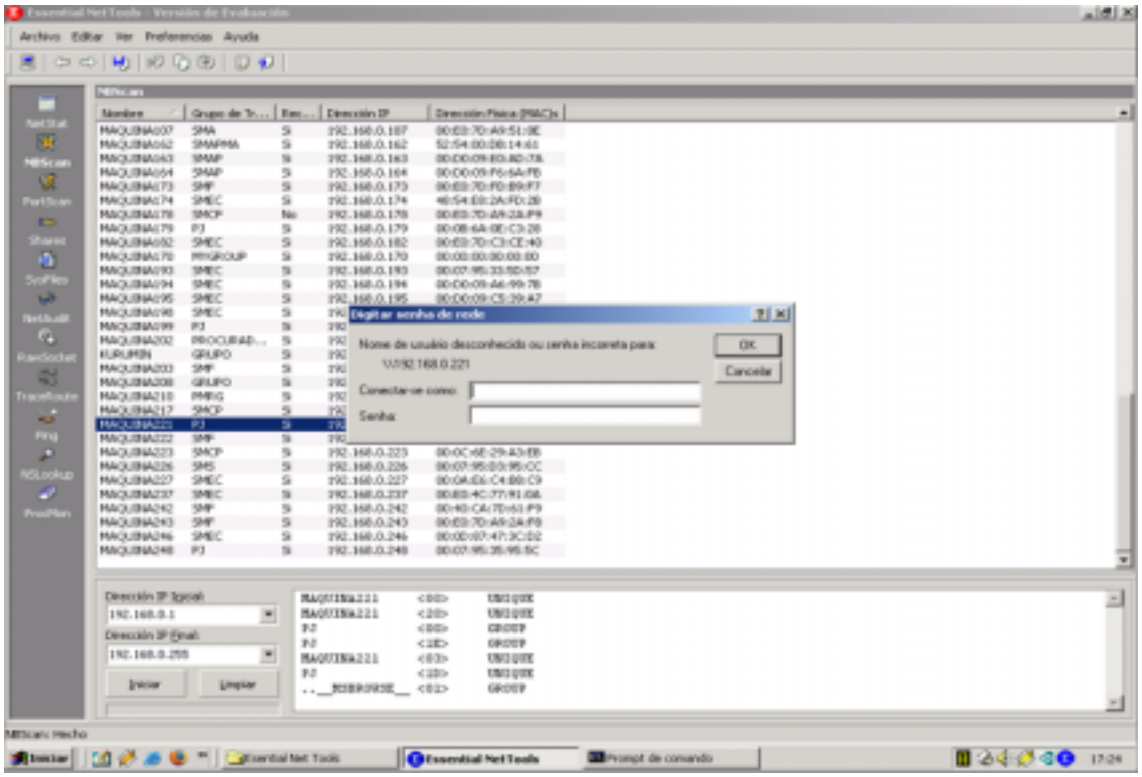

Figura 9 – Tentativa de Invasão à máquina de IP 192.168.0.221 através da Intranet

6 CONCLUSÃO

Conclui-se, com o estudo realizado para o desenvolvimento deste trabalho que, por maior que seja a proteção adotada, um sistema estará sempre sujeito a invasões, roubos e ataques, pois, com a mesma velocidade com que são desenvolvidas ferramentas de proteção contra ameaças virtuais, são criados novos vírus, *worms* e técnicas para exploração de novas falhas de sistema, sistemas estes desenvolvidos sem o número mínimo de testes necessários para serem comercializados com segurança. Esta “pressa” do mercado para sua comercialização torna as aplicações vulneráveis, sendo seus *bugs* o principal alvo de *hackers* em tentativas de invasão.

Outra grande ameaça à segurança da informação é a internet, que facilitou extraordinariamente a comunicação entre empresas e pessoas no mundo inteiro, mas os dados que trafegam livremente nesta grande rede mundial ficaram mais vulneráveis a roubo, acesso e alterações não permitidas. Também é através desta rede que as principais pragas virtuais se propagam (utilizando-se de softwares para troca de mensagens, para *download* de arquivos de música ou mesmo através de *sites* considerados seguros) e é possível invadir computadores através de técnicas relativamente simples, com a invasão por IP. Portanto, restam às empresas e aos usuários domésticos se valerem de todos os recursos de segurança disponíveis atualmente para proteção de seus dados e para amenizar o risco de invasões. Alguns recursos são bem simples de serem implementados, como a configuração adequada dos computadores.

No exemplo prático deste trabalho, que foi aplicado na Prefeitura Municipal do Rio Grande, a utilização da ferramenta *Essential Net Tools* revelou que várias de suas máquinas possuem compartilhamento de informações e recursos e, como esse compartilhamento não é espontâneo, a aplicação desta ferramenta mostrou-se de grande utilidade para a identificação dos microcomputadores que disponibilizam suas informações livremente pela rede, sem o conhecimento do usuário. Por isso, cuidados mais severos devem ser aplicados à rede da Prefeitura internamente, pois as diversas secretarias que a compõem estão vulneráveis a invasões de qualquer máquina

pertencente à rede, pois em diversas delas foi detectada a porta *NetBIOS* aberta, o que permite facilmente uma invasão.

Como nos planos futuros da Prefeitura está um projeto que visa a inclusão digital de alunos das escolas municipais, esta rede será, brevemente, utilizada por qualquer aluno de trinta e três escolas da cidade de Rio Grande. Portanto, esta rede não será mais somente utilizada por funcionários, o que justifica uma maior atenção com relação à segurança de suas informações e, como projeto futuro, a correção das vulnerabilidades encontradas através desta aplicação.

A tentativa de invasão externa não foi possível de ser realizada devido ao grau de dificuldade, visto que a Prefeitura possui alguns recursos de segurança, como a utilização de *firewall* no *proxy*, que dificulta a invasão fora do domínio da rede. Seria necessário um tempo maior que o disponível e a utilização de várias técnicas de invasão e obtenção de informações paralelamente, para que a tentativa fosse válida e obtivesse êxito.

REFERÊNCIAS

ANÔNIMO. Segurança Máxima. Traduzido por: Edson Furmankiewicz. 3.ed. Rio de Janeiro: Campus, 2001.

AUTOSCAN – Exames de Vulnerabilidades
Disponível em: <http: // www.autoscan.com.br>
Acesso em: 16 de dezembro de 2004.

Boletins de Segurança – *Spoofing* do IP
Disponível em:
<http://www.geocities.com/siliconvalley/network/1493/security/fa/fa00001.htm>
Acesso em: 5 de dezembro de 2004.

CAMPELLO, Rafael Saldanha, WEBER, Raul Fernando. Sistemas de Detecção de Intrusão.
Disponível em: <http://www.inf.ufrgs.br/~gseg/producao/minicurso-ids-sbrc-2001.pdf>
Acesso em: 10 de janeiro de 2005.

CERQUEIRA, Eduardo. Auditoria e Segurança
Disponível em: <http://www.inf.ufsc.br/~eduardoc/aula/AulaIX_AS.ppt>
Acesso em: 08 de fevereiro de 2005.

CHAN TUCK WAI**.** *Conducting a penetration test on an organization. SANS Institute, 2002.*
Disponível em: <http://www.sans.org/rr/whitepapers/auditing/67.php>
Acesso em: 8 de outubro de 2004.

DANDREA, Marcelo M. Ferramentas para Segurança na Internet. Porto Alegre: Instituto de Informática, PPGC/ UFRGS,1999.

DIAS, Cláudia. Segurança e Auditoria da Tecnologia da Informação. Rio de Janeiro: Axcel Books, 2000.

FREIRE, Alexandre – iMasters Segurança
Disponível em: <http: //www.imasters.com.br>
Acesso em: 27 de dezembro de 2004.

LIMA JR, Ary Vaz de. *Bugs* e *patches* ou brechas e remendos
Disponível em: <http://www.jseg.net/ed96/informatica_96.htm>
Acesso em: 18 de dezembro de 2004.

LINUX
Disponível em: <http:// olinux.uol.com.br/artigos/263/print_preview.html>
Acesso em: 28 de dezembro de 2004.

MARTINS, José Carlos Cordeiro. Gestão de Projetos de Segurança da Informação. de Janeiro: Brasport, 2003.

McCLURE, Stuart.SCAMBRAY, Joel. *Beware of the obvius: ubiquitous SNMP provides a back door to your network secrets. February 1,1999. 01/01/99.*
Disponível em: <http:// www.infoworld.com/cgi-bin/displayNew.pl?/security/990201sw.htm>
Acesso em: 15 de outubro de 2004.

NAKAMURA, Emílio Tissato; GEUS, Paulo Lício de. Segurança de Redes em Ambientes Cooperativos. 2.ed. – São Paulo: Futura, 2003.

NETSEC – Assessoria em Informática Ltda.
Disponível em: <http://www.netsec.com.br/tecnologia/tecnicas_ataque.htm>
Acesso em: 4 de dezembro de 2004.

NUNES, Giovanni dos Reis – *Sniffers*
Disponível em : <http://www.ufrnet.br/~rk/seguranca/docs/sniffers.txt>
Acesso em: 5 de dezembro de 2004.

PELEGRINA, J. A, ZAMBRANA, Alberto. DicWeb
Disponível em: <http://www.dicweb.com/ww.htm>
Acesso em: 17 de dezembro de 2004.

REZENDE, Edmar Roberto Santana de. Segurança e Vulnerabilidades de Redes
Disponível em:
<http://www.las.ic.unicamp.br/~edmar/Palestras/FAGOC/Seguranca_Vulnerabilidades.pdf>
Acesso em: 08 de fevereiro de 2005.

ROCHA, Cláudio. InformaBR – Administração e Segurança da Informação
Disponível em: <http://www.informabr.com.br/index.htm>
Acesso em: 04 de janeiro de 2005.

SILVA, Fernando. *Network Security*
Disponível em: **<**http://paginas.fe.up.pt/~mgi98020/pgr/DoS.htm>
Acesso em: 12 de dezembro de 2004.

STARLIN, Gorki, NOVO, Rafael. Segurança Completa contra Hackers. Rio de Janeiro: Ed. Book Express, 2000.

WACK John, MILES Tracey. *DRAFT Guideline on Network Security Testing*. *Recommendations of the National Institute of Standards and Technology: Special Publication 800-42 (Feb. 2002).*

WEBER, Raul F. Segurança na Internet. Porto Alegre: Instituto de Informática, UF 2000.

ANEXO A

Relatório Gerado pelo *PortScan* após a varredura dos IPs analisados

Reporte: PortScan

Generado en 6/1/2005 a 17:56:04 por Essential NetTools.

Dirección IP	Puertos Abiertos	Nro. De Puertos Cerrados	Nro. De Puertos Silenciosos
192.168.0.1	ftp;telnet;smtp;http;netbios-ssn;https	26	0
192.168.0.2	sunrpc;3128	30	0
192.168.0.3		32	0
192.168.0.4		0	32
192.168.0.5		0	32
192.168.0.6		0	32
192.168.0.7	netbios-ssn	31	0
192.168.0.8		32	0
192.168.0.9		32	0
192.168.0.10		0	32
192.168.0.11	netbios-ssn	31	0
192.168.0.12		0	32
192.168.0.13		0	32
192.168.0.14		32	0
192.168.0.15		0	32
192.168.0.16		0	32
192.168.0.17		32	0
192.168.0.18		0	32
192.168.0.19	netbios-ssn	31	0
192.168.0.20		0	32
192.168.0.21	netbios-ssn	31	0
192.168.0.22	epmap;netbios-ssn;microsoft-ds	29	0
192.168.0.23		0	32
192.168.0.24	netbios-ssn	31	0
192.168.0.25	http;sunrpc;8080	29	0
192.168.0.26		0	32
192.168.0.27		14	18
192.168.0.28		0	32

192.168.0.29	netbios-ssn	31	0
192.168.0.30		0	32
192.168.0.31	epmap;netbios-ssn	30	0
192.168.0.32		0	32
192.168.0.33		0	32
192.168.0.34		0	32
192.168.0.35		0	32
192.168.0.36		32	0
192.168.0.37	sunrpc	31	0
192.168.0.38		0	32
192.168.0.39	netbios-ssn	31	0
192.168.0.40		0	32
192.168.0.41	netbios-ssn	31	0
192.168.0.42	epmap;netbios-ssn;microsoft-ds	29	0
192.168.0.43		32	0
192.168.0.44		0	32
192.168.0.45		0	32
192.168.0.46	netbios-ssn	31	0
192.168.0.47		32	0
192.168.0.48	netbios-ssn	31	0
192.168.0.49	netbios-ssn	31	0
192.168.0.50		0	32
192.168.0.51		32	0
192.168.0.52		32	0
192.168.0.53		32	0
192.168.0.54	netbios-ssn	31	0
192.168.0.55	netbios-ssn	31	0
192.168.0.56		0	32
192.168.0.57	epmap;netbios-ssn;microsoft-ds	29	0
192.168.0.58	sunrpc	31	0
192.168.0.59		0	32
192.168.0.60	epmap;netbios-ssn;microsoft-ds	29	0
192.168.0.61		0	32
192.168.0.62	netbios-ssn	31	0

192.168.0.63		0	32
192.168.0.64	netbios-ssn	31	0
192.168.0.65	epmap;netbios-ssn;microsoft-ds	29	0
192.168.0.66		0	32
192.168.0.67	netbios-ssn	31	0
192.168.0.68		32	0
192.168.0.69	sunrpc	31	0
192.168.0.70	netbios-ssn	31	0
192.168.0.71		0	32
192.168.0.72		0	32
192.168.0.73		0	32
192.168.0.74	epmap;netbios-ssn;1080	29	0
192.168.0.75		0	32
192.168.0.76		0	32
192.168.0.77		32	0
192.168.0.78		0	32
192.168.0.79		0	32
192.168.0.80	netbios-ssn	30	1
192.168.0.81	epmap;netbios-ssn;microsoft-ds	29	0
192.168.0.82		0	32
192.168.0.83		32	0
192.168.0.84	netbios-ssn	31	0
192.168.0.85		0	32
192.168.0.86	netbios-ssn	31	0
192.168.0.87		0	32
192.168.0.88	epmap;netbios-ssn;microsoft-ds	29	0
192.168.0.89		32	0
192.168.0.90		0	32
192.168.0.91		0	32
192.168.0.92	epmap;netbios-ssn;microsoft-ds	29	0
192.168.0.93	netbios-ssn	31	0
192.168.0.94		0	32
192.168.0.95	sunrpc	31	0
192.168.0.96	epmap;netbios-ssn	30	0

192.168.0.97		0	32
192.168.0.98		0	32
192.168.0.99		0	32
192.168.0.100	sunrpc	31	0
192.168.0.101	epmap;netbios-ssn;microsoft-ds	29	0
192.168.0.102		0	32
192.168.0.103		0	32
192.168.0.104		0	32
192.168.0.105	ftp;http	30	0
192.168.0.106	netbios-ssn	31	0
192.168.0.107	epmap;netbios-ssn	30	0
192.168.0.108		0	32
192.168.0.109	netbios-ssn	31	0
192.168.0.110		0	32
192.168.0.111		5	27
192.168.0.112		0	32
192.168.0.113		0	32
192.168.0.114		32	0
192.168.0.115	sunrpc	31	0
192.168.0.116		0	32
192.168.0.117		32	0
192.168.0.118	netbios-ssn	31	0
192.168.0.119	netbios-ssn	31	0
192.168.0.120		0	32
192.168.0.121	sunrpc	31	0
192.168.0.122		0	32
192.168.0.123		0	32
192.168.0.124		32	0
192.168.0.125		32	0
192.168.0.126		0	32
192.168.0.127	ftp	29	2
192.168.0.128	sunrpc	31	0
192.168.0.129		32	0
192.168.0.130	netbios-ssn	31	0

192.168.0.131		0	32
192.168.0.132	sunrpc	31	0
192.168.0.133	netbios-ssn	31	0
192.168.0.134	netbios-ssn	31	0
192.168.0.135	netbios-ssn	31	0
192.168.0.136	epmap;netbios-ssn;microsoft-ds	29	0
192.168.0.137		32	0
192.168.0.138		0	32
192.168.0.139	netbios-ssn	31	0
192.168.0.140		0	32
192.168.0.141		0	32
192.168.0.142	epmap;netbios-ssn	30	0
192.168.0.143		0	32
192.168.0.144		0	32
192.168.0.145	netbios-ssn	31	0
192.168.0.146	epmap;netbios-ssn;microsoft-ds	29	0
192.168.0.147		0	32
192.168.0.148	epmap;netbios-ssn;microsoft-ds	29	0
192.168.0.149		0	32
192.168.0.150		0	32
192.168.0.151		0	32
192.168.0.152		0	32
192.168.0.153		0	32
192.168.0.154		0	32
192.168.0.155		32	0
192.168.0.156	epmap;netbios-ssn;microsoft-ds	29	0
192.168.0.157		0	32
192.168.0.158		0	32
192.168.0.159		0	32
192.168.0.160	sunrpc	31	0
192.168.0.161		0	32
192.168.0.162	netbios-ssn	31	0
192.168.0.163	epmap;netbios-ssn	30	0
192.168.0.164	netbios-ssn	20	11

192.168.0.165		0	32
192.168.0.166		0	32
192.168.0.167		0	32
192.168.0.168		30	2
192.168.0.169		32	0
192.168.0.170	echo;ftp;telnet;netbios-ssn	28	0
192.168.0.171	http;sunrpc;https	29	0
192.168.0.172		0	32
192.168.0.173	netbios-ssn	31	0
192.168.0.174	netbios-ssn	31	0
192.168.0.175	sunrpc	31	0
192.168.0.176		0	32
192.168.0.177	sunrpc	31	0
192.168.0.178	netbios-ssn	31	0
192.168.0.179	netbios-ssn	31	0
192.168.0.180	telnet;http	30	0
192.168.0.181		0	32
192.168.0.182	netbios-ssn	31	0
192.168.0.183		0	32
192.168.0.184		0	32
192.168.0.185		32	0
192.168.0.186		0	32
192.168.0.187		32	0
192.168.0.188		32	0
192.168.0.189		0	32
192.168.0.190		0	32
192.168.0.191		0	32
192.168.0.192		32	0
192.168.0.193	http;netbios-ssn	30	0
192.168.0.194	epmap;netbios-ssn;microsoft-ds	29	0
192.168.0.195	epmap;netbios-ssn;microsoft-ds	29	0
192.168.0.196		0	32
192.168.0.197		0	32
192.168.0.198	http;epmap;netbios-ssn;microsoft-ds;1080	27	0

192.168.0.199	netbios-ssn	31	0
192.168.0.200	telnet;http	30	0
192.168.0.201		0	32
192.168.0.202	netbios-ssn	31	0
192.168.0.203	sunrpc;netbios-ssn	30	0
192.168.0.204		0	32
192.168.0.205	epmap;netbios-ssn;microsoft-ds	29	0
192.168.0.206		0	32
192.168.0.207		0	32
192.168.0.208	epmap;netbios-ssn;microsoft-ds	29	0
192.168.0.209		0	32
192.168.0.210	netbios-ssn	21	10
192.168.0.211		32	0
192.168.0.212		0	32
192.168.0.213	sunrpc	31	0
192.168.0.214		0	32
192.168.0.215		0	32
192.168.0.216		0	32
192.168.0.217	epmap;netbios-ssn	30	0
192.168.0.218		0	32
192.168.0.219		32	0
192.168.0.220		32	0
192.168.0.221	epmap;netbios-ssn;microsoft-ds	29	0
192.168.0.222	epmap;netbios-ssn;microsoft-ds	29	0
192.168.0.223	Dirección local - omitida		
192.168.0.224		0	32
192.168.0.225		0	32
192.168.0.226	netbios-ssn	31	0
192.168.0.227	epmap;netbios-ssn;microsoft-ds	29	0
192.168.0.228		0	32
192.168.0.229		32	0
192.168.0.230		0	32
192.168.0.231		32	0
192.168.0.232		32	0

192.168.0.233	smtp;domain;http;sunrpc	28	0
192.168.0.234		0	32
192.168.0.235		32	0
192.168.0.236	http;sunrpc	30	0
192.168.0.237	epmap;netbios-ssn;microsoft-ds	29	0
192.168.0.238		32	0
192.168.0.239		0	32
192.168.0.240		0	32
192.168.0.241		0	32
192.168.0.242	epmap;netbios-ssn;microsoft-ds	29	0
192.168.0.243	netbios-ssn	31	0
192.168.0.244	ftp;http	30	0
192.168.0.245	sunrpc	31	0
192.168.0.246	epmap;netbios-ssn;microsoft-ds	29	0
192.168.0.247		0	32
192.168.0.248	netbios-ssn	31	0
192.168.0.249		0	32
192.168.0.250		32	0
192.168.0.251		0	32
192.168.0.252		0	32
192.168.0.253		32	0
192.168.0.254		0	32
192.168.0.255		0	32

ANEXO B

Relatório Gerado pelo *NBScan* após a varredura dos IPs analisados

Reporte: NBScan

Generado en 6/1/2005 a 17:17:56 por Essential NetTools.

Nombre	Grupo de Trabajo	Recursos Compartidos	Dirección IP	Dirección Física (MAC)s	Nombre de Tabla
PMRG1	SMCP	Si	192.168.0.1	00:00:00:00:00:00	PMRG1 <00> UNIQUE PMRG1 <03> UNIQUE PMRG1 <20> UNIQUE ..__MSBROWSE__ <01> GROUP SMCP <00> GROUP SMCP <1D> UNIQUE SMCP <1E> GROUP
MAQUINA7	SMCP	No	192.168.0.7	00:0C:6E:29:A3:9B	MAQUINA7 <00> UNIQUE SMCP <00> GROUP MAQUINA7 <03> UNIQUE DILMA <03> UNIQUE
MAQUINA11	PJ	Si	192.168.0.11	00:0D:87:9C:7D:09	MAQUINA11 <00> UNIQUE PJ <00> GROUP MAQUINA11 <03> UNIQUE MAQUINA11 <20> UNIQUE PJ

					<1E> GROUP
MAQUINA19	SMCP	Si	192.168.0.19	00:E0:7D:C3:CE:3A	MAQUINA19 <00> UNIQUE SMCP <00> GROUP MAQUINA19 <03> UNIQUE MAQUINA19 <20> UNIQUE SMCP <1E> GROUP
MAQUINA21	PJ	Si	192.168.0.21	00:D0:09:D8:EA:17	MAQUINA21 <00> UNIQUE PJ <00> GROUP MAQUINA21 <03> UNIQUE MAQUINA21 <20> UNIQUE PJ <1E> GROUP JURIDICO <03> UNIQUE
MAQUINA22	SMCP	Si	192.168.0.22	00:E0:7D:A9:49:0C	MAQUINA22 <00> UNIQUE SMCP <00> GROUP MAQUINA22 <03> UNIQUE MAQUINA22 <20> UNIQUE SMCP <1E> GROUP
MAQUINA24	SMF	Si	192.168.0.24	00:00:B4:98:85:87	MAQUINA24 <00> UNIQUE SMF <00> GROUP MAQUINA24 <03> UNIQUE MAQUINA24 <20> UNIQUE SMF <1E> GROUP
MAQUINA26	SMF	Si	192.168.0.26	00:07:95:33:A1:94	MAQUINA26 <00> UNIQUE SMF <00> GROUP MAQUINA26 <20> UNIQUE SMF <1E> GROUP

MAQUINA29	SMF	Si	192.168.0.29	00:07:95:E7:7B:27	MAQUINA29 <00> UNIQUE SMF <00> GROUP MAQUINA29 <03> UNIQUE MAQUINA29 <20> UNIQUE SMF <1E> GROUP
MAQUINA33	SMF	Si	192.168.0.33	00:0A:E6:49:65:D8	MAQUINA33 <00> UNIQUE SMF <00> GROUP MAQUINA33 <03> UNIQUE MAQUINA33 <20> UNIQUE SMF <1E> GROUP LÚCIA <03> UNIQUE
MAQUINA35	PMRG	Si	192.168.0.35	00:08:54:0E:3C:1F	MAQUINA35 <00> UNIQUE PMRG <00> GROUP MAQUINA35 <03> UNIQUE MAQUINA35 <20> UNIQUE PMRG <1E> GROUP
MAQUINA39	SMF	Si	192.168.0.39	00:02:96:01:27:AB	MAQUINA39 <00> UNIQUE SMF <00> GROUP MAQUINA39 <03> UNIQUE MAQUINA39 <20> UNIQUE SMF <1E> GROUP
MAQUINA41	SMF	Si	192.168.0.41	00:E0:7D:FB:3A:4F	MAQUINA41 <00> UNIQUE SMF <00> GROUP MAQUINA41 <03> UNIQUE MAQUINA41 <20> UNIQUE SMF <1E> GROUP

MAQUINA42	SMF	Si	192.168.0.42	00:0D:87:07:79:2C	MAQUINA42 <00> UNIQUE SMF <00> GROUP MAQUINA42 <20> UNIQUE SMF <1E> GROUP SMF <1D> UNIQUE ..__MSBROWSE__ <01> GROUP
MAQUINA46	SMF	Si	192.168.0.46	48:54:E8:2B:00:98	MAQUINA46 <00> UNIQUE SMF <00> GROUP MAQUINA46 <03> UNIQUE MAQUINA46 <20> UNIQUE SMF <1E> GROUP
MAQUINA49	SMCP	Si	192.168.0.49	00:E0:7D:ED:E6:63	MAQUINA49 <00> UNIQUE SMCP <00> GROUP MAQUINA49 <03> UNIQUE MAQUINA49 <20> UNIQUE SMCP <1E> GROUP
MAQUINA48	SMF	Si	192.168.0.48	00:E0:7D:C3:CE:44	MAQUINA48 <00> UNIQUE SMF <00> GROUP MAQUINA48 <03> UNIQUE MAQUINA48 <20> UNIQUE SMF <1E> GROUP
MAQUINA31	GABEX	Si	192.168.0.31	00:0A:E6:CF:FE:36	MAQUINA31 <00> UNIQUE GABEX <00> GROUP MAQUINA31 <03> UNIQUE MAQUINA31 <20> UNIQUE GABEX <1E> GROUP

MAQUINA54	SMF	Si	192.168.0.54	00:80:AD:12:16:DB	MAQUINA54 <00> UNIQUE SMF <00> GROUP MAQUINA54 <03> UNIQUE MAQUINA54 <20> UNIQUE SMF <1E> GROUP
MAQUINA55	SMF	Si	192.168.0.55	00:08:54:12:73:EA	MAQUINA55 <00> UNIQUE SMF <00> GROUP MAQUINA55 <03> UNIQUE MAQUINA55 <20> UNIQUE SMF <1E> GROUP
MAQUINA57	SMF	Si	192.168.0.57	00:E0:7D:A9:49:0F	MAQUINA57 <00> UNIQUE SMF <00> GROUP MAQUINA57 <20> UNIQUE MAQUINA57 <03> UNIQUE SMF <1E> GROUP
MAQUINA59	SMF	Si	192.168.0.59	00:07:95:35:8E:21	MAQUINA59 <00> UNIQUE SMF <00> GROUP MAQUINA59 <20> UNIQUE MAQUINA59 <03> UNIQUE SMF <1E> GROUP
MAQUINA60	SMF	Si	192.168.0.60	00:0A:E6:DF:AE:50	MAQUINA60 <00> UNIQUE SMF <00> GROUP MAQUINA60 <03> UNIQUE MAQUINA60 <20> UNIQUE SMF <1E> GROUP REGINA <03> UNIQUE

MAQUINA62	SMF	Si	192.168.0.62	00:E0:4C:7B:5E:7E	MAQUINA62 <00> UNIQUE SMF <00> GROUP MAQUINA62 <03> UNIQUE MAQUINA62 <20> UNIQUE SMF <1E> GROUP SEPRIM <03> UNIQUE
MAQUINA64	PJ	Si	192.168.0.64	00:07:95:35:AF:54	MAQUINA64 <00> UNIQUE PJ <00> GROUP MAQUINA64 <03> UNIQUE MAQUINA64 <20> UNIQUE PJ <1E> GROUP MAQUINA64 <1F> UNIQUE
MAQUINA65	SMF	Si	192.168.0.65	00:08:54:0E:31:A7	MAQUINA65 <00> UNIQUE SMF <00> GROUP MAQUINA65 <20> UNIQUE SMF <1E> GROUP
MAQUINA66	SMF	Si	192.168.0.66	00:0A:E6:E8:7D:73	MAQUINA66 <00> UNIQUE SMF <00> GROUP MAQUINA66 <03> UNIQUE MAQUINA66 <20> UNIQUE SMF <1E> GROUP MAQUINA26 <03> UNIQUE
MAQUINA67	PMRG	Si	192.168.0.67	52:54:00:E0:49:33	MAQUINA67 <00> UNIQUE PMRG <00> GROUP MAQUINA67 <03> UNIQUE MAQUINA67 <20> UNIQUE PMRG <1E>

					GROUP USUARIO PC <03> UNIQUE
MAQUINA70	SMF	Si	192.168.0.70	00:80:AD:42:9D:C3	MAQUINA70 <00> UNIQUE SMF <00> GROUP MAQUINA70 <03> UNIQUE MAQUINA70 <20> UNIQUE SMF <1E> GROUP
MAQUINA74	SMAPMA	Si	192.168.0.74	00:00:00:00:00:00	MAQUINA74 <00> UNIQUE SMAPMA <00> GROUP MAQUINA74 <03> UNIQUE MAQUINA74 <20> UNIQUE SMAPMA <1E> GROUP SMAPMA <1D> UNIQUE ..__MSBROWSE__ <01> GROUP
MAQUINA80	SMF	No	192.168.0.80	00:00:B4:5F:19:E2	MAQUINA80 <00> UNIQUE SMF <00> GROUP MAQUINA80 <03> UNIQUE
MAQUINA81	SMF	Si	192.168.0.81	00:E0:7D:A7:1F:E8	MAQUINA81 <00> UNIQUE SMF <00> GROUP MAQUINA81 <20> UNIQUE MAQUINA81 <03> UNIQUE SMF <1E> GROUP
MAQUINA86	SMA	Si	192.168.0.86	00:0A:E6:43:83:08	MAQUINA86 <00> UNIQUE SMA <00> GROUP MAQUINA86 <03> UNIQUE MAQUINA86 <20> UNIQUE SMA <1E>

					GROUP LENIRA <03> UNIQUE
MAQUINA88	SMA	Si	192.168.0.88	00:0B:CD:BB:8C:61	MAQUINA88 <00> UNIQUE SMA <00> GROUP MAQUINA88 <03> UNIQUE MAQUINA88 <20> UNIQUE SMA <1E> GROUP SMA <1D> UNIQUE ..__MSBROWSE__ <01> GROUP
MAQUINA92	GABEX	Si	192.168.0.92	00:E0:06:09:55:66	MAQUINA92 <00> UNIQUE GABEX <00> GROUP MAQUINA92 <03> UNIQUE MAQUINA92 <20> UNIQUE GABEX <1E> GROUP GABEX <1D> UNIQUE ..__MSBROWSE__ <01> GROUP
MAQUINA93	SMA	No	192.168.0.93	00:07:95:35:A4:23	MAQUINA93 <00> UNIQUE SMA <00> GROUP MAQUINA93 <03> UNIQUE PMRG <03> UNIQUE
MAQUINA96	PJ	Si	192.168.0.96	00:00:21:4C:4C:49	MAQUINA96 <00> UNIQUE PJ <00> GROUP MAQUINA96 <03> UNIQUE MAQUINA96 <20> UNIQUE PJ <1E> GROUP
MAQUINA101	GRUPO	Si	192.168.0.101	00:E0:06:09:55:66	MAQUINA101 <00> UNIQUE GRUPO <00>

					GROUP MAQUINA101 <20> UNIQUE MAQUINA101 <03> UNIQUE GRUPO <1E> GROUP USUARIO <03> UNIQUE
MAQUINA106	GABEX	Si	192.168.0.106	00:0D:87:A6:BC:E6	MAQUINA106 <00> UNIQUE GABEX <00> GROUP MAQUINA106 <03> UNIQUE MAQUINA106 <20> UNIQUE GABEX <1E> GROUP MAQUINA126 <03> UNIQUE
MAQUINA109	SMA	Si	192.168.0.109	00:0C:6E:8C:FE:16	MAQUINA109 <00> UNIQUE SMA <00> GROUP MAQUINA109 <03> UNIQUE MAQUINA109 <20> UNIQUE SMA <1E> GROUP
MAQUINA111	SMA	Si	192.168.0.111	00:50:FC:0A:1A:BD	MAQUINA111 <00> UNIQUE SMA <00> GROUP MAQUINA111 <03> UNIQUE MAQUINA111 <20> UNIQUE SMA <1E> GROUP
MAQUINA118	PMRG	Si	192.168.0.118	00:E0:7D:A9:2E:BE	MAQUINA118 <00> UNIQUE PMRG <00> GROUP MAQUINA118 <03> UNIQUE MAQUINA118 <20> UNIQUE PMRG <1E> GROUP
MAQUINA119	PMRG	Si	192.168.0.119	00:30:4F:0A:CF:74	MAQUINA119 <00>

					UNIQUE PMRG <00> GROUP MAQUINA119 <03> UNIQUE MAQUINA119 <20> UNIQUE PMRG <1E> GROUP PMRG <1D> UNIQUE ..__MSBROWSE__ <01> GROUP
MAQUINA130	PJ	Si	192.168.0.130	00:04:AC:BF:1B:FA	MAQUINA130 <00> UNIQUE PJ <00> GROUP MAQUINA130 <03> UNIQUE MAQUINA130 <20> UNIQUE PJ <1E> GROUP
MAQUINA131	PJ	Si	192.168.0.131	00:D0:09:A9:76:5D	MAQUINA131 <00> UNIQUE PJ <00> GROUP MAQUINA131 <03> UNIQUE MAQUINA131 <20> UNIQUE PJ <1E> GROUP
MAQUINA133	PJ	Si	192.168.0.133	00:07:95:10:39:96	MAQUINA133 <00> UNIQUE PJ <00> GROUP MAQUINA133 <03> UNIQUE MAQUINA133 <20> UNIQUE PJ <1E> GROUP
MAQUINA134	PJ	Si	192.168.0.134	00:07:95:35:DC:8B	MAQUINA134 <00> UNIQUE PJ <00> GROUP MAQUINA134 <03> UNIQUE MAQUINA134 <20> UNIQUE PJ <1E> GROUP

MAQUINA135	PJ	Si	192.168.0.135	00:07:95:F9:EB:E6	MAQUINA135 <00> UNIQUE PJ <00> GROUP MAQUINA135 <03> UNIQUE MAQUINA135 <20> UNIQUE PJ <1E> GROUP PROCURADORIA <03> UNIQUE
MAQUINA136	SMCP	Si	192.168.0.136	00:0B:6A:5E:29:9C	MAQUINA136 <00> UNIQUE SMCP <00> GROUP MAQUINA136 <03> UNIQUE MAQUINA136 <20> UNIQUE SMCP <1E> GROUP SUZEL <03> UNIQUE
MAQUINA139	SMEC	Si	192.168.0.139	00:0B:6A:0B:95:1D	MAQUINA139 <00> UNIQUE SMEC <00> GROUP MAQUINA139 <03> UNIQUE MAQUINA139 <20> UNIQUE SMEC <1E> GROUP ROSILETESMEC <03> UNIQUE
MAQUINA145	SMCP	Si	192.168.0.145	00:E0:7D:C3:CE:2D	MAQUINA145 <00> UNIQUE SMCP <00> GROUP MAQUINA145 <03> UNIQUE MAQUINA145 <20> UNIQUE SMCP <1E> GROUP
MAQUINABALAS	SMCP	Si	192.168.0.146	00:0D:87:8E:5C:B4	MAQUINABALAS <00> UNIQUE SMCP <00> GROUP MAQUINABALAS <03> UNIQUE MAQUINABALAS <20>

					UNIQUE SMCP <1E> GROUP
MAQUINA148	SMCP	Si	192.168.0.148	E0:20:03:30:0A:DA	MAQUINA148 <00> UNIQUE SMCP <00> GROUP MAQUINA148 <03> UNIQUE MAQUINA148 <20> UNIQUE SMCP <1E> GROUP FABIANO <03> UNIQUE
MAQUINA107	SMA	Si	192.168.0.107	00:E0:7D:A9:51:0E	MAQUINA107 <00> UNIQUE SMA <00> GROUP MAQUINA107 <03> UNIQUE MAQUINA107 <20> UNIQUE SMA <1E> GROUP
MAQUINA162	SMAPMA	Si	192.168.0.162	52:54:00:DB:14:61	MAQUINA162 <00> UNIQUE SMAPMA <00> GROUP MAQUINA162 <03> UNIQUE MAQUINA162 <20> UNIQUE SMAPMA <1E> GROUP
MAQUINA163	SMAP	Si	192.168.0.163	00:D0:09:E0:AD:7A	MAQUINA163 <00> UNIQUE SMAP <00> GROUP MAQUINA163 <03> UNIQUE MAQUINA163 <20> UNIQUE SMAP <1E> GROUP SMAP <1D> UNIQUE ..__MSBROWSE__ <01> GROUP
MAQUINA164	SMAP	Si	192.168.0.164	00:D0:09:F6:6A:FB	MAQUINA164 <00> UNIQUE SMAP <00>

					GROUP MAQUINA164 <03> UNIQUE MAQUINA164 <20> UNIQUE SMAP <1E> GROUP
MAQUINA173	SMF	Si	192.168.0.173	00:E0:7D:FD:B9:F7	MAQUINA173 <00> UNIQUE SMF <00> GROUP MAQUINA173 <03> UNIQUE MAQUINA173 <20> UNIQUE SMF <1E> GROUP
MAQUINA174	SMEC	Si	192.168.0.174	48:54:E8:2A:FD:2B	MAQUINA174 <00> UNIQUE SMEC <00> GROUP MAQUINA174 <03> UNIQUE MAQUINA174 <20> UNIQUE SMEC <1E> GROUP IEDAFREITAS <03> UNIQUE
MAQUINA178	SMCP	No	192.168.0.178	00:E0:7D:A9:2A:F9	MAQUINA178 <00> UNIQUE SMCP <00> GROUP MAQUINA178 <03> UNIQUE
MAQUINA179	PJ	Si	192.168.0.179	00:0B:6A:0E:C3:28	MAQUINA179 <00> UNIQUE PJ <00> GROUP MAQUINA179 <03> UNIQUE MAQUINA179 <20> UNIQUE PJ <1E> GROUP
MAQUINA182	SMEC	Si	192.168.0.182	00:E0:7D:C3:CE:40	MAQUINA182 <00> UNIQUE SMEC <00> GROUP MAQUINA182 <03> UNIQUE MAQUINA182 <20>

					UNIQUE SMEC <1E> GROUP
MAQUINA170	MYGROUP	Si	192.168.0.170	00:00:00:00:00:00	MAQUINA170 <00> UNIQUE MAQUINA170 <03> UNIQUE MAQUINA170 <20> UNIQUE ..__MSBROWSE__ <01> GROUP MYGROUP <00> GROUP MYGROUP <1E> GROUP
MAQUINA193	SMEC	Si	192.168.0.193	00:07:95:33:5D:57	MAQUINA193 <00> UNIQUE SMEC <00> GROUP MAQUINA193 <03> UNIQUE MAQUINA193 <20> UNIQUE SMEC <1E> GROUP
MAQUINA194	SMEC	Si	192.168.0.194	00:D0:09:A6:99:7B	MAQUINA194 <00> UNIQUE SMEC <00> GROUP MAQUINA194 <03> UNIQUE MAQUINA194 <20> UNIQUE SMEC <1E> GROUP SMEC <1D> UNIQUE ..__MSBROWSE__ <01> GROUP
MAQUINA195	SMEC	Si	192.168.0.195	00:D0:09:C5:39:A7	MAQUINA195 <00> UNIQUE MAQUINA195 <20> UNIQUE SMEC <00> GROUP SMEC <1E> GROUP
MAQUINA198	SMEC	Si	192.168.0.198	00:07:95:34:16:C3	MAQUINA198 <00> UNIQUE MAQUINA198 <20> UNIQUE SME <00>

					GROUP SMEC <1E> GROUP
MAQUINA199	PJ	Si	192.168.0.199	00:0B:6A:0E:C0:1C	MAQUINA199 <00> UNIQUE PJ <00> GROUP MAQUINA199 <03> UNIQUE MAQUINA199 <20> UNIQUE PJ <1E> GROUP
MAQUINA202	PROCURADORIA	Si	192.168.0.202	00:0B:6A:0E:C0:21	MAQUINA202 <00> UNIQUE PROCURADORIA <00> GROUP MAQUINA202 <03> UNIQUE MAQUINA202 <20> UNIQUE PROCURADORIA <1E> GROUP PROCURADORIA <1D> UNIQUE ..__MSBROWSE__ <01> GROUP
KURUMIN	GRUPO	Si	192.168.0.203	00:00:00:00:00:00	KURUMIN <00> UNIQUE KURUMIN <03> UNIQUE KURUMIN <20> UNIQUE ..__MSBROWSE__ <01> GROUP GRUPO <00> GROUP GRUPO <1D> UNIQUE GRUPO <1E> GROUP
MAQUINA203	SMF	Si	192.168.0.204	00:0D:87:06:EC:5D	MAQUINA203 <00> UNIQUE SMF <00> GROUP MAQUINA203 <03> UNIQUE MAQUINA203 <20> UNIQUE SMF <1E> GROUP FLAVINHA <03> UNIQUE

MAQUINA208	GRUPO	Si	192.168.0.208	00:0B:CD:32:A6:02	MAQUINA208 <00> UNIQUE MAQUINA208 <03> UNIQUE MAQUINA2081 <03> UNIQUE GRUPO <00> GROUP GRUPO <1E> GROUP MAQUINA208 <20> UNIQUE
MAQUINA210	PMRG	Si	192.168.0.210	00:08:54:08:CB:D3	MAQUINA210 <00> UNIQUE PMRG <00> GROUP MAQUINA210 <03> UNIQUE MAQUINA210 <20> UNIQUE PMRG <1E> GROUP
MAQUINA217	SMCP	Si	192.168.0.217	00:07:95:37:10:A5	MAQUINA217 <00> UNIQUE SMCP <00> GROUP MAQUINA217 <03> UNIQUE MAQUINA217 <20> UNIQUE SMCP <1E> GROUP
MAQUINA221	PJ	Si	192.168.0.221	00:0D:87:9C:7E:1A	MAQUINA221 <00> UNIQUE MAQUINA221 <20> UNIQUE PJ <00> GROUP PJ <1E> GROUP MAQUINA221 <03> UNIQUE PJ <1D> UNIQUE ..__MSBROWSE__ <01> GROUP
MAQUINA222	SMF	Si	192.168.0.222	00:E0:7D:F5:99:A5	MAQUINA222 <00> UNIQUE SMF <00> GROUP MAQUINA222 <03> UNIQUE MAQUINA222 <20>

					UNIQUE SMF <1E> GROUP ADMINISTRADOR <03> UNIQUE
MAQUINA223	SMCP	Si	192.168.0.223	00:0C:6E:29:A3:EB	MAQUINA223 <00> UNIQUE SMCP <00> GROUP MAQUINA223 <20> UNIQUE SMCP <1E> GROUP
MAQUINA226	SMS	Si	192.168.0.226	00:07:95:D3:95:CC	MAQUINA226 <00> UNIQUE SMS <00> GROUP MAQUINA226 <03> UNIQUE MAQUINA226 <20> UNIQUE SMS <1E> GROUP SMS <1D> UNIQUE ..__MSBROWSE__ <01> GROUP
MAQUINA227	SMEC	Si	192.168.0.227	00:0A:E6:C4:B8:C9	MAQUINA227 <00> UNIQUE MAQUINA227 <20> UNIQUE SMEC <00> GROUP SMEC <1E> GROUP
MAQUINA237	SMEC	Si	192.168.0.237	00:E0:4C:77:91:0A	MAQUINA237 <00> UNIQUE SMEC <00> GROUP MAQUINA237 <03> UNIQUE MAQUINA237 <20> UNIQUE SMEC <1E> GROUP
MAQUINA242	SMF	Si	192.168.0.242	00:40:CA:7D:61:F9	MAQUINA242 <00> UNIQUE SMF <00> GROUP MAQUINA242 <20> UNIQUE SMF <1E>

					GROUP
MAQUINA243	SMF	Si	192.168.0.243	00:E0:7D:A9:2A:F8	MAQUINA243 <00> UNIQUE SMF <00> GROUP MAQUINA243 <03> UNIQUE MAQUINA243 <20> UNIQUE SMF <1E> GROUP
MAQUINA246	SMEC	Si	192.168.0.246	00:0D:87:47:3C:D2	MAQUINA246 <00> UNIQUE MAQUINA246 <20> UNIQUE SMEC <00> GROUP SMEC <1E> GROUP
MAQUINA248	PJ	Si	192.168.0.248	00:07:95:35:95:5C	MAQUINA248 <00> UNIQUE PJ <00> GROUP MAQUINA248 <03> UNIQUE MAQUINA248 <20> UNIQUE PJ <1E> GROUP

ANEXO C

Autorização concedida pela Prefeitura Municipal do Rio Grande para divulgação dos resultados do teste