OUTROS TRABALHOS EM:
www.projetoderedes.com.br

CENTRO UNIVERSITÁRIO RITTER DOS REIS

FACULDADE DE SISTEMAS DE INFORMAÇÃO

TECNOLOGIAS PARA IMPLANTAÇÃO DE SEGURANÇA DE INFORMAÇÃO EM REDES DE PEQUENAS E MÉDIAS EMPRESAS ATRAVÉS DE SOFTWARE LIVRE

Monografia

Juarez Dubois da Costa

Porto Alegre, 2005

CENTRO UNIVERSITÁRIO RITTER DOS REIS

FACULDADE DE SISTEMAS DE INFORMAÇÃO

JUAREZ DUBOIS DA COSTA

TECNOLOGIAS PARA IMPLANTAÇÃO DE SEGURANÇA DE INFORMAÇÃO EM REDES DE PEQUENAS E MÉDIAS EMPRESAS ATRAVÉS DE SOFTWARE LIVRE

Monografia apresentada ao Curso de Sistemas de Informação, como requisito do Trabalho de Conclusão 2 sob Orientação do Professor Dr. Luís Fernando Fortes Garcia.

Porto Alegre
2005

SUMÁRIO

LISTA DE FIGURAS 6
INTRODUÇÃO 7
1 SERVIÇOS UTILIZADOS NAS PMEs 9
1.1 Serviços utilizados em pequenas e microempresas 9
1.1.1 – Empresa administrativa 10
1.1.1.2 – Software 10
1.1.1.3 – Topologia de rede 11
1.1.2 – Empresa prestadora de serviços 13
1.1.2.2 – Software 13
1.1.2.3 – Topologia de rede 14
1.2 – Casos de médias empresas 14
1.2.1 – Infraestrutura de rede das médias empresas 16
1.2.1.1 – As necessidades de uma rede de perímetro 16
1.2.1.2 – Outras necessidades 18
2 PROBLEMAS DE SEGURANÇA 19
2.1 – TCP/IP 20
2.1.1 – Portas abertas 21
2.2 – Acesso remoto e transferências de arquivos 22
2.2.1 – Dados em trânsito 23
2.3 – Tráfego http 24
2.3.1 – Riscos de acesso http 25
2.3.2 – Riscos do servidor http 26
2.4 – Malwares e vírus 27
3 PROPOSTA DE SOLUÇÃO 28
3.1 - Descrição da rede de testes 28
3.1.1 – Objetivos 29
3.1.2 - Topologia de redes 30

3.1.3 - Computadores utilizados *31*
3.1.4 - Hardware *32*
3.1.5 - Sistemas operacionais utilizados *33*
3.1.5.1 - Serviços utilizados 34
3.1.5.1.2 - Serviço de DHCP 35
3.1.5.1.3 - mysql 36
3.1.5.1.4 - Apache 36
3.1.5.2 - Softwares instalados 37
3.1.5.2.1 - snort 37
3.1.5.2.2 - nessus 38
3.1.5.2.3 – ACID 39
3.1.5.2.4 - squid 39
3.1.5.2.5 - clamAV 39
3.2 – Verificação de vulnerabilidades na rede 40
3.2.1 – Verificação de vulnerabilidades com o Nessus *40*
3.2.1.1 – Vericações realizadas nos sistemas 44
3.2.1.1.1 – Verirficação de 15/11/2005 44
3.2.1.1.2 – Verirficação de 22/11/2005 47
3.2.1.1.2 – Verirficação de 10/12/2005 47
3.3 – Intervenção nos sistemas 48
3.3.1 – Prevenindo as portas TCP *49*
3.3.1.1 – Finalizando os serviços desnecessários 49
3.3.1.1.1 - Interrompendo os serviços no Slackware 50
3.3.1.2 – Filtragem de datagramas da rede: 58
3.3.1.2.1 – O comando iptables 58
3.3.1.2.2 – Funcionamento do iptables 59
3.3.2 – Alternativas seguras à transferência de arquivos e acesso remoto *64*
3.3.3 – Prevenção de tráfego http *66*
3.3.3.1 – Segurança do servidor web Apache 66
3.3.3.1.1 – Servidores web externos 66
3.3.3.2 – Implementando serviço proxy com squid 68
3.4 – Utilização do software antivírusclamAV 68

CONCLUSÃO 71
REFERÊNCIAS BIBLIOGRÁFICAS 72
APÊNDICES 73
APÊNDICE A - Configuração do mysql 73
APÊNDICE B - Instalação dos arquivos do *ACID* no *apache*.....76
APÊNDICE C - Controle de acesso no apache 78
APÊNDICE D – Instalação e configuração do *snort* 84
APÊNDICE E – Instalação e configuração do *nessus* 91
APÊNDICE F – Instalação e configuração do proxy-cache *squid* 97
APÊNDICE G – Utilização do Open SSH.....106
APÊNDICE H – Instalação do antivírus *clamAV* 115
ANEXOS 117
ANEXO A – Relatório geral do *nessus* no dia 15 de novembro de 2005. 117
ANEXO B – Relatório do nessus para o host de bastião no endereço IP 192.168.1.1 no dia 15 de novembro de 2005. 120
ANEXO C – Relatório do nessus para o servidor no endereço IP 192.168.1.4 no dia 15 de novembro de 2005. 127
ANEXO D – Relatório do *nessus* para o cliente no endereço IP 192.168.1.3 no dia 15 de novembro de 2005. 131
ANEXO E – Relatório geral do *nessus* no dia 22 de novembro de 2005. 139
ANEXO F – Relatório do *nessus* para o host de bastião no endereço IP 10.1.1.24 no dia 22 de novembro de 2005. 142
ANEXO G – Relatório do *nessus* para o host de bastião no endereço IP 192.168.1.1 no dia 22 de novembro de 2005. 145
ANEXO H – Relatório do *nessus* para o servidor no endereço IP 192.168.1.4 no dia 22 de novembro de 2005. 149
ANEXO I – Relatório geral do *nessus* no dia 10 de dezembro de 2005.153
ANEXO J – Relatório do *nessus* para o host de bastião no endereço IP 192.168.1.1 no dia 10 de dezembro de 2005.....156
ANEXO K – Relatório do *nessus* para o servidor no endereço IP 192.168.1.4 no dia 10 de dezembro de 2005.....161

LISTA DE FIGURAS

Figura 1 - Topologia de rede da empresa administrativa.....11

Figura 2– Topologia de rede da rede de testes.....30

Figura 3 - Janela de login do cliente nessus.....42

Figura 4 - Janela de plug-ins do cliente nessus.....43

INTRODUÇÃO

O objetivo deste trabalho é implementar e avaliar o nível de segurança oferecido por alguns dos serviços de rede, baseados na tecnologia de software livre, mais utilizados nas empresas de pequeno e médio porte (PME), com a finalidade de ser um guia de referência de implementação e configuração segura destas tecnologias com um enfoque didático e voltado à minimização de vulnerabilidades nos sistemas e ferramentas de rede e de segurança administrativa de rede.

O público alvo do trabalho são administradores de redes de empresas de pequeno porte ou ainda aos prestadores de serviço às empresas de pequeno porte que dispõem de poucos recursos financeiros para treinamento, tendo o aspecto educacional na questão de minimizar a vulnerabilidade nos sistemas de informação de tais empresas através da utilização de técnicas e ferramentas, baseadas na tecnologia de software livre, disponibilizados na internet pela comunidade de desenvolvedores da de software livre existente hoje.

O trabalho baseou-se em casos de empresas de pequeno porte e nas suas necessidades, avaliando também as necessidades de empresas de porte médio. Apesar de o porte das empresas mudar as suas necessidades de recursos de Tecnologia da Informação (TI) aumentam apenas em termos de volume de dados e de processamento, sendo os recursos de software e as

ferramentas utilizadas muito semelhantes, podendo ser avaliadas da mesma forma para cada tipo de empresa.

O trabalho envolve o estudo de casos de empresas de pequeno e médio porte envolvendo as suas necessidades de TI oferecidas por sistemas e ferramentas baseados na tecnologia de software livre a fim de implementar estas ferramentas em uma rede de testes com o objetivo de avaliar o nível de segurança oferecido por estas ferramentas.

Através da verificação de vulnerabilidades conhecidas com a utilização de ferramentas de detecção de vulnerabilidades baseadas em software livre nos serviços disponíveis, avaliou-se o nível de segurança e, caso necessário, propôs-se alternativas aos riscos apresentados. A ferramenta de verificação utilizada possui uma base de dados de informações sobre vulnerabilidades que pode ser periodicamente atualizada. Ela contém a avaliação de níveis de segurança, segundo a equipe que a desenvolve, que pesquisa as vulnerabilidades e propõe alternativas de solução.

Na primeira parte do trabalho avaliamos as necessidades e os serviços de rede baseados na tecnologia de software livre utilizados nas PMEs através da observação dos serviços os quais as empresas se utilizam. Na segunda parte desta pesquisa avaliamos alguns dos problemas que estão relacionados aos serviços de rede baseados em software livre utilizados nas PMEs Na última parte do trabalho implementou-se uma rede de computadores e avaliou-se o nível de segurança, segundo uma determinada ferramenta de verificação de vulnerabilidades, e propôs-se alternativas de configuração e implementação dos serviços de rede baseados em software livre.

1 SERVIÇOS UTILIZADOS NAS PMEs

Aqui apresentamos casos de empresas que utilizam sistemas baseados em software livre. Cada empresa está inclusa em uma categoria específica de porte empresarial para ilustrar a variabilidade de configurações encontradas nas PME. Os sistemas utilizados são baseados em GNU/Linux, diferindo na distribuição, na configuração e na aplicação.

Os sistemas aqui descritos são baseados em empresas reais onde algumas vezes fui chamado para administrar. Omitiu-se aqui o nome real das empresas para proteger seu patrimônio digital, financeiro, moral e também devido à ética profissional.

1.1 Serviços utilizados em pequenas e microempresas

Apresentamos nesta seção dois casos de empresas de porte de microempresa e de pequena empresa, sendo uma empresa do ramo administrativo e outra voltada à prestação de serviços.

A empresa prestadora de serviços é um exemplo de microempresa e a empresa administrativa é um exemplo de pequena empresa.

1.1.1 – Empresa administrativa

O modelo utilizado neste exemplo se baseou em uma empresa de estacionamentos que se utiliza de um sistema *GNU/Linux* para gerir e guardar os seus dados.

A solicitação feita ao orientando eram administrativo de gerenciamento de rede *Windows* e seu *backup*, tendo reestruturado conforme o planejado pelo diretor de informática conforme as necessidades da presidência regional da empresa em relação aos usuários da rede.

1.1.1.2 – Software

O sistema servidor da empresa se baseia na distribuição *Conectiva Linux 8*, utilizando-se do sistema de arquivos *reiserFS*, serviço de compartilhamento de recursos de rede Windows baseados em *CIFS/SMB* e uma base de dados *Interbase 6* para *Linux*, além de se utilizar um filtro de pacotes TCP/IP com o *iptables*, realizando a filtragem e seleção através da ferramenta *iptables* e mais um *cache* de *http* com o *squid*.

O *backup* de dados é realizado nos arquivos da base de dados e arquivos dos funcionários relativos ao trabalho toda a sexta-feira por um script automatizado que reúne as pastas do sistema a serem guardadas criando-se

um arquivo de *imagem* de CD do tipo *ISO9660* e gravando em um *CD-RW* na sexta-feira.

As estações são baseadas em *windows 98* compartilhando pastas na rede, variando conforme a necessidade de cada usuário e setor.

1.1.1.3 – Topologia de rede

A topologia de rede utilizada no sistema da empresa de estacionamento é do tipo estrela, tendo um modem *ADSL* para acesso à internet com um *link* de 256 Kbits/s, o servidor atuando como filtro de datagramas IP entre a internet e a rede local e um *Switch* para distribuir o sinal de rede entre os computadores. A figura a seguir ilustra a topologia de rede da empresa de administrativa.

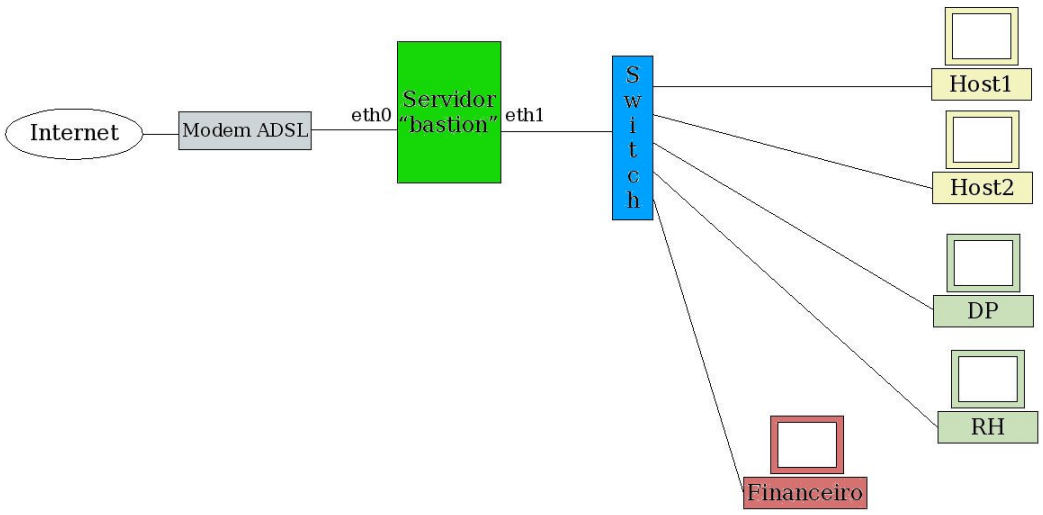


Figura 1: Topologia de rede da empresa administrativa.

A figura acima mostra a internet, à extrema esquerda, com um modem do tipo *ADSL* chegando até a primeira interface ethernet do servidor (*eth0*) usando a pilha de protocolos TCP/IP e repassando a conexão de internet para um switch na rede interna através da segunda interface ethernet (*eth1*) utilizando também a pilha de protocolos do TCP/IP.

O endereçamento do modem até a interface de rede eth0 é do tipo *classe A* sendo o modem com endereço IP de valor 10.1.1.1 e a interface *eth0* recebendo endereço IP através do serviço *DHCP* provido pelo modem de endereço IP de valor 10.1.1.5.

O endereçamento da rede interna é do tipo *classe C*, com o servidor *DHCP* servindo endereçamento IP dinâmico através de um servidor *DHCP* configurado especificamente para os computadores da rede interna.

O servidor tem algumas funcionalidades básicas como a filtragem de datagramas TCP/IP, filtragem de tráfego *http* através de um serviço de *proxy* utilizando o software *squid*, repositório de arquivos da rede utilizando o servidor *Samba*, e serviço de endereçamento IP dinâmico (bastante estático) através de um servidor *DHCP*, além do servidor de banco de dados *Interbase 6*.

A figura 1 ilustra o modelo padrão de topologia de rede utilizada pelas empresas de pequeno porte, sendo que nas empresas de porte médio normalmente envolvem um ou dois servidores a mais e mais especializados, contendo ainda uma *Zona Desmilitarizada* (DMZ) ou ainda conhecida por *rede de perímetro*, uma maior segmentação de redes de acordo com critérios de necessidade, etc. sendo o modelo acima apresentado fortemente desaconselhado por ZWICKY (2000, p. 144) pelo fato de o host de bastião ser a única froteira entre a rede interna e a rede externa no caso a internet,

localizada após o modem *ADSL*.

A rede existente entre o modem e o computador *firewall* poderia ser considerada uma *rede de perímetro* ou *DMZ*, mas tendo o computador *firewall* entre o segmento interno da rede e o segmento de perímetro (*DMZ*) é considerado um perigo por estar o bastião com o tráfego da rede interna visível dependendo da cofiguração da rede, se é comutada ou chaveada tendo como principal preocupação a exposição do tráfego da rede interna para o *firewall* no caso de comprometimento do *firewall* por um invasor.

1.1.2 – Empresa prestadora de serviços

O estudo envolve uma empresa prestadora de serviços com as mesmas necessidades básicas da empresa de estacionamentos descrita anteriormente, utilizando compartilhamento de arquivos de rede *Windows* para armazenamento de uma base de dados *Paradox*, e filtragem de datagramas da pilha TCP/IP.

1.1.2.2 – Software

O sistema servidor é constituído por um sistema operacional *Red Hat Linux* versão 9 executando os serviços de filtragem de datagramas da pilha de

protocolos TCP/IP com o software *iptables*, serviço de *cache http* com o software *squid* e serviço de compartilhamento de arquivos e autenticação de redes *windows* utilizando o servidor *samba*.

1.1.2.3 – Topologia de rede

A topologia de rede da empresa de prestadora de serviços é baseada na mesma topologia de rede da empresa administrativa, mudando apenas a forma de endereçamento do protocolo IP das estações que é feita manualmente nas máquinas, mas continuando a ter endereçamento do tipo *classe C*. Outra mudança significativa é a utilização do banco de dados *Paradox*, que em comparação a outros bancos de dados que utilizam conexões TCP/IP para a conexão remota este utiliza-se do compartilhamento da rede *Windows* para ser armazenado em arquivos no servidor sob uma pasta compartilhada específica.

1.2 – Casos de médias empresas

As médias empresas são um pouco mais complexas devido a necessidades maiores, como mais de um servidor, por exemplo. Em muitos casos há a necessidade de máquinas maiores e mais especializadas, além de um número de redes maior do que o das pequenas empresas, incluindo,

algumas vezes, mais de um ponto de entrada e saída da rede de perímetro, exigindo mais da administração de rede.

Estas empresas exigem uma hierarquização maior em termos de controle, *backups*, políticas, configurações além de um aumento na carga de trabalho do administrador no próprio monitoramento.

Cada sistema fornecedor de algum tipo de serviço da rede deve ser tratado da mesma forma de instalação e configuração em termos de segurança do que o computador *firewall*, também conhecido como *host de bastião* ou *bastion host*, como ZWICKY (2000, p.247) descreve no capítulo sobre o *host de bastião*:

> Os princípios e procedimentos para construir um host de bastião são extensões dos princípios para proteger qualquer host. Você deverá usá-los, ou variações deles, para qualquer outro host que seja importante para a segurança, e possivelmente para hosts que são críticos em outros aspectos (por exemplo, servidores importantes na sua rede interna).

Conforme a citação acima as demais máquinas servidoras de serviços na rede devem como base as medidas adotadas na instalação e configuração de um host de bastião, devendo ser razoavelmente bem protegidas. Adotamos uma abordagem semelhante em nossa rede de testes descrita no *Capítulo 4*.

Os motivos que levam à adoção de mais de uma máquina servidora nas médias empresas se deve pela necessidade de maiores capacidades de tráfego de dados, fornecimento de serviços com muitos usuários e até mesmo mais serviços como serviços de resolução de nomes (*DNS*), configuração dinâmica de hosts (*DHCP*), etc. A adoção de múltiplas máquinas se dá por

melhor desempenho e/ou maiores capacidades, redundância ou com relação ao nível de segurança oferecida pelo tipo de serviço de rede necessário como comentado por ZWICKY (2000, p.140).

1.2.1 – Infraestrutura de rede das médias empresas

A infraestrutura de rede das médias empresas normalmente inclui uma rede de perímetro (*DMZ*) sendo filtrada nas pontas e onde mais for possível, como nas possíveis subredes existentes.

Quanto maior o porte da empresa maiores são as suas necessidades em termos de recursos de TI, tanto em quantidade quanto em capacidade e redundância, e em se falando de médias empresas já não temos mais uma serviço de acesso à internet, uma base de dados e um compartilhamento de arquivos, este tipo de empresa normalmente tem um site próprio hospedado na rede da empresa, um servidor de correio eletrônico, servidores de *DNS*, servidores *DHCP*, etc.

ZWICKY (2000, p. 140) propõe diversos tipos de particionamento dos serviços entre as máquinas servidoras, sendo que as mais práticas resultam no particionamento dos serviços pelo nível de segurança oferecido pelos tipos de serviços e por desempenho que se baseou nossa abordagem.

1.2.1.1 – <u>As necessidades de uma rede de perímetro</u>

A *rede de perímetro*, ou ainda z*ona desmilitarizada*, é o segmento mais próximo da rede externa (internet) e está no topo da hierarquia da rede, sendo as demais redes subredes desta ou outras rede independentes. Segundo ZWICKY (2000, p. 132):

> A rede de perímetro é uma rede adicional entre a rede externa e a rede interna protegida. Se um atacante invade com sucesso os limites mais externos de seu firewall, a rede de perímetro oferece uma camada adicional de segurança entre esse atacante e seus sistemas internos.

Assim sendo, o tráfego existente entre dois hosts na rede mais interna não pode ser visualizado na rede de perímetro, gerando um nível a mais de segurança. Os sistemas servidores que provem serviços considerados inseguros devem ser colocados na rede de perímetro e suas comunicações devem ser de e para o host de bastião externo apenas, como um servidor *web* externo.

No caso de serviços que necessitam de comunicação de ambos os lados, como serviços de e-mail local, deve-se controlar este tipo de comunicação com a filtragem de rede e certificar-se de se estar o mais seguro possível com relação aos clientes na rede do segmento dos usuários e sistemas internos.

O segmento existente entre o modem ADSL e o host de bastião (*firewall*) pode ser considerada uma rede de perímetro, onde as máquinas que provem serviços inseguros e/ou públicos devem ser colocadas.

1.2.1.2 – Outras necessidades

Empresas de porte médio podem envolver processos de comunicação menos comuns como, por exemplo, a utilização de *redes privadas virtuais* (*VPNs*) para a criação de túneis criptografados para a comunicação de redes através de redes inseguras como a internet.

Estes tipos de funcionalidades devem estar previstos nas necessidades da empresa e ser cuidadosa e devidamente implementadas com foco na segurança a ser oferecida para e pelo serviço.

Cada tipo de necessidade em termos de serviços de rede deve ser cuidadosamente avaliada para que se possa ter certeza dos riscos, vulnerabilidades, exposição e todos os possíveis aspectos relativos aos serviços necessários e, caso o serviço seja aprovado, a implementação deve ter um enfoque fortemente embasado na segurança do serviço, do servidor, dos dados e dos usuários do serviço.

Não abordaremos outros serviços que não são implementados na rede de testes devido ao escopo do trabalho ser voltado aos sistemas utilizados nos casos das empresas estudadas e o tempo de pesquisa e implementação ser limitado.

Conhecendo então os serviços utilizados pelas PMEs buscou-se na literatura as vulnerabilidades e problemas conhecidos para propor alternativas durante a implementação de uma rede. No próximo capítulo comentaremos a respeito de problemas e vulnerabilidades encontrados na literatura a respeito dos serviços utilizados.

2 PROBLEMAS DE SEGURANÇA

Neste capítulo descrevemos os principais problemas relativos à segurança da informação e dos sistemas relacionados à informação de pequenas e médias empresas.

Os problemas de segurança a serem enfrentados por qualquer tipo e porte de empresa são normalmente os mesmos, já que os serviços e sistemas utilizados não variam tanto como o porte da empresa e sim com as suas necessidade, sendo normalmente os mesmos como, por exemplo, um servidor *web*, um filtro de datagramas IP, um servidor *proxy*, um sistema de banco de dados, etc.

Os problemas geralmente são associados a esquemas de segurança mal definidos (ou muitas vezes não definidos), resultando, normalmente, em vazamento de informações por falha de implementação e/ou falha na configuração dos sistemas que geram, tratam e/ou transferem a informação.

As regras de segurança a serem definidas depende da dependência da empresa com relação à tecnologia da informação, variando conforme a área de atuação e necessidades em relação aos recursos de TI.

O vazamento das informações são geralmente associados a usuários mal intencionados com relação à empresa, sejam funcionários ou atacantes de empresas concorrentes que estejam interessados em tais informações. Algumas vezes é possível que ocorra acidentes devido à falta de atenção de

alguns administradores novatos em sistemas livres e a perda e o vazamento de informações não é raro de se acontecer.

A seguir apresentamos alguns problemas relacionados à segurança da informação relativos a alguns aspectos relevantes de sistemas baseados em tecnologias de software livre utilizados por muitas empresas e que comumente são origens de exposições de informações, do sistema ou de ambos, podendo gerar um quadro péssimo dentro da estrutura empresarial.

Vamos apresentar as topologias possíveis e as possíveis vulnerabilidades apresentadas, aplicando algumas sugestões e conceitos de gestão segurança apresentadas na literatura voltadas à proteção destes sistemas.

2.1 – TCP/IP

Na questão relativa à segurança de rede existe um problema característico das comunicações baseadas no conjunto de protocolos da família TCP/IP, especificamente no que tange ao protocolo TCP ou UDP, conhecido por abertura de portas.

O protocolo TCP realiza todo o controle de conexão durante a transferência de dados em uma rede IP e para isto utiliza-se de um conceito chamado de *porta* que define um valor determinado de uma determinada *porta* de um determinado tipo de serviço de rede para que haja comunicação controlada entre o cliente e o servidor (ou serviço oferecido).

No caso de servidores de rede estes necessitam de uma ou mais portas

TCP (ou UDP no caso de conexões não controladas) que necessitam estar disponíveis e esperando (ou *escutando*) por solicitações dos clientes. Já nas estações de trabalho (ou clientes de rede) existe a necessidade por manter determinadas portas abertas para que possa haver comunicação entre o computador (ou software) cliente e o servidor (ou aplicação servidora).

O problema ocorre que normalmente em ambientes de software livre estas configurações fogem ao escopo padrão de administradores inexperientes ou novos em sistemas *Unix* (ou *Posix*), ficando com os administradores de rede e de sistemas o papel de configuração das estações e dos servidores.

De modo semelhante ao problema encontrado nas permissões de usuários a falta de uma definição da utilização da rede e das premissas de segurança as configurações são mal realizadas, deixando de lado muitos serviços de rede ainda ativos nos sistemas (e suas portas junto), fazendo com que uma enorme vulnerabilidade se torne presente.

Muitos destes problemas são resultado de falha de configuração após a instalação, bastando para isto reconfigurar os serviços de rede disponíveis nos sistemas operacionais podendo ainda configurar um esquema de filtragem de pacotes/endereços de rede através do *firewall* e de técnicas de detecção de anormalidades como sistemas de detecção de intrusão ou *IDSs*.

2.1.1 – Portas abertas

Os riscos relativos a portas abertas se referem a um maior número de

possibilidades de um potencial invasor ganhar acesso à máquina detentora das portas abertas, aumentando as possibilidades de uma negação de serviço, acesso não autorizado e exploração de brechas de segurança dos serviços disponibilizados pelas portas abertas, podendo levar os serviços que mantém determinada porta aberta se tornar uma possível brecha para a execução de comandos arbitrários no sistema, acesso não autorizado, injeção de código, consultas não documentadas, etc.

O grau de ameaça de sistemas mal configurados que venham a disponibilizar portas *TCP* ou *UDP* é de certo modo variável, pois depende de quais serviços estão disponíveis, das versões instaladas dos softwares que se utilizam e de suas vulnerabilidades, mas em geral o risco de se manter serviços de rede não necessários é grande, devendo-se parar os serviços não utilizados (e mantê-los parados), para que não possam fornecer serviços desnecessários e possíveis brechas que venham a facilitar possíveis varreduras, obtenção de versões, negações de serviço, execução de comandos arbitrários e/ou ainda acessos não autorizados, como dito anteriormente.

2.2 – Acesso remoto e transferências de arquivos

Muitas das necessidades existentes em ambientes computacionais são relativas às transferências de arquivos e acesso remoto aos sistemas através das redes de computadores, normalmente utilizando-se dos serviços de *ftp* e *telnet* para transferir e para acesso remoto respectivamente. Existe ainda os

chamados *serviços r* bastante conhecidos e utilizados no mundo *Unix* como o *remote shell*, *remote login* e o *remote copy* que realizam estes papéis de transferência de arquivos.

O problema destes antigos e conhecidos sistemas de interconexão de rede é a falta de segurança quanto à conexão, fazendo os dados (tanto de nome de usuário e senha quanto os dados em si) passarem *em claro* pela rede, podendo ser facilmente capturados por grampos de rede ou *Sniffers*.

O software *openSSH* prevê as funcionalidades de transferência de arquivos e de acesso remoto além da utilização de criptografia tanto para o processo de autenticação quanto para o tráfego dos dados através da rede.

Uma solução quanto a isso é a utilização do *openSSH* que implementa as duas facilidades, a de acesso remoto e a de transferência de arquivos, em um mesmo pacote de software, através das ferramentas *ssh* e *scp* através da utilização de criptografia, sendo um ótimo substituto ao *telnet* e ao *ftp*. Comentaremos a utilização e configuração destes serviços no apêndice G - *Utilização do Open SSH*.

2.2.1 – Dados em trânsito

Os riscos relativos ao acesso remoto e à transferência de arquivos é que os dados podem ser facilmente visualizados através de *farejadores de pacotes* ou comumente chamados *sniffers*. Estes programas de rede capturam o tráfego em *broadcast*, fazendo a interface de rede atuar em modo *promíscuo*,

sendo um grande problema em redes que utilizam-se de *HUBs como barramento* para o segmento de rede, onde as informações são compartilhadas com todos os computadores da rede ou subrede. Utiliza-se estes serviços para que se possa administrar hosts remotamente para realizar manutenção e/ou monitoramento, sendo um risco para a administração de rede.

Embora o uso de *comutadores* de rede (*switch*) dificulte o *farejamento* de pacotes na rede eles não impedem que isto aconteça, pois em tais equipamentos o endereçamento se dá através do endereço físico do dispositivo de rede, podendo ainda ter as informações que trafegam na rede serem comprometidas com a técnica conhecida como *arpspoof*, onde um dispositivo de rede se faz passar por outro através da alteração de seu endereço físico da interface de rede.

O grande problema relativo aos dados em trânsito é o fato de os nomes de usuários e as senhas passarem sem nenhuma proteção criptográfica ou de qualquer espécie, inclusive os dados; fazendo desta vulnerabilidade um sério risco para qualquer um que possa ameaçar a integridade dos dados ou de seus usuários. Esta vulnerabilidade oferece um alto risco aos dados e aos usuários da empresa.

2.3 – Tráfego http

Praticamente todos os computadores de uso pessoal utilizam-se da internet para alguma finalidade, em geral buscando alguma informação para

fins de pesquisa e consulta de informações além da utilização de alguns serviços disponíveis pela internet, podendo também necessitar prover páginas para a internet.

Existem dois problemas básicos relativos ao tráfego *http*:

- tráfego de conteúdo *http* que entra e sai da rede interna e/ou;
- páginas *web* servidas pela empesa, seja para a rede interna ou para a rede externa.

O tráfego *http* que entra e sai da rede interna é, normalmente, o tráfego de internet dos usuários da rede, podendo ser (e muitas vezes devendo ser) filtrado em nível IP pelo *firewall* e/ou ainda filtrando-se a nível de aplicação através de um serviço de *proxy*, fazendo com que se possa fazer uma filtragem por conteúdo e por endereço.

Páginas *web* servidas pela própria empresa devem ter cuidados especiais com relação ao servidor, pois o serviço *http* é um dos mais explorados atrás de brechas de segurança, devendo ser um serviço bem protegido e com seus riscos minimizados ao máximo, tanto pelo roteador externo à *rede de perímetro* quanto à própria máquina servidora do serviço de *http*.

2.3.1 – Riscos de acesso http

O filtro a nível de conteúdo *http* é de extrema valia para os

administradores, pois muitos conteúdos da *web* são de cunho perigoso tanto quanto à conteúdo como em exploração de vulnerabilidades, tanto na questão de usuários estarem procurando por vulnerabilidades para atacar os computadores da rede como sites perigosos à integridade da rede.

Como é impossível fazer uma busca por todos os sites indevidos ou perniciosos, deve-se estar atento aos registros gerados pelos usuários com o seu tráfego *web* e realizar uma avaliação constante de riscos relacionados aos sites visitados, adicionando os sites considerados perigosos às *listas de controle de acesso* ou *ACLs*.

2.3.2 – Riscos do servidor http

Muitas empresas têm seus *sites web* em servidores dentro da própria empresa, necessitando assim de um servidor de internet atuando na internet e expoxto. O servidor *web* disponível na maioria dos sistemas livres e o mais comum é sem dúvida o *apache*, tendo uma das suas grandes vantagens (ou desvantagem de segurança) a sua modularização.

A maior parte das vulnerabilidades apresentadas pelo *apache* se dá em relação a seus módulos, como o comumente usado *PHP,* pois os módulos acessam outros recursos que são externos ao servidor *web* e podem acessar um módulo com alguma vulnerabilidade que venha a comprometer o sistema no qual o servidor *web* está implementado.

O *apache* é extremamente flexível em termos de configuração e pode ser

configurado de forma bastante segura dependendo do tipo de necessidade de utilização.

2.4 – Malwares e vírus

Os vírus e outros tipos de *malware* (softwares que realizam tarefas maliciosas) estão se desenvolvendo cada vez mais e se transportam rapidamente pela internet. Muitas pessoas estão utilizando-se de ferramentas de software conhecidas como antivírus para verificar seus arquivos em busca de vírus e outros tipos de programas de conteúdo prejudicial.

Uma das principais ferramentas de software livre para este tipo de vulnerabilidade bastante comum é o *clamAV* que implementamos neste trabalho para podermos verificar os arquivos do computador como os arquivos das caixas de entrada do serviço de e-mail e arquivos dos usuários.

3 PROPOSTA DE SOLUÇÃO

Como prevenção e correção dos problemas de qualquer sistema o melhor aliado está no conhecimento dos recursos que compõem o sistema, tanto no que tange ao hardware quanto ao software, à rede, às pessoas envolvidas na utilização, no contexto de utilização, etc.

Abordamos alguns aspectos técnicos da configuração durante a implementação dos sistemas e da rede, juntamente com a utilização de algumas ferramentas de software livre que ajudaram a prevenir ou solucionar alguns aspectos selecionados conforme os problemas apresentados.

Sugere-se neste capítulo algumas formas de configurar o sistema de forma preventiva e algumas ferramentas que ajudam a solucionar aspectos relativos à segurança, além de descrevermos o processo de implementação e de verificação de vulnerabilidades.

3.1 - Descrição da rede de testes

A seguir descreve-se a rede de testes com relação ao seu hardware, software, serviços e configuração além dos testes realizados.

3.1.1 – Objetivos

O objetivo da rede de testes é medir o nível de segurança oferecido pelos softwares livres que provem serviços de TI para as PME e servir de guia para a configuração segura destes serviços baseados nas tecnologias de software livre.

A rede de testes é composta de três computadores do tipo *IBM PC* em uma rede padrão *Ethernet* e foi implementada conforme as recomendações de GOMES (2001), ZWICKY (2000), HICKS (2005), SCAMBRAY (2001), CONECTIVA 1 (2001) e CONECTIVA 2 (2001) nos respectivos capítulos sobre instalação e configuração segura dos sistemas e dos serviços.

Um dos sistemas foi instalado para ser um *firewall* entre a rede interna de testes e a internet (chamado de host de bastião ou ainda bastion host), o segundo para ser um sistema servidor e outro para representar um cliente ou um usuário malicioso (e possível atacante). Apesar de desaconselhado por ZWICKY (2000) este é um recurso que foi verificado no caso das empresas avaliadas e o que necessita-se aqui avaliar é o nível de segurança mais real possível, além de não dispormos de mais recursos em termos de hardware para podermos implementar mais computadores na rede assim como as PMEs muitas vezes não dispõem de recursos para investimentos em mais máquinas.

A rede de testes foi testada com uma ferramenta livre disponível para avaliar a segurança da rede chamada *nessus*. O software *nessus* buscou as vulnerabilidades iniciais e buscamos soluções para as vulnerabilidades

encontradas após as verificações.

3.1.2 - Topologia de redes

A topologia de rede baseou-se no padrão *Fast Ethernet* de 100 Mbits por segundo, com 4 cabos do tipo par trançado categoria 5 e conectores rj45. A rede foi composta de um modem *ADSL D-Link 500G* para acesso à internet, um equipamento do tipo *switch* de rede *Encore* 8 portas e três computadores como ilustrado na figura 2 abaixo:

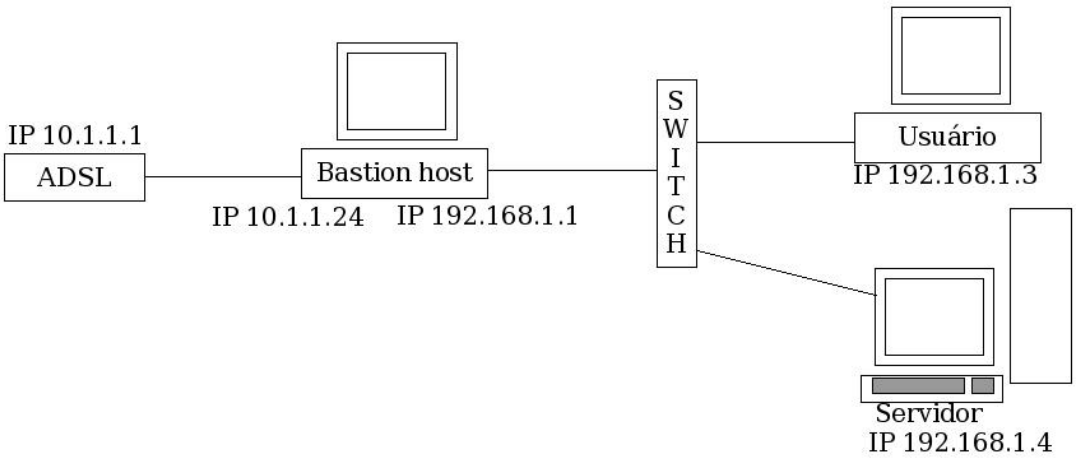


Figura 2– Topologia de rede da rede de testes

A rede de teste mostrada acima é dividida em uma rede externa ao Host de bastião (bastion host) de tipo de endereçamento IP local da classe A de número 10.0.0.0/8 e uma rede interna ao Host de bastião de tipo de endereçamento IP local da classe C de número 192.168.1.0/24.

O modem ADSL possui endereço IP de valor 10.1.1.1 e fornece endereço IP de valor 10.1.1.24 para a interface ethernet 0 (eth0) do host de bastião e o host de bastião tem a sua interface ethernet 1 (eth1) com endereço IP configurado para iniciar com endereço IP de valor 192.168.1.1 e fornece serviço de configuração dinâmica de host (serviço *dhcp*) para os computadores da rede interna conforme o endereço físico (*mac*), apenas par aos computadores da rede, sem exceções. O sistema servidor foi configurado com o nome juarezdc pelo serviço de *DHCP* do *firewall* e recebe o endereço IP de valor 192.168.1.4. O sistema representando sistemas de usuários na rede interna foi configurado pelo serviço *DHCP* do *firewall* com o nome de luiza e recebe o endereço IP de valor 192.168.1.3.

Foi escolhido um equipamento comutador, ou *switch*, para centralizar e concentrar as conexões físicas dos computadores, evitando assim o *grampo de rede* (também conhecido como *sniffing*) de datagramas no segmento da rede com o modo promíscuo da interface de um usuário malicioso ativado, por exemplo.

3.1.3 - Computadores utilizados

Foram utilizados três computadores para a rede de testes, sendo o primeiro o host de bastião, atuando como *firewall* e provedor de serviços inseguros não essenciais ao funcionamento da empresa e ainda serviços que exigem menos recursos de hardware conforme sugerido por ZWICKY (2000, p.

140). O segundo computador, servidor, atuando como servidor de rede e fornecendo os serviços indispensáveis ao funcionamento dos sistemas internos da empresa fictícia e dos dados da empresa, além de executar os serviços mais exigentes em termos de hardware. E o terceiro computador, cliente, simulando clientes da rede interna e possíveis interessados em informações da empresa ou possíveis atacantes, sendo posicionado tanto no segmento da rede interna ao firewall quanto no segmento externo ao firewall.

O particionamento dos serviços foi realizado conforme sugestão de ZWICKY (2000, p.140) onde sugere o particionamento por nível de segurança do serviço e por desempenho da máquina em relação ao serviço. Assim sendo o servidor ficou encarregado de fornecer dos serviços com maior demanda de hardware e com os serviços considerados mais seguros e importantes, o firewall ficou com os serviços menos exigentes em termos de hardware e com os serviços considerados mais arriscados de se manter e menos primordiais ao funcionamento dos recursos da rede interna.

3.1.4 - Hardware

O host de bastião possui um processador *Intel Pentium MMX* de 233 Mhz, com 32 Mbytes de memória RAM, um disco rígido de 1,2 Gbytes e duas interfaces de rede 3com Etherlink III. ZWICKY (2000, pág 253) comenta que o host de bastião deve ser uma máquina confiável e conhecida, não devendo possuir hardware muito poderoso em termos de processamento para evitar que

um possível invasor tenha acesso a uma máquina muito eficiente.

O dispostivo de rede do computador firewall que se conecta ao segmento de rede externo é referenciado no sistema por *eth0* com endereço de físico (conhecido também por endereço *mac*) utilizado para a comunicação com a rede externa é 00:60:97:C4:11:17 e endereço ip 10.1.1.24 recebido por DHCP do modem ADSL. A interface externa tem seu endereço físico de 00:60:97:23:19:15 e é configurada para iniciar com endereço IP de valor 192.168.1.1. O nome do host de bastião é configurado manualmente e tem nome server no domínio interno juarezdc.org.

O sistema servidor possui processador *Intel Pentium 4 HT* de 3.0Ghz, com 512 Mbytes de memória RAM, 80GBytes de disco rígido, e uma interface de rede sis900 de endereço físico 00:11:D8:36:D8:78 recebendo endereço IP de valor 192.168.1.4 e nome juarezdc através do serviço DHCP do sistema firewall.

O sistema cliente, ou atacante, possui um processador *Intel Pentium 3* de 750Mhz com 256 Mbytes de memória RAM, 40GBytes de disco rígido e placa de rede Realtek 8139c de endereço físico 00:e0:7d:de:c9:61 recebendo endereço IP de valor 192.168.1.3 e recebe o nome de luiza quando posicionado na rede interna através do serviço DHCP do firewall e recebendo o endereço IP de valor 10.1.1.5 do modem ADSL quando posicionado no segmento externo ao firewall.

3.1.5 - Sistemas operacionais utilizados

O sistema instalado nos sistemas são provenientes da distribuição *Slackware GNU/Linux* versão 10.2 obtido em <www.slackware.org>, tanto no *bastion host* quanto no sistema *servidor* e no computador *cliente/atacante* o computador cliente utilizou-se ainda do sistema operacional Microsoft Windows 2000 para simular a atuação de um cliente da rede interna.

Os pacotes instalados no *firewall* foram apenas o cd1 da distribuição. A distribuição separa os pacotes de software de acordo com as funcionalidades básicas dos softwares, esta separação é descrita como uma série. As séries são explicadas por no ítem 2.3.1 de HICKS (2005).

Para o bastion host utilizou-se as séries *a*, *ap*, *d*, *l*, *n* e *y*. No sistema servidor e no computador cliente (na instalação do linux do computador cliente) utilizou-se as séries *a*, *ap*, *d*, *f*, *k*, *l*, *n*, *x*, *xap* e *y*.

3.1.5.1 - Serviços utilizados

O host de bastião tem a tarefa de atuar como *proxy* para os clientes *http*, *cache* de DNS e serviço *dhcp* e monitorar o tráfego da rede interna para a rede externa, sendo aqui apenas saída para o modem ADSL e conseqüentemente para a internet.

3.1.5.1.1 - Cache de DNS com o BIND

Foi utilizado o software BIND versão 9.3.1 que estava disponível na instalação da distribuição *slackware 10.2*.

A configuração do *BIND* foi feita com um domínio interno de exemplo de nome *juarezdc.org* mapeando os endereços de hardware com endereços IP e com o domínio designados especificamente a estes endereços físicos pelo serviço *DHCP* e sendo atualizada a base de dados de hosts pelo serviço *DHCP*.

3.1.5.1.2 - Serviço de DHCP

O serviço de configuração dinâmica de host (*DHCP*) foi configurado com endereços fixos para esta rede local, já que continha poucos hosts.

O sistema servidor recebe o endereço IP 192.168.1.4 e o nome *juarezdc.juarezdc.org,* o sistema cliente recebe o endereço IP 192.168.1.3 e nome *luiza.juarezdc.org*.

Em uma rede de maior porte ou de necessidades diferentes, como em redes que necessitem de adição dinâmica de hosts, em uma rede acadêmica de universidade, por exemplo, deve-se separar as subredes com sistemas semelhantes ao bastion host do tipo *dual homed* (ZWICKY, 2000 p. 105) para manter-se segura as redes internas onde estão os usuários da empresa. Os sistemas que separam as outras subredes devem ser basicamente uma cópia

das regras de filtragem do host de bastião de entrada da *rede de perímetro* ajustados conforme as necessidades de cada subrede, com as mesmas regras de filtragem em geral, sofrendo apenas as adaptações necessárias ao contexto de utilização de cada subrede.

Então o sistema *firewall* separador da subrede forneceria os serviços de *dhcp* e sub-árvore *dns* para a subrede.

Foi utilizada a versão 3.0.3 do servidor *dhcpd* disponível no *slackware 10.2*.

3.1.5.1.3 - mysql

Utilizamos o sistema de banco de dados *mysql* versão 4.1.14 disponível na distribuição *slackware 10.2*, para utilizar em conjunto com o *ACID* e o *snort*. O *mysql* foi utilizado no sistema *firewall* a fins de registros apenas, e no sistema servidor simulando uma base de dados operacional na rede interna de uma empresa.

A configuração do *mysql* está no apêndice A – *Configuração do mysql*.

3.1.5.1.4 - Apache

O *apache* é um dos servidores *web* mais populares e utilizados do

mundo, suportando diversos módulos para as mais diversas funcionalidades. O apache aqui apresentado é a versão 1.3.33 disponível na distribuição *slackware 10.2*.

Nesta seção descrevemos a instalação de um sistema de verificação dos registros do *snort* passados para o banco de dados *mysql*. Esta funcionalidade serve de exemplo para aplicações *web* reais e serviu de exemplo para a implementação de uma configuração preventiva, um pouco mais segura, do servidor *web apache* com controle de acesso e configurações de rede do servidor.

3.1.5.2 - Softwares instalados

Descrevemos aqui os softwares instalados nos sistemas da rede de testes.

3.1.5.2.1 - snort

O sofware *snort* é um sistema de detecção de intrusão de rede, podendo ser utilizado para análise do tráfego da rede atuando como um *farejado de pacotes*, *grampo de rede* ou ainda *sniffer.* O *snort* foi utilizado como sistema de detecção de intrusão de rede para avaliar anomalias no tráfego da rede e gerar

dados para o banco de dados do *mysql* no host de batião e apenas como sistema de detecção de intrusão no sistema servidor, mas sem utilizar o *mysql*.

A instalação do snort é descrita no apêndice D - Instalação e configuração do *snort.*

3.1.5.2.2 - nessus

O *nessus* é um software de detecção de vulnerabilidades da rede, sendo muito útil na verificação de aspectos críticos à segurança dos hosts da rede. O *nessus* é capaz de realizar varreduras simultâneas de hosts da rede, realizar varreduras nas portas *TCP* e *UDP* (*portscans*), verificar as versões de sistemas operacionais e ferramentas além de procurar por vulnerabilidades nos serviços sendo executados nas máquinas dentre outras funcionalidades como a exibição de relatórios nos mais diversos formatos.

O *nessus* funciona sob o conceito de arquitetura *cliente-servidor*, possuindo muitas configurações customizáveis para garantir segurança, como autenticação, certificados, configuração de clientes e permissões. Instalaremos o nessus da forma mais simples para apenas detectar as vulnerabilidades da rede e exibir em um relatório para o administrador.

Como o *nessus* é um software exigente em termos de hardware, foi instalado apenas no sistema servidor e no sistema atacante, que possui mais recursos em termos de hardware. A instalação do nessus está descrita no apêndice E – *Instalação e configuração do nessus.*

3.1.5.2.3 – ACID

O *ACID* é uma ferramenta *web* para visualizar os registros do *snort* realizadas em um banco de dados como o *mysql*. A instalação do *ACID* é descrita no apêndice B - *Instalação dos arquivos do ACID no apache*.

3.1.5.2.4 - squid

O software *squid* é um software *proxy* com características de *cache http* para hosts clientes. O *proxy* é um procurador de endereços *http*, onde os softwares de internet são configurados para realizar suas requisições através do servidor *proxy*.

O *squid* aceita diversos tipos de filtragem, como por endereço físico (*mac*), host, redes, portas TCP, domínios e ainda expressões. A documentação sobre a configuração do *squid* e das *ACLs* encontra-se no manual de configuração do *squid* disponível em: <http://squid.visolve.com/squid/configuration_manual_24.htm>.

3.1.5.2.5 - clamAV

O *clamAV* é um antivírus baseado na tecnologia de software livre para sistemas Unix. A instalação e configuração do *clamAV* é descrita no apêndice G – *Instalação do antivírus clamAV.* O *clamAV* foi instalado no host de bastião e no sistema servidor.

3.2 – Verificação de vulnerabilidades na rede

Esta seção dedica-se à varredura em busca de vulnerabilidades nos sistemas da rede de testes.

Aplicou-se a varredura do *nessus* que é executado em maior profundidade e também realiza varreduras específicas para cada tipo de serviço encontrado. Tendo sido gerados relatórios dos testes realizados com o *nessus* constantes nos anexos.

3.2.1 – Verificação de vulnerabilidades com o Nessus

O software *nessus* é voltado para a detecção de vulnerabilidades na rede e possui diversos *plug-ins* para varreduras dos mais variados tipos de serviços. A instalação e configuração do *nessus* é descrito no apêndice E – *Instalação e*

configuração do nessus.

Para executar uma verificação da rede primeiro devemos executar o serviço do *nessus* através do comando:

```
#nessusd -D
```

O serviço demora para executar na primeira vez pois cerregará os *plug-ins*. Deve-se lembrar de que o *nessus* atua na arquitetura cliente-servidor e deixará uma porta TCP aberta, devendo então ser executado apenas durante verificações da rede.

A seguir deve-se executar um cliente para o *nessus*. No linux devemos estar executando um ambiente gráfico para podermos utilizar o cliente do *nessus*. Para executarmos o cliente basta teclarmos em um terminal a partir de um servidor *X11* ou *XFree*:

```
# nessus &
```

A seguinte janela aparecerá:

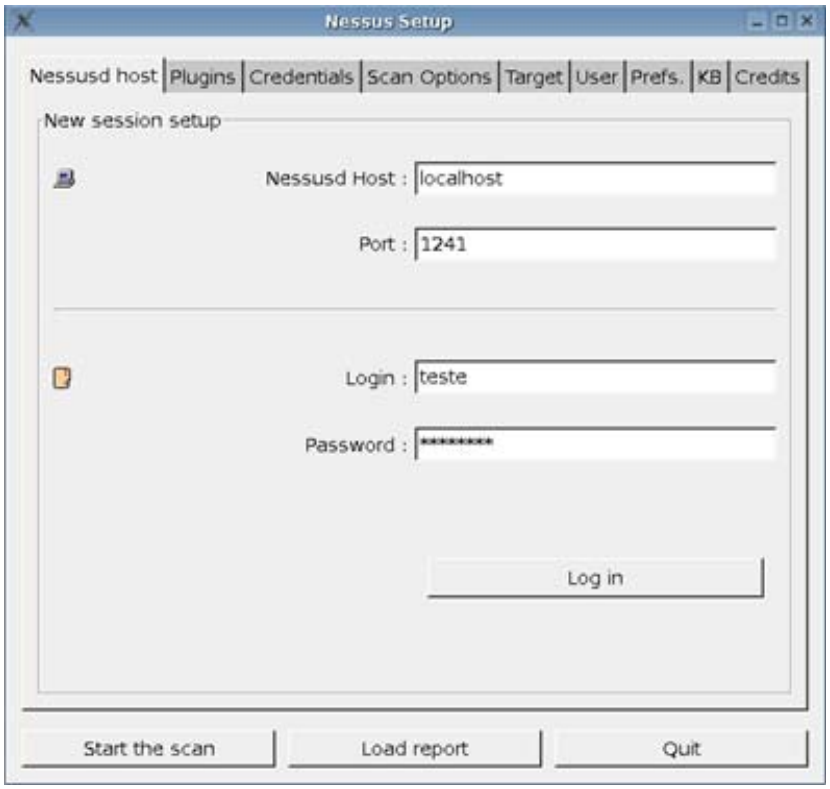


Figura 3 - janela de login do cliente nessus

Insira o nome e a senha do usuário criado anteriormente e clique no botão *log in*. O processo de autenticação pode demorar um pouco, após o processo de autenticação, será exibida a seguinte aba:

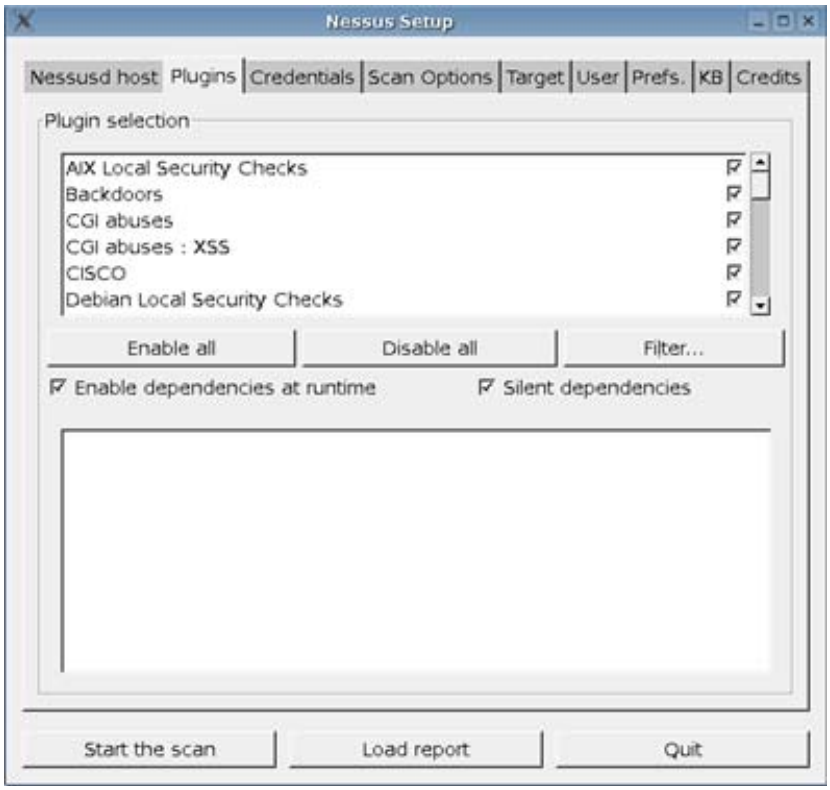


Figura 4 - janela de plug-ins do cliente nessus

A aba acima determina quais *plug-ins* devem ser utilizados na verificação, escolhemos todos para verificar a rede de testes.

Deve-se especificar quais hosts da rede se deseja verificar. Para os testes da rede de testes foram utilizados os endereços 192.168.1.1, 192.168.1.3, 192.168.1.4, 10.1.1.1, 10.1.1.24 e, quando o computador cliente está posicionado no segmento externo ao host de bastião, utilizou-se também o endereço IP 10.1.1.5 e não o endereço IP 192.168.1.3.

3.2.1.1 – Vericações realizadas nos sistemas

Os sistemas foram verificados em cinco momentos diferentes, sendo detectadas algumas vulnerabilidades e aplicando medidas de prevenção para serem testadas no próximo teste após as correções realizadas.

Identificaremos aqui apenas as vulnerabilidades e correções que representam um risco considerável para a integridade dos dados da empresa com relação à importância do serviço prestado.

Os relatórios do nessus constam em anexo ao trabalho e contém gráficos que podem ser comparados ao longo das datas de verificação. As configurações utilizadas como forma de intervenção nos sistemas constam na seção 3.3 – Prevenção das vulnerabilidades**.**

3.2.1.1.1 – Verirficação de 15/11/2005

Em 15/11/2005 foi realizada uma verificação inicial após a instalação dos sistemas sem configurações preventivas e sem sistema de *proxy* com o *squid*, sendo o computador *firewall* apenas um roteador e o sistema cliente estava executando o windows 2000 representando os clientes da rede interna.

Segundo o relatório do *nessus,* constantes nos anexos A, B, C e D, foram detectadas as seguintes vulnerabilidades:

a) Firewall:

Porta *TCP* 22 do *ssh* – Apenas registrou a porta aberta, o teste representou o serviço como de baixo risco e foi realizada a configuração do ssh conforme o anexo G – *Utilização do Open SSH* no ítem *Configurando o servidor ssh para aceitar apenas a versão 2*;

- Porta *TCP* e *UDP* 53 do *bind* – Apenas registrou a porta aberta, o teste detectou o serviço como de baixo risco nos aspectos apresentados. Nada foi feito;
- Porta *TCP* 80 do *apache* – Foram encontradas duas vulnerabilidades de alto risco envolvendo a possibilidade de ataque de *injeção de sql* no ACID; uma vulnerabilidade de baixo risco envolvendo diretórios não utilizados no apache, o apache foi configurado conforme o apêndice x e só seria iniciado quando o administrador fosse analisar os registros do ACID e após isto o *apache* deve ser finalizado o que não serve para servidores *web* externos que devem estar sempre disponíveis;
- Porta *TCP* 139 e 445 e porta *UDP* 137 do compartilhamento de rede Windows com o S*amba* – Encontrou informações a respeito do compartilhamento de arquivos CIFS/SM B do S*amba* e este foi finalizado no *firewall* devido à falta de necessidade do serviço neste host;
- Porta *TCP* 3306 do *mysql* – Apenas encontrou a porta do *mysql*, não representa risco de acesso, a porta foi bloqueada para acessos da rede externa;
- Porta *UDP* 67 do servidor *dhcp* – Foi detectada a porta do servidor

DHCP e registrou um tipo de relatório do tipo *falso positivo*, recomendando estar com a versão atualizada do servidor, o que de fato, na data, estava atualizado;

b) Servidor:

- Porta *TCP* 22 do *ssh* – As detecções e prevenções foram idênticas à do sistema *firewall*;
- Porta *TCP* 139 e 445 e porta UDP 137 do compartilhamento de rede windows com o *samba* – Encontrou informações a respeito do compartilhamento de arquivos CIFS/SMB do *samba* a prevenção se deu através da filtragem de datagramas no servidor para impedir este tráfego de passar para a rede interna;
- Porta *TCP* 6000 do servidor gráfico *X11*: Foi encontrada a porta 6000 do servidor gráfico X11, esta porta foi bloqueada tanto no *firewall* quanto no sistema servidor. Não representa um risco sério tendo sido filtrada com o *iptables*;

c) Cliente:

O computador cliente possuía tantas vulnerabilidades que a opção foi mais plausível foi de bloquear a entrada de qualquer tipo de tráfego destas portas para a rede interna no *firewall*.

3.2.1.1.2 – Verirficação de 22/11/2005

A segunda verificação ocorreu no dia 22/11/2005 e segundo os relatórios do *nessus,* constantes nos anexos E, F, G e H, foram observadas as seguintes características no *firewall* e no servidor:

a) *Firewall*: O *firewall* apresentou um relatório bastante seguro, já não possuindo vulnerabilidades de alto risco. O servidor *web* já não possuia riscos relativos às *CGIs* e às *injeções de SQL* no *ACID* inclusive estando a esperar conexões apenas na interface da rede interna (192.168.1.1). O grau de proteção do *firewall* chegou a um nível bastante satisfatório e foi apenas adicionado o serviço de *proxy* com o *squid* após o teste e o roteamento foi desabilitado com a implementação do *proxy*;

b) Servidor: O servidor também apresentou um relatório com um quadro bastante seguro segundo os relatórios do *nessus*, não necessitando de alterações embora o compartilhamento de arquivos através do *SMB/CIFS* ainda exibisse bastante informação relativa ao host, devendo lembrar que o firewall está bloqueando o acesso às portas relativas ao *samba*;

3.2.1.1.2 – Verirficação de 10/12/2005

A terceira e última verificação ocorreu no dia 10/12/2005 e segundo os relatórios do *nessus,* constantes nos anexos I, J, K e L, foram observadas as seguintes características no *firewall* (atuando agora como *proxy*) e no servidor:

a) Firewall: Nesta verificação o *firewall* não realiza mais a tarefa de roteamento, impedindo assim que sistemas externos o utilizem de roteador para acessar a rede interna. Neste aspecto a segurança da rede interna aumentou, mas diminuiu a segurança do *firewall* com um serviço a mais sendo executado porém não foi descrita nenhuma vulnerabilidade preocupante nos registros do *nessus*;

b) Servidor: O sistema servidor manteve-se com riscos relativamente baixos segundo os relatórios do nessus, apenas com algumas portas a mais abertas, como a de serviço *POP* e *IMAP* destinadas ao tráfego de e-mail que, se necessário for, podem facilmente serem filtradas no *iptables* do servidor, porém nenhuma medida preventiva foi realizada.

Após as verificações dos principais sistemas de segurança da rede de testes e das medidas preventivas adotadas após estas verificações os sistemas demonstraram estar bastante confiáveis segundo os relatórios do *nessus*.

As prevenções

3.3 – Intervenção nos sistemas

Apresentamos nesta seção algumas configurações utilizadas na instalação no sistema servidor e no bastion host de forma preventiva tentando minimizar os riscos dos sistemas em relação à sua instalação e utilização da rede.

3.3.1 – Prevenindo as portas TCP

O problema relativo às portas abertas está normalmente relacionado a serviços que são deixados em execução após a instalação do sistema. Para prevenir que portas *TCP* e *UDP* desnecessárias fiquem abertas é necessário que se interrompa a execução de serviços que não serão utilizados, uma configuração cuidadosa dos serviços que devem ser utilizados e a utilização de ferramentas que venham a facilitar a administração e manter o sistema seguro e em ordem, como por exemplo a filtragem de pacotes da rede.

3.3.1.1 – Finalizando os serviços desnecessários

Cada tipo de sistema *Posix* ou variante de *unix* ou ainda cada tipo de distribuição de linux, por exemplo, tem suas particularidades quanto à

configuração dos serviços de rede disponíveis. Estes serviços são geralmente interrompidos durante o processo de inicialização para que não venham a ser carregados durante a inicialização da máquina.

Estas configurações também variam conforme o tipo de sistema utilizado. Como se estamos utilizando para os testes a distribuição *slackware* GNU/Linux explicaremos as configurações aplicada a este sistema, seguindo as recomendações de ZWICKY (2000, p. 284), ANÔNIMO (2000, p.69) e HICKS (2005, cap 14).

3.3.1.1.1 - Interrompendo os serviços no Slackware

Seguiu-se as recomendações de ZWICKY (2005, p. 284) e HICKS (2005, cap. 14) para implementar um sistema base para as diretivas a serem aplicadas nos computadores da nossa solução.

Para iniciar, interromper ou parar os serviços do sistema na distribuição *slackware* deve-se ir até o diretório */etc/rc.d*:

```
#cd /etc/rc.d
```

A seguir deve-se verificar as permissões de execução do arquivo desejado, exemplificaremos com o serviço *http*:

```
# ls -l rc.httpd
```

A saída resultante do comando acima (estando no diretório */etc/rc.d*) foi:

```
-rwx--x--x  1 root root 401 2003-03-05 18:28 rc.httpd
```

O caractere *x* no início da saída nos indica que o *script* está com a permissão de execução ativada. Para saber se o processo está em execução na memória basta verificar na lista de processos se existe um processo chamado *httpd*. Para verificar isto tecla-se em um terminal:

```
# ps ax | grep httpd
```

Caso a resposta seja semelhante a isto:

```
# ps ax | grep httpd
 2064 ?        Ss     0:00 /usr/sbin/httpd
 2065 ?        S      0:00 /usr/sbin/httpd
 2066 ?        S      0:00 /usr/sbin/httpd
 2067 ?        S      0:00 /usr/sbin/httpd
 2068 ?        S      0:00 /usr/sbin/httpd
 2069 ?        S      0:00 /usr/sbin/httpd
```

Significa que o serviço está executando. Caso não haja resposta ou a resposta se assemelhe a isto:

```
# ps ax | grep httpd
 2094 ?        Ss     0:00 grep httpd
```

Significa que o serviço não está rodando. Para se desativar o serviço, caso esteja ativo, basta executar o *script* com o parâmetro *stop*:

```
# ./rc.httpd stop
/usr/sbin/apachectl stop: httpd stopped
```

E o serviço não estará mais em execução. Para tanto verifique novamente com o comando *ps ax* | *grep httpd*.

Cada arquivo de *script* disposto em */etc/rc.d* inicializa um tipo de serviço diferente ou um nível de execução. Abordou-se os serviços de rede que devem ser configurados durante uma instalação. Segue uma listagem dos arquivos de *script* mais importantes segundo HICKS (2005, seção 4.1.3):

- rc.atalk – Inicializa, pára ou reinicia o suporte ao protocolo *AppleTalk;*
- rc.httpd – Relativo ao servidor *web apache;*
- rc.samba – *Script* relativo ao compartilhamento de arquivos através dos protocolos *CIFS/SMB* relativos ao compartilhamento de arquivos em redes *windows*;
- rc.alsa – *Script* relativo aos *módulos* (*drivers*) do sistema de som *ALSA* do linux;
- rc.acpid – *Script* relativo ao suporte *ACPI*, como reconhecer as teclas de *sleep*, *power*, e *wake up* de teclados e placas que suportem o padrão *ACPI*;
- rc.bind – *Script* relativo ao servidor de nomes. Se seu computador não é o *DNS* desative-o;
- rc.cups - *Script* relativo ao servidor de impressão *CUPS*;
- rc.dnsmasq - serviço relativo a um serviço de *DHCP* e servidor de nomes;

- rc.gpm - Serviço de suporte ao mouse, copiar e colar texto no console;
- rc.hotplug - Inicializa os serviços de interconexão de dispositivos do tipo *hotplug* como *PCMCIA*, *USB* e assemelhados;
- rc.inet1 - Este *script* inicializa ou desativa as interfaces de rede;
- rc.inet1.conf - Arquivo de configuração dos *módulos* das interfaces de rede. Utilizado por *rc.inet1*;
- rc.inet2 - Executa todos os demais *scripts* de rede, inicializando todos os serviços relativos à rede que tenham permissão de execução habilitada;
- rc.inetd - *Script* relativo ao *Internet Super Server (conhecido por inet)*, inicializa serviços por demanda e abarca outros serviços como *ftp*, *telnet*, serviços de execução e autenticação remota, *finger*, etc;
- rc.ip_forward - Ativa ou encerra a função de encaminhamento IP do *kernel*;
- rc.keymap - Carrega o mapa de teclado;
- rc.local - Executa comandos que o usuário gostaria de executar na inicialização. Os comandos desejados pelo usuário devem ser colocados aqui. Adicionaremos mais tarde alguns comandos aqui como filtragem de pacotes, inicialização de programas instalados, etc;
- rc.modules - Carrega os módulos do *kernel* relativos aos dispositivos de hardware;
- rc.mysqld - Script de inicialização do servidor de bancos de dados MySQL;

- rc.netdevice - Carrega o *módulo* do *kernel* relativo à interface de rede;
- rc.nfsd – Inicializa ou desativa o serviço de *Network Filesystem* ou *NFS*;
- rc.pcmcia - Inicializa os dispositivos PCMCIA;
- rc.portmap – Inicializa ou interrompe o serviço *portmap* que é requerido pelo NFS por exemplo;
- rc.sendmail – Inicializa ou interrompe o servidor de e-mail *sendmail*;
- rc.serial - Inicializa dispositivos seriais, como as portas seriais;
- rc.sshd - inicializa o servidor *OpenSSH* que realiza transferência de arquivos e acesso remoto de forma totalmente *criptografada*;
- rc.syslog - Inicializa o serviço de registros do sistema;
- rc.sysvinit - Inicializa o serviço de compatibilidade com o sistema de *scripts* de inicialização baseados no *System V*;
- rc.udev - Inicializa o serviço de sistema *UDEV* que inclui apenas os arquivos relativos aos dispositivos existentes no diretório */dev*;
- rc.wireless - Inicializa as interfaces de rede que se utilizam do padrão *wireless*;
- rc.wireless.conf - Configurações do *script rc.wireless*;
- rc.yp - Inicializa o serviço *Network Information Service* ou *NIS* ou também conhecido por *Yellow Pages*.

Para que nosso sistema seja mais seguro ele deve inicializar apenas os serviços que serão realmente utilizados. Os demais serviços serão primeiro parados e depois terão suas permissões de execução removidas para que não sejam inicializados quando o computador iniciar.

Para parar os serviços no *slackware* basta se posicionar no diretório

/etc/rc.d:

```
#cd /etc/rc.d
```

A seguir é necessário parar os serviços, bastando executar o *script* desejado seguido do parâmetro *stop*. Por exemplo, para se interromper o serviço *httpd* do *apache* deve-se executar os comandos:

```
#./rc.httpd stop
/usr/sbin/apachectl stop: httpd stopped
```

Realiza-se a operação de parar cada um dos serviços que não serão utilizados pelo sistema. No caso parou-se os seguintes serviços:

```
#./rc.wireless stop
#./rc.yp stop
#./rc.hotplug stop
#./rc.inetd stop
#./rc.nfsd stop
#./rc.pcmcia stop
#./rc.samba stop
#./rc.sshd stop
```

A seguir se deve retirar a permissão de execução de cada um dos arquivos desabilitados. Para isto utilizou-se o comando *chmod* que altera as permissões de arquivos no sistema de arquivos do linux:

```
#chmod -x rc.wireless
#chmod -x rc.yp
#chmod -x rc.hotplug
#chmod -x rc.inetd
#chmod -x rc.nfsd
#chmod -x rc.pcmcia
#chmod -x rc.samba
```

#chmod -x rc.sshd

Os serviços que restaram com permissão de execução no diretório */etc/rc.d* foram:

- rc.acpid
- rc.font.new
- rc.gpm
- rc.inet1
- rc.inet2
- rc.keymap
- rc.local
- rc.modules
- rc.netdevice
- rc.serial
- rc.syslog
- rc.sysvinit

Os *scripts* que deve-se manter com permissões de execução para que o sistema *funcione* normalmente são:

- rc.4
- rc.6
- rc.K
- rc.M
- rc.S

Visto que estes arquivos são imprescindíveis ao bom funcionamento do sistema eles devem ser mantidos intactos.

Para reiniciar os serviços basta realizar o processo inverso, ou seja, colocar a permissão de execução no arquivo de *script* do serviço desejado e executar o *script* com o parâmetro *start*.

Exemplificaremos com este exemplo de iniciar o serviço de *Secure Shell*, também conhecido como *sshd*.

Primeiramente colocou-se a permissão de execução de volta no arquivo de *script*. Estando-se posicionado no diretório */etc/rc.d* executa-se o comando:

```
#chmod +x rc.sshd
```

A seguir basta executar o *script* com o parâmetro *start*:

```
#./rc.sshd start
```

Caso seja a primeira execução de inicialização do serviço serão exibidas mensagens de geração das chaves criptográficas públicas e privadas e após isto o serviço estará funcionando normalmente. Para verificar a existência do processo em execução na memória basta digitarmos o seguinte comando:

```
#ps ax | grep sshd
 1444 ?        Ss     0:00 /usr/sbin/sshd
 1487 pts/1    S+     0:00 grep sshd
```

A linha contendo a *string /usr/sbin/sshd* indica que o processo está em memória.

Uma sugestão realizadas por ZWICKY (2000, p.287) foi a de editar os arquivos de *script* que não são mais utilizados e comentar as seções que não mais serão utilizadas.

3.3.1.2 – Filtragem de datagramas da rede:

Uma das características principais do linux, e dos sistemas do tipo *Unix,* é a disponibilidade de um filtro de datagramas IP de modo nativo ao *kernel*. O *firewall* (filtro de datagramas) do linux existe desde as versões 2 e superiores, sendo chamado de *ipfwadm* nas versões 2.0 do *kernel*, *ipchains* na versão 2.2 e *iptables* ou *netfilter* nas versões 2.4 e 2.6 do *kernel*.

Para resolvermos alguns problemas de mantenebilidade de portas TCP e UDP que devem ser mantidas abertas devido a serviços necessários pelos usuários, podemos utilizar o *netfilter* do linux para filtrar o tráfego destas portas que devem ser mantidas abertas.

3.3.1.2.1 – O comando iptables

O comando *iptables* fornece uma interface de controle para o suporte ao filtro de pacotes do *kernel* e a sua flexibilidade é grande, sendo bastante robusto e flexível, aceitando uma variedade enorme de opções de configuração. Devido à alta complexidade de opções disponíveis documentaremos exclusivamente as opções utilizadas. A documentação completa pode ser encontrada na página de manual do *iptables* ou em http://www.netfilter.org/documentation/. Para maiores informações a página oficial do projeto está disponível em NETFILTER (2005).

Utilizamos algumas regras de filtragem conforme as necessidades de cada serviço e cada regra é baseada na rede que utiliza o serviço, nas permissões que cada máquina da rede tem de acessar cada tipo de serviço.

3.3.1.2.2 – Funcionamento do iptables

O *iptables* é um filtro de pacotes TCP/IP que acompanha o *kernel* linux e suporta vários aspectos da família de protocolos TCP/IP, podendo-se aplicar regras de filtragem em quase todos os níveis da pilha de protocolos.

O filtro do linux é baseado na idéia de entrada e saída de datagramas em dois modos: entrando e saindo da máquina local ou entrando na máquina local com destino a outro endereço IP, este último serviço é chamado a nível de *kernel* de *IP forward* ou *forwarding*.

A literatura a respeito e a página de manual do *iptables* trata cada pacote de dados identificando-o e comparando-o às regras de filtragem em cada um

dos estados de entrada e saída, para pacotes destinados e originados no computador local, ou ainda em termos de pré-encaminhamento, pós-encaminhamento e saída para os pacotes destinados à outros computadores da rede, estas definições são chamadas de *chains* pelo *iptables*.

Além do conceito acima existem ainda as tabelas de regras de filtragem que são listas das ações a serem realizadas quando um pacote se encaixa em determinada regra de filtragem.

Cada tabela utiliza-se de determinadas *chains*, que são os pontos onde o pacote deve ser verificado. Explicaremos aqui apenas os recursos utilizados, devendo ser verificada a página de manual do *iptables* para maiores informações, não se restringindo aqui os recursos do *iptables*.

Seguimos as recomendações de filtragem de CONECTIVA 1 (2001, Apêndice A), HICKS (2005, seção 14.2.1) e ZWICKY (2000, capítulo 8).

As tabelas usadas e suas *chains* foram:

- Tabela de filtragem: denominada *filter* pelo *iptables* ela é a tabela padrão utilizada pelo *iptables* quando não especifica-se a tabela. Esta tabela é utilizada para se registrar, negar, bloquear ou permitir a passagem ou entrada de pacotes nas chains *input* (entrada para a máquina local), *forward* (repasse dos pacotes para outros computadores ou para outra interface) e *output* (pacotes gerados na máquina local que estão saindo de alguma interface do computador);
- Tabela de conversão de endereços (*nat):* esta tabela é a tabela de tradução de endereços e permite realizar a troca de endereços de origem e destino dos pacotes, permitindo também o redirecionamento

de pacotes para outros endereços IP e/ou portas TCP. Esta tabela opera sobre as *chains prerouting* (pacotes entrando em alguma interface do computador local com destino a outros computadores), *postrouting* (pacotes saindo de alguma interface da máquina local com destino a outros hosts em caso de repasse do pacote) e *output* (pacotes saindo de alguma interface do computador);

Com a tabela de filtragem realizamos os bloqueios e registros de conexões, endereços, portas e etc que não se enquadram nas especificações de segurança da rede a ser protegida conforme os riscos, serviços utilizados e conforme a política de segurança da empresa.

A tabela *nat* é usada em dois momentos diferentes para propósitos diferentes. Em um primeiro momento utilizamos a tabela *nat* para realizar a troca de endereço de origem dos pacotes saindo do sistema enviados pelos computadores da rede interna para o endereço IP do roteador, afim de que o roteador entregue os pacotes com destino à rede externa e fazendo o encaminhamento para o host solicitante na rede interna.

Para que se possa utilizar as *chains* FORWARD e POSTROUTING é necessário ativar o encaminhamento IP do *kernel* com o comando:

```
#echo 1 > /proc/sys/net/ipv4/ip_forward
```

A seguir insere-se o seguinte comando no iptables:

```
#iptables -t nat -A POSTROUTING -i eth0 -j SNAT –to 10.1.1.24
```

Os comandos acima ativam o roteamento de pacotes IP e devem ser empregados apenas em sistemas que fazem roteamento. Em sistemas do tipo *proxy* (e outros tipos de sistemas servidores) o roteamento e encaminhamento de datagramas deve estar desativado conforme sugestão de ZWICKY (2000, p.268).

Algumas opções do *iptables*:

- -t <tabela>: Esta opção especifica a tabela à qual a regra deve ser adicionada. Como a tabela de filtragem é a padrão no caso de não especificar a tabela, só utilizamos esta opção quando se trata de implementar regras na tabela de conversão de endereços*;*
- -A <chain a ser utilizada>: Adiciona uma regra à *chain* especificada, pode ser *INPUT*, *OUTPUT* e *FORWARD* para a tabela de filtragem e *PREROUTING*, *OUTPUT* e *POSTROUTING* para a tabela de conversão de endereços;
- -s <endereço IP de origem>: Especifica o endereço IP de origem do pacote;
- -d <endereço IP de destino>: Especifica o endereço IP de destino do pacote;
- -i <interface de entrada>: Especifica dispositivo de rede no qual o pacote chegou;
- -o <interface de saída> : Especifica dispositivo de rede no qual o pacote sairá;
- -p <protocolo>: especifica o tipo de pacote de dados a ser utilizado.

Pode receber como parâmetro as opções *icmp*, *tcp* e *udp*;

- --sport <porta de origem>: Especifica a porta de origem do pacote no nivel do protocolo *tcp* ou *udp*. Esta opção é válida somente com a opção *-p tcp* ou *-p udp*;
- --dport <porta de destino>: Especifica a porta de destino do pacote no nivel do protocolo *tcp* ou *udp*. Esta opção é válida somente com a opção *-p tcp* ou *-p udp*;
- -j <ação a ser tomada>: Especifica a ação a ser tomada com o pacote que se enquadra na regra. Esta opção aceita os parâmetros *REDIRECT*, *ACCEPT*, *DENY*, *DROP*, *LOG*. As ações variam conforme a tabela e com a *chain* especificada;
- --to <endereço IP>: Esta opção é utilizada quando a ação for *REDIRECT* ou *SNAT* e serve para especificar para qual endereço IP o pacote deve ser redirecionado (no caso de se usar a ação *REDIRECT*) ou para especificar o endereço de origem que será colocado como sendo o endereço de origem do pacote;
- --to-port <porta tcp ou udp>: especifica a porta na qual o pacote deve ser encaminhado quando se utiliza a ação *REDIRECT*.

O *iptables* controla os parâmetros e as opções em conformidade com a utilização, portanto o *iptables* não aceita comandos obviamente errados como, por exemplo, especificar a opção da interface de entrada (-i <interface>) para utilizar em conjunto de uma regra *SNAT* que deve ter a opção da interface de saída (-o <interface>); portanto tenha cuidado ao especificar as suas regras.

A título de exemplo, no *iptables*, os comandos para serem executados no

firewall para bloquear o tráfego da rede *windows* para a interface externa e, conseqüentemente, para a rede externa e internet seriam:

```
#iptables -A INPUT -i eth0 -d 192.168.1.0/24 -p tcp –dport 139 -j REJECT
#iptables -A INPUT -i eth0 -d 192.168.1.0/24 -p tcp –dport 445 -j REJECT
#iptables -A INPUT -o eth0 -s 192.168.1.0/24 -p udp –dport 138 -j REJECT
#iptables -A INPUT -o eth0 -s 192.168.1.0/24 -p udp –dport 445 -j REJECT
```

Estas regras de filtragem foram colocadas no arquivo */etc/rc.d/rc.local* para que sejam utilizadas na inicialização do host de bastião.

Caso desejar registrar as tentativas nos arquivos de registro do sistema basta substituir o parâmetro *REJECT* por *LOG*, mas ainda sim tendo de se usar as linhas que contém *REJECT*. Exemplo para registrar o tráfego e negar o tráfego para a porta TCP 139 com destino à rede interna (192.168.1.0) vindos da interface externa (eth0):

```
#iptables -A INPUT -i eth0 -d 192.168.1.0/24 -p tcp –dport 139 -j LOG
#iptables -A INPUT -i eth0 -d 192.168.1.0/24 -p tcp –dport 139 -j REJECT
```

3.3.2 – Alternativas seguras à transferência de arquivos e acesso remoto

Como citado anteriormente, a necessidade de utilização da rede envolve a transferência de arquivos e a necessidade de acesso remoto em alguns casos, como na administração, manutenção e monitoramento remoto de *hosts*.

O problema envolve a questão da sigilosidade dos dados que trafegam através da rede como dados de autenticação e os dados propriamente ditos.

Os serviços de *FTP* e *telnet* fazem autenticação sem criptografia e transferem os dados sem criptografia também o que pode causar problemas de vazamento ou roubo de informações sigilosas da empresa. Os serviços remotos dos sistemas *Unixes* oferecem os mesmos serviços, porém com as mesmas falhas de segurança relativas ao tráfego de dados de forma não criptografada.

Uma solução seria a criação de túneis criptografados utilizando a *OpenSSL*, uma camada segura de criptografia para redes. Outra solução bastante utilizada seria o protocolo SSH, que além de ser bem mais simples é uma forma bastante segura de transferência e acesso remoto. Para resolvermos este tipo de problema utilizaremos o *OpenSSH* que é uma versão livre do protocolo *SSH*. A utilização do OpenSSH consta no apêndice G – *Utilização do Open SSH*.

3.3.3 – Prevenção de tráfego http

Esta seção dedica-se à proteção do servidor *web apache* na forma de configuração preventiva e do software *proxy squid* para realizar a filtragem de conteúdo *http* para os clientes da rede interna.

3.3.3.1 – Segurança do servidor web Apache

A segurança do servidor *web apache* é descrita no apêndice C - Controle de acesso no apache e recomendações de segurança para servidores *web* externos estão disponíveis no ítem 3.3.3.1.1 - *Servidores web externos*, em ZWICKY (2000, capítulo 15, p. 395) e em CONECTIVA 2 (2001, capítulo 8, p. 115), além de existirem diversas referências sobre o assunto disponíveis na literatura sobre segurança.

3.3.3.1.1 – Servidores web externos

Máquinas provendo serviços para a rede externa e provendo serviços não seguros, como no caso de um servidor *web*, não devem fornecer recursos à rede interna e devem utilizar o mínimo de recursos externos a ela, como um

servidor de banco de dados, por exemplo.

Como muitas vezes se torna necessário utilizar um servidor *web* voltado para atender às requisições vindas da rede externa, utiliza-se a abordagem de ZWICKY (2000, p.249 e p.250) de se utilizar *máquinas vítima* para hospedar apenas a página destinada ao acesso vindo da rede externa e, caso haja necessidade, uma base de dados local configurada apenas para receber conexões da máquina local onde o servidor *web* se encontra instalado.

A segurança deste tipo de *host* deve ser bastante cuidadosa em relação à segurança para minimizar todos os riscos possíveis. Servidores *web* que provem conteúdo externo devem ter *backups* freqüentes e estarem sempre ser verificados e atualizados, porém estar sempre prontos para serem substituídos ou reparados, pois sua exposição constante o torna amplamente explorado.

Estes servidores devem estar localizados no segmento da rede de perímetro, entre o *firewall* interno e o *firewall* externo, tendo seu tráfego permitido no roteador externo apenas e negado no roteador interno, devendo negar acesso a todas as outras máquinas que não sejam o host de bastião externo e, caso necessário, a uma base de dados com um controle de regras bem definido e cuidadosamente implementado de acordo com o banco de dados e os recursos dos quais o sistema *web* se utiliza.

Na rede de testes colocou-se apenas um servidor *web* no firewall como exemplo de configuração para controle de acesso do *apache* e pela falta de recursos para investir em uma nova máquina e implementá-la no segmento externo o que é uma limitação bastante comum nas PMEs.

3.3.3.2 – Implementando serviço proxy com squid

O software *squid* é um software *proxy* com características de *cache http* para *hosts* clientes. O *proxy* é como um procurador de endereços *http*, onde os softwares de internet são configurados para realizar suas requisições através do servidor *proxy*.

O squid é um *proxy* bastante configurável, permitindo os mais diversos tipos de filtragem de conexões do tráfego http como endereço IP de origem ou destino, domínios de destino, portas TCP de origem e de destino, etc.

O *proxy* foi instalado no sistema *firewall* no dia 29/11/2005 para que possa ser evitado o tráfego realizado através de roteamento e assim desabilitar o encaminhamento de pacotes IP no kernel do linux. Este procedimento evita que o firewall faça roteamento e que algum usuário na rede externa possa utilizar o firewall como gateway para chegar até a rede interna. A instalação e configuração do squid é descrita no apêndice F – Instalação e configuração do proxy-cache *squid.*

3.4 – Utilização do software antivírusclamAV

O *clamav* é um software antivírus disponibilizado para os usuários de software livre, podendo ser utilizado para verificar arquivos à procura de vírus. O software é capaz de verificar por vírus em arquivos comprimidos e pode ter

sua base de dados de vírus atualizada através da internet. O sotware pode ser obtido em www.clamav.com na seção *downloads*. A instalação e configuração do clamAV encontra-se no apêndice G – *Instalação do antivírus clamAV.*

Após a instalação basta utilizar o comando *clamscam* para poder utilizar-se de um antivírus, bastando passar como parâmetro do comando o diretório no qual deve-se procurar por arquivos infectados com vírus. O *clamAV* já disponibiliza arquivos de teste para verificar o funcionamento do programa, vamos procurar nos arquivos de instalação com o comando:

```
#clamscam /home/clamav/clamav-0.87.1/*/*
```

Os arquivos contendo uma saída *OK* estão limpos e os que apresentarem a saída com *FOUND* foram encontrados vírus ou algum tipo de *malware*. O exemplo a seguir é a saída do comando acima:

```
...
/home/clamav/clamav-0.87.1/sigtool/vba.o: OK
/home/clamav/clamav-0.87.1/test/README: OK
/home/clamav/clamav-0.87.1/test/clam-error.rar: RAR module failure
/home/clamav/clamav-0.87.1/test/clam.cab: ClamAV-Test-File FOUND
/home/clamav/clamav-0.87.1/test/clam.exe: ClamAV-Test-File FOUND
/home/clamav/clamav-0.87.1/test/clam.exe.bz2: ClamAV-Test-File FOUND
/home/clamav/clamav-0.87.1/test/clam.rar: ClamAV-Test-File FOUND
/home/clamav/clamav-0.87.1/test/clam.zip: ClamAV-Test-File FOUND
/home/clamav/clamav-0.87.1/test/mbox/debugm.c: OK

----------- SCAN SUMMARY -----------
Known viruses: 41430
Engine version: 0.87.1
Scanned directories: 15
Scanned files: 608
Infected files: 5
Data scanned: 17.79 MB
Time: 7.502 sec (0 m 7 s)
```

Para verificar os e-mails basta passar como parâmetro o diretório onde os arquivos de e-mail dos usuários estão, no caso do *slackware* é na pasta */var/spool/mail*, logo o comando seria:

```
#clamscam /var/spool/mail/* /var/spool/mail/*/*
```

No caso de diretórios muito grandes deve-se filtrar a saída utilizando-se o comando:

```
#clamscam /var/spool/mail/* /home/*/* | grep FOUND
```

Podendo ainda ser colocada em um arquivo para análise posterior:

```
#clamscam /var/spool/mail/* /home/*/* | grep FOUND > /home/usuario/clamscan.log
```

Agora já podemos utilizar os recursos de um antivírus para varrer os diretórios importantes do sistema, dos usuários e dos e-mails.

CONCLUSÃO

Ao longo do curso deste trabalho verificou-se as necessidades de TI e os recursos e serviços utilizados pelas PMEs e utilizou-se destes serviços em uma rede de testes para avaliar o nível de segurança oferecido através de uma ferramenta de detecção de vulnerabilidades (*nessus*) e aplicou-se algumas medidas preventivas e corretivas para minimizar os riscos relativos a estes serviços segundo os critérios apresentados nos relatórios do *nessus*.

Após a implementação, verificação e correção dos sistemas chegou-se à conclusão de que os serviços e sistemas testados baseados em tecnologias de software livre utilizados por pequenas de pequeno e médio porte são bastante seguros e estão em conformidade com níveis mínimos de segurança para a utilização em ambientes empresariais de pequeno e médio porte segundo as análises realizadas com a ferramenta *nessus*.

Esta referência baseada nas análises realizadas com o *nessus* pode servir de base para técnicos, estudantes, prestadores de serviços das PMEs que estão se familiarizando ou estejam em planos de implementação de redes baseadas em tecnologias de software livre.

REFERÊNCIAS BIBLIOGRÁFICAS

ANÔNIMO. **Segurança máxima para linux.** São Paulo: Editora Campus, 2000. 476 p.

CONECTIVA 1, Equipe. **Segurança de Redes:** Firewall. Curitiba: Conectiva SA, 2001. 174 p.

CONECTIVA 2, Equipe. **Segurança de Redes:** Ferramentas e serviços. Curitiba: Conectiva SA, 2001. 146 p.

GOMES, Christian Lyra et al. **Guia do servidor Conectiva Linux 7.0**. Segunda edição. Curitiba: Conectiva SA, 2001. 347 p.

HICKS, Allan et al. **Slackware Linux Essentials**. 2. ed. Disponível em: <http://www.slackbook.org/html/index.html>. Acesso em: 15 agosto 2005.

NETFILTER, team. **Netfilter/iptables project homepage**. Acessado em 10/9/2005 disponível em < http://www.netfilter.org/projects/iptables/index.html>.

SCAMBRAY, Joel; McCLURE, Stuart; KURTZ George. **Hackers Expostos:** segredos e soluções para a segurança de redes. 2. ed. São Paulo: Makron Books, 2001. 694 p.

VISOLVE. **Squid Configuration manual 2.4**. acessado em 20/11/2005. disponível em <http://squid.visolve.com/squid/squid24s1/contents.htm>.

VOLKERDING, Patrick. **Slackware GNU/Linux versão 10.2**. obtido 25/9/2005 em <www.slackware.com>.

ZWICKY, Elizabeth D.;COOPER Simon; CHAPMAN, D. Brent. **Construindo Firewalls para a Internet**. 2. ed. (publicação em português). Rio de Janeiro: Campus, 2000. 889 p.

APÊNDICES

APÊNDICE A - Configuração do mysql

Aqui mostramos a configuração realizada no *mysql* para utilizar-se em conjunto com o *snort* e com o ACID.

```
#su mysql
$mysql_install_db
Installing all prepared tables
Fill help tables

To start mysqld at boot time you have to copy support-files/mysql.server
to the right place for your system

PLEASE REMEMBER TO SET A PASSWORD FOR THE MySQL root USER !
To do so, start the server, then issue the following commands:
/usr/bin/mysqladmin -u root password 'new-password'
/usr/bin/mysqladmin -u root -h juarezdc password 'new-password'
See the manual for more instructions.

You can start the MySQL daemon with:
cd /usr ; /usr/bin/mysqld_safe &

You can test the MySQL daemon with the benchmarks in the 'sql-bench' directory:
cd sql-bench ; perl run-all-tests

Please report any problems with the /usr/bin/mysqlbug script!

The latest information about MySQL is available on the web at
http://www.mysql.com
Support MySQL by buying support/licenses at https://order.mysql.com
```

```
$exit
#cd /etc/rc.d
#chmod 700 rc.mysqld
#./rc.mysqld start
Starting mysqld daemon with databases from /var/lib/mysql
#mysqladmin -u root password senha-root
```

A seguir devemos criar uma base de dados no *mysql* para que o *snort* coloque seus dados e o *ACID* consiga gerar seus relatórios:

```
#mysql -u root -p
Enter password:

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2 to server version: 4.1.14

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> create database snort;
   Query OK, 1 row affected (0.00 sec)

mysql>grant insert, select on snort.* to snort@localhost identified by “senha-snort”;
Query OK, 0 rows affected (0.00 sec)

mysql>grant insert, select, delete, update, create on snort.* to acid@localhost identified by "acid-senha";
Query OK, 0 rows affected (0.00 sec)

mysql>quit
```

A seguir utilizaremos o *script* do *snort* para criação das tabelas do *snort* no *mysql* no diretório dos arquivos de instalação do *snort* no diretório *schemas*. Como no nosso sistema colocamos os arquivos de instalação na pasta */opt/snort-2.4.2* utilizaremos este diretório como exemplo, caso tenha colocado os arquivos em outro local modifique o local dos arquivos de instalação:

```
#cd /opt/snort-2.4.2/schemas
#mysql -u root -p snort < create_mysql
```

O comando acima solicitará a senha do usuário *root* do *mysql*, insira a

senha no prompt e o *script* de instalação será executado.

A seguir é necessário descomentar a seguinte linha do arquivo de configuração do *snort* em */etc/snort/snort.conf*:

```
#output database: log, mysql, user=snort password=senha-snort dbname=snort host=localhost
```

APÊNDICE B - Instalação dos arquivos do *ACID* no *apache*

Para instalarmos os arquivos do *ACID* no apache devemos baixar os arquivos e dependências do *ACID*:

- *ACID* pode ser obtido em: <http://acidlab.sourceforge.net/acid-0.9.6b23.tar.gz >;
- A biblioteca *JPGraph* é necessária como dependência e pode ser obtida em: <http://members.chello.se/jpgraph/jpgdownloads/jpgraph-1.20.tar.gz>;
- *ADODB* é uma camada de abstração para acesso aos bancos de dados para scripts *perl* e *php*, o *ADODB* pode ser obtido em : <http://ufpr.dl.sourceforge.net/sourceforge/adodb/adodb468.tgz>.

Primeiro vamos colocar os arquivos da pasta */var/www/htdocs* em um diretório à parte:

```
#cd /var/www/htdocs
#mkdir old
# mv *.* old
#mv manual old
```

Devemos iniciar com a instalação da biblioteca *JPGraph*:

```
#cd /opt
# tar xvfz /root/jpgraph-1.20.tar.gz
#cp  jpgraph-1.20/* /var/www/htdocs
```

A seguir faremos o mesmo com o pacote do *ADODB*:

```
#cd /opt
# tar xvfz /root/adodb468.tgz
# cp -r adodb/* /var/www/htdocs
```

E então faremos os mesmo processo para o *ACID*:

```
#cd /opt
#tar xvfz acid-0.9.6b23.tar.gz
#cp acid/* /var/www/htdocs
```

As configurações do *apache* foram retiradas do capítulo 8 de CONECTIVA 2 (2001, p.115).

APÊNDICE C - Controle de acesso no apache

Implementamos controle de acesso para o *apache* conforme abordado em CONECTIVA 2 (2001, p.117) para o diretório */var/www/htdocs* que contem os arquivos do *ACID*. Primeiramente criamos o arquivo *.htaccess* no diretório */var/www/htdocs* com o comando *touch*:

```
#cd /var/www/htdocs
#touch .htaccess
```

E, a seguir, inserimos as seguintes informações no arquivo */var/www/htdocs/.htaccess*:

```
Deny from All
Allow from 192.168.1.0/255.255.255.0
AuthName "Somente para internos"
AuthType Basic
AuthUserFile /var/www/htdocs/.htpasswd-admin
Require valid-user
Satisfy Any
```

As linhas *Deny from All* e *Allow from 192.168.1.0/255.255.255.0* limitam o acesso desta pasta do apache a apenas à rede interna.

A linha *AuthUserFile /var/www/htdocs/.htpasswd-admin* especifica o arquivo de especificação de nome de usuário e senha de acesso para usuários que não estejam na rede local. Esta configuração de acesso foi baseada em CONECTIVA 2 (2001) capítulo 8.

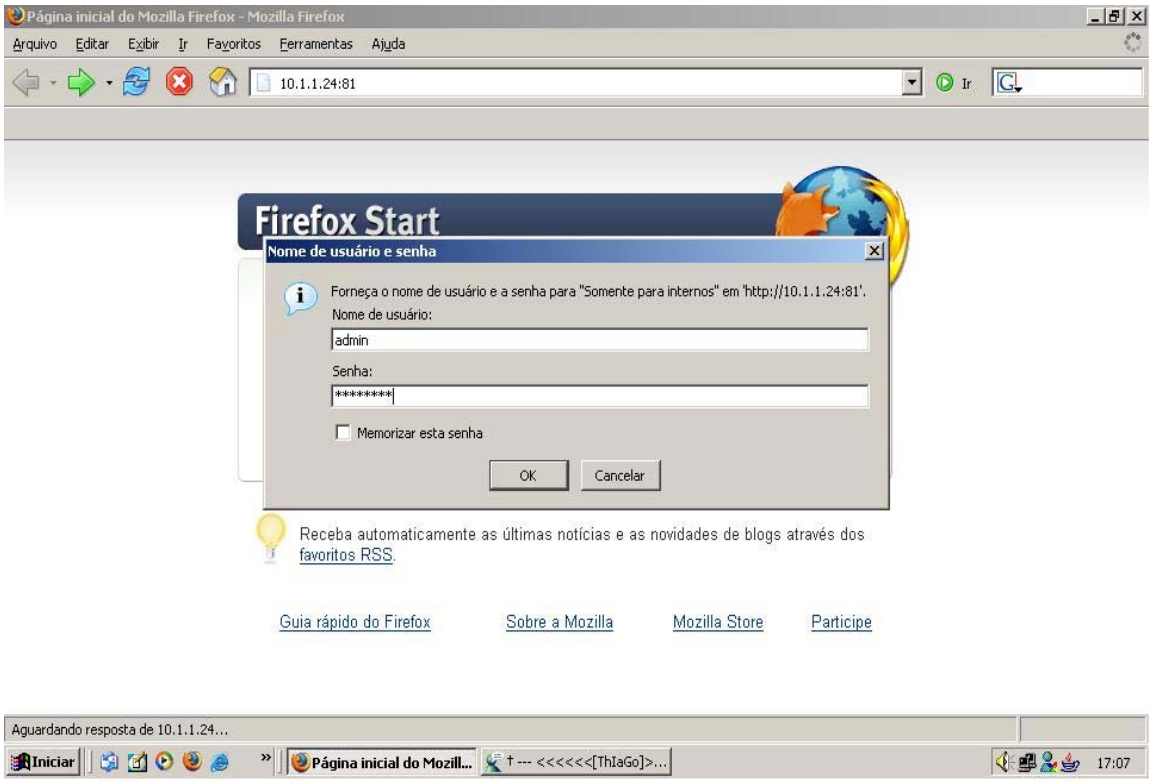
Para criarmos o arquivo de senhas, execute no console o comando *htpasswd*:

```
#cd /var/www/htdocs
#htpasswd -c -d .htpasswd-admin admin
New password:
Re-type new password:
Adding password for user admin
```

Agora deve-se reiniciar o servidor *web* para que as alterações tenham efeito com os comandos:

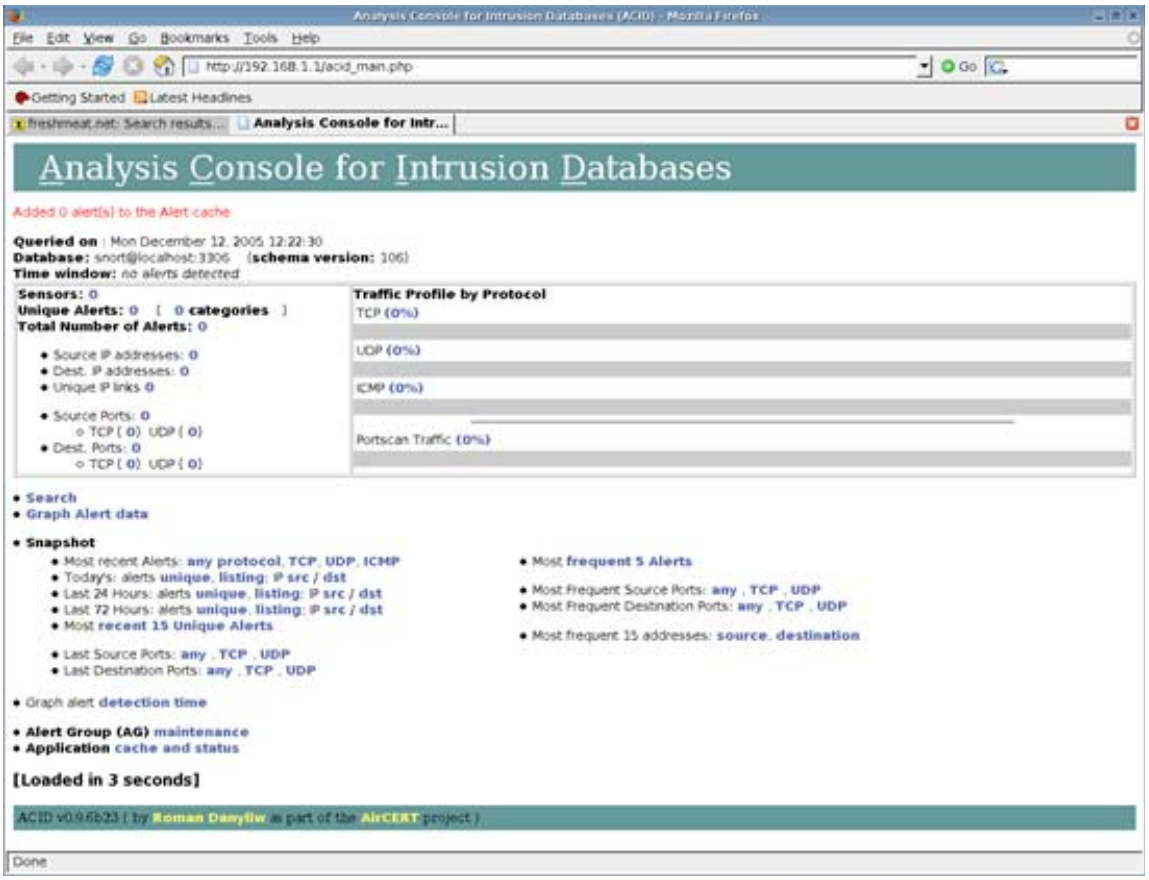
```
#cd /etc/rc.d
#./rc.httpd restart
```

Agora os usuários que tentarem acessar o servidor *web* fora da rede interna receberão uma janela de autenticação para que possam se autenticar. Fizemos isto colocando o computador cliente no segmento da rede externa com endereço IP 10.1.1.5 e tentamos acesso ao servidor *web* e recebemos a seguinte janela de autenticação:



janela de login de acesso para o *apache*.

No caso de cancelarmos a autenticação, será enviada uma página de negação de acesso. No caso de autenticação positiva ou de se estar na rede interna, como no caso abaixo, teremos a tela inicial do *ACID*:



página inicial do ACID

Como vamos utilizar apenas para o acesso interno, podemos definir a seguinte linha no arquivo de configuração */etc/apache/httpd.conf* no *firewall*:

Listen 192.168.1.1:80

Esta linha define que o *apache* deverá aguardar conexões apenas na interface da rede interna. A saída do comando *nmap* nas interfaces 10.1.1.24 e 192.168.1.1 (e mais ainda a interface de *loopback,* caso se esteja na máquina *firewall*) mostrará que o *apache* agora está apenas na interface interna (192.168.1.1). Para constatarmos esta modificação, basta verificamos as portas

TCP das interfaces do sistema *firewall*. Por exemplo:

```
#nmap 192.168.1.1;  nmap 10.1.1.24;  nmap 127.0.0.1

Starting nmap 3.93 ( http://www.insecure.org/nmap/ ) at 2005-12-12 12:26 BRST
Interesting ports on 192.168.1.1:
(The 1663 ports scanned but not shown below are in state: closed)
PORT     STATE SERVICE
22/tcp   open  ssh
53/tcp   open  domain
80/tcp   open  http
3128/tcp open  squid-http
3306/tcp open  mysql

Nmap finished: 1 IP address (1 host up) scanned in 1.560 seconds

Starting nmap 3.93 ( http://www.insecure.org/nmap/ ) at 2005-12-12 12:26 BRST
Interesting ports on 10.1.1.24:
(The 1666 ports scanned but not shown below are in state: closed)
PORT     STATE SERVICE
53/tcp   open  domain
3306/tcp open  mysql

Nmap finished: 1 IP address (1 host up) scanned in 1.411 seconds

Starting nmap 3.93 ( http://www.insecure.org/nmap/ ) at 2005-12-12 12:26 BRST
Interesting ports on localhost (127.0.0.1):
(The 1665 ports scanned but not shown below are in state: closed)
PORT     STATE SERVICE
53/tcp   open  domain
953/tcp  open  rndc
3306/tcp open  mysql

Nmap finished: 1 IP address (1 host up) scanned in 1.304 seconds
```

Agora, o servidor *web* está um pouco realizando autenticação em casos de clientes que se conectam da rede externa, esperando conexões apenas na interface interna. O tipo de autenticação aqui utilizado é considerado inseguro para a utilização em uma rede pública como a internet devendo, nestes casos, utilizar técnicas de criptografia como o *Secure Socket Layer* também conhecido por *SSL* e não foi abordado por este trabalho.

Algumas mudanças no arquivo de configuração para melhorar a

segurança foram:

- MinSpareServers 1: limita o número de servidores mínimos para 1;
- MaxSpareServers 3: limita o número máximo de servidores para 3;
- StartServers 1: Inicia apenas 1 servidor;
- MaxClients 3: configura o número máximo de clientes;
- Listen 192.168.1.1:80: faz com que o servidor aguarde conexões apenas na interface da rede interna;
- User nobody: executa o *apache* com o usuário *nobody*;
- Group nobodyexecuta o *apache* com o grupo *nobody*;
- ServerAdmin juarezdc@hotmail.com: e-mail do administrador, para que os usuários possam ter a quem enviar as mensagens de erro exibidas;
- ServerName www.juarezdc.org: configura o nome da máquina para ser exibida nas páginas de erro;
- ScriptAlias /cgi-bin/ “/var/www/chi-bin” e a seção <Directory “/var/www/cgi-bin”>: Esta linha e esta seção foram comentadas para que não se possa utilizar o mecanismo *cgi-bin* do *apache*;
- AllowOverride All: permite sobrescrita do arquivo de acesso *.htaccess* constante nos diretórios locais das páginas mapeadas em cada entrada <*Directory*> no arquivo *httpd.conf*;
- Include /etc/apache/mod_php.conf: Utilizamos o módulo *php* para instalar e executar o *ACID*;

APÊNDICE D – Instalação e configuração do *snort*

O snort está disponível em <http://www.snort.org/dl/current/snort-2.4.3.tar.gz>, após baixá-lo deve-se descomprimí-lo, baixamos no diretório do superusuário e vamos descompactá-lo no diretório */opt* como mostrado no comando abaixo:

```
#cd /opt
#tar xvfz /root/snort-2.4.3.tar.gz
```

A seguir o software será descomprimido e será criada uma pasta, dentro do diretório */opt*, com o software, chamada de *snort-2.4.3*.

Para a instalação mais básica devemos prosseguir da seguinte forma:

```
#cd /opt/snort-2.4.3
#./configure
#make
#make install
```

OBSERVAÇÕES: Para utilizar-se o *snort* em conjunto com o *mysql* deve-se executar o *script* de configuração chamado *configure*, constante no diretório dos arquivos de instalação do *snort*, com o parâmetro *–with-mysql*. O comando para a configuração da instalação do *snort* para utilização com o *mysql* deve ser:

```
#./configure
#make; make install
```

Para utilizar o *snort* em conjunto com o *mysql* deve-se retirar o caractere # da seguinte linha do arquivo */etc/snort/snort.conf*:

```
#output database: log, mysql, user=snort password=senha-snort dbname=snort host=localhost
```

Caso todos os requisitos tenham sido satisfeitos pelo *script configure* o *snort* está agora compilado e instalado.

Devemos agora criar um diretório para os arquivos de configuração do *snort*. Criaremos o diretório */etc/snort* e copiamos o conteúdo do diretório */opt/snort-2.4.3/etc* para o diretório */etc/snort* com o comando abaixo:

```
#mkdir /etc/snort
#cp /opt/snort-2.4.3/etc/* /etc/snort
```

O próximo passo é baixar as regras de detecção de intrusão no endereço <http://www.snort.org/pub-bin/downloads.cgi/Download/vrt_pr/snortrules-pr-2.4.tar.gz>. Estas regras são de domínio público e não requerem registro na página *web* do *snort*, sendo verificadas pela equipe do *snort.*

Baixamos as regras no diretório do *superusuário* e descomprimindo-as em */etc/snort/rules* com os comandos:

```
#mkdir /etc/snort/rules
#cd /etc/snort/rules
#tar xvfz /root/snortrules-pr-2.4.tar.gz
```

Basta aguardar a descompressão da regras e configurar o caminho no

arquivo de configuração como */etc/snort/rules/rules*.

O próximo passo será criar o diretório de registros do *snort* em */var/log/snort* com os comandos a seguir:

```
#mkdir /var/log/snort
```

O próximo passo é a configuração do arquivo de configuração em */etc/snort/snort.conf*. Comentaremos no próprio arquivo de configuração as linhas que foram alteradas. O arquivo de configuração ficou como descrito no apêndice C.

O próximo passo é iniciar o software com o comando:

```
#snort -A full -devyD -c /etc/snort/snort.conf -l /var/log/snort/
```

Onde as opções utilizadas foram:

- "A full": modo de alerta do *snort* tido como total (*full*);
- "d": exibe as informações da camada de aplicação;
- "e": exibe informações de cabeçalho da segunda camada (*ARP*);
- "v": modo mais informativo;
- "y": imprime o ano nos registros (*logs*) do *snort*;
- "D": roda o *snort* em modo de serviço (*daemon mode*);
- "c /etc/snort/snort.conf": indica o arquivo de configuração a ser lido pelo software;
- "l /var/log/snort": especifica o diretório de registros (*logs*) para /var/log/snort.

Para verificar se o *snort* está em execução, basta executar o comando:

```
# ps ax | grep snort
 7776 ?        Ss     0:02 snort -A full -devyD -c /etc/snort/snort.conf -l /var/log/snort/
```

Caso o comando *ps ax | grep snort* retorne o comando digitado anteriormente o *snort* estará em execução.

Agora para que o comando execute a cada inicialização deve ser incluída a linha do comando no arquivo de inicialização */etc/rc.d/rc.local*. Pode-se digitar o comando ou teclar a seta para cima do teclado até encontrar a linha do comando e modificar a linha para:

```
#echo “snort -A full -devyD -c /etc/snort/snort.conf -l /var/log/snort/” >> /etc/rc.d/rc.local
```

Para verificar a existência da linha no arquivo de inicialização local utiliza-se o comando *cat*:

```
# cat /etc/rc.d/rc.local
#!/bin/sh
#
# /etc/rc.d/rc.local:  Local system initialization script.
#
# Put any local setup commands in here:

#modificando o comportamento de resposta a ping em broadcast para rejeitá-los
sysctl net.ipv4.icmp_echo_ignore_broadcasts=1

#habilita syncookies no kernel, técnica contra o ataque “syn flood”
sysctl net.ipv4.tcp_syncookies=1

#inicializando ip_forward pelo kernel, caso utilize-se roteamento.
#caso for utilizar proxy, esta opção deve ser 0
sysctl net.ipv4.ip_forward=0
```

```
#inicializacao do snort em modo NIDS
/usr/local/bin/snort -A full -vedD -c /etc/snort/snort.conf -l /var/log/snort

#mudando o endereco de origem com SourceNAT, deixando a origem do pacote
#ao sair com o endereco ip de saida do firewall.
#Regra válida para o caso de roteamento.
# se for utilizar proxy comente esta linha.
#iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to 10.1.1.24

#iniciando servico de DHCP na interface da rede interna
dhcpd eth1 -cf /etc/dhcpd.conf

#Loga sobre IPs vindos da interface externa com endereço de origem origem na rede interna
iptables -A INPUT -i eth0 -s 192.168.1.0/24 -j LOG
#Descarta os datagramas IP acima.
iptables -A INPUT -i eth0 -s 192.168.1.0/24 -j DROP

#bloqueando as portas abertas do servidor e da rede interna vindas da interface externa (eth0)
#portas relativas à smb/CIFS
iptables -A INPUT -i eth0 -p tcp --dport 139 -j DROP
iptables -A INPUT -i eth0 -p tcp --dport 445 -j DROP
iptables -A INPUT -o eth0 -s 192.168.1.0/24 -p udp –dport 138 -j REJECT
iptables -A INPUT -o eth0 -s 192.168.1.0/24 -p udp –dport 445 -j REJECT

#porta relativa ao servidor gráfico X11 dos Unixes
iptables -A INPUT -i eth0 -p tcp --dport 6000 -j DROP

#portas encontradas com o software nessus e nmap
iptables -A INPUT -i eth0 -p tcp --dport 1025 -j DROP
iptables -A INPUT -i eth0 -p tcp --dport 1024 -j DROP

#nega tráfego http de entrada vindos da rede interna.
iptables -A INPUT -d 192.168.1.0/24 -i eth0 -p tcp --dport 80 -j DROP

#bloqueando conexoes externas para o squid.
iptables -A INPUT -i eth0 -d 10.1.1.24 -p tcp --dport 3128 -j DROP
```

Observação: as linhas contendo #, após a linha que contém o comando cat */etc/rc.d/rc.local,* são comentários do arquivo *rc.local*.

A saída do comando acima nos mostra que foi incluída a linha no arquivo de inicialização *rc.local*. Quando o computador iniciar novamente o *snort* estará executando em modo *NIDS* (modo de detecção de intrusão) e gerando

registros de conexões, alertas e de verificação de portas TCP por outros hosts em */var/log/snort*.

O *snort* também pode servir para ler os arquivos de *log*. Para isto deve-se utilizar a sintaxe:

```
#snort -r /var/log/snort/snort.log.1134069996
```

É exibido na tela todo o tráfego capturado pelo arquivo de *log* naquela sessão de execução do *snort*. É importante verificar regularmente os arquivos de alerta e de varredura de portas, configurados aqui para residirem em */var/log/snort* sendo os arquivos, respectivamente, *alert* e *portscan.log*. Logo o caminho do arquivo de alerta é */var/log/snort/alert* e o caminho do arquivo de registros de tentativas de varreduras de portas TCP é */var/log/snort/portscan.log*.

Para visualizar os arquivos de registros de alerta e de varredura de portas basta utilizar um comando de leitura ou visualização de texto. Exemplo com o comando *cat*:

```
#cat /var/log/snort/portscan.log
#cat /var/log/snort/alert
```

Agora, o sistema operacional já possui um sistema de detecção de intrusão de rede implementado com o *snort*.

Caso você perceba muitos registros de tentativas de ataques através dos registros do *snort* pode-se bloquear as tentativas através do *iptables*.

Verifique o registro de alerta e de varredura, caso haja tentativas recentes,

ou acontecendo caso o arquivos esteja sendo atualizado, e bloqueie o endereço IP do possível atacante com o comando:

```
#iptables -A INPUT -i eth0 -s <ip_de_origem> -j DROP
```

APÊNDICE E – Instalação e configuração do *nessus*

O software pode ser obtido através do endereço <http://www.nessus.org/download/>. Deve-se ler os termos de utilização do software e concordar com ela para que se possa realizar o *download* do arquivo de instalação, após isto deve-se preencher o campo de registro com um endereço de e-mail para que se possa receber uma chave de registro que permitirá atualizar os *plug-ins* de verificação de vulnerabilidades.

Após baixá-lo deve-se executá-lo, baixamos no diretório do superusuário e o executamos como mostrado no comando abaixo:

```
#chmod 700 nessus-installer-2.2.6.sh
#./nessus-installer-2.2.6.sh
```

A seguir serão exibidas as telas de configuração da instalação do nessus:

```
---------------------------------------------------
                NESSUS INSTALLATION SCRIPT
---------------------------------------------------

Welcome to the Nessus Installation Script !

This script will install Nessus 2.2.6 (STABLE) on your system.

Please note that you will need root privileges at some point so that
the installation can complete.

Nessus is released under the version 2 of the GNU General Public License
(see http://www.gnu.org/licences/gpl.html for details).

Most of the Nessus plugins are released under the Tenable License
(see http://www.nessus.org/plugins/DirectFeed.pdf for details)
```

To get the latest version of Nessus, visit http://www.nessus.org

Press ENTER to continue

A instalação descomprimirá o arquivo após o usuário teclar e*nter*:

x - creating lock directory
x - extracting nessus.tar.gz (binary)
x - now extracting this archive
x – done

A seguir, a instalação solicitará o diretório onde o *nessus* deve ser instalado. As opções que aparecem entre colchetes são tidas como padrão, podendo ser alteradas, caso desejado, e mantendo-se como padrão no caso de ser teclado apenas Enter.

Nessus installation : installation location

Where do you want the whole Nessus package to be installed ?
[/usr/local]

Na tela de solicitação do diretório de instalação tecle *enter* para instalar no diretório padrão que é */usr/local*. O software irá então compilar e instalar o *nessus*:

```
-------------------------------------------------
                Nessus installation : Ready to install
-------------------------------------------------


Nessus is now ready to be installed on this host.
The installation process will first compile it then install it

Press ENTER to continue

x - Compiling the libraries
x -- Configuring the sources for your system
x -- Uninstalling any previous version of Nessus
+ rm -f /usr/local/bin/nasl
+ rm -f /usr/local/bin/nasl-config
+ rm -f /usr/local/bin/nessus
+ rm -f /usr/local/bin/nessus-config
+ rm -f /usr/local/bin/nessus-build
+ rm -f /usr/local/bin/nessus-mkrand
+ rm -f /usr/local/bin/nessus-mkcert-client
+ rm -f /usr/local/sbin/nessus-adduser
+ rm -f /usr/local/sbin/nessus-rmuser
+ rm -f /usr/local/sbin/nessusd
+ rm -f /usr/local/sbin/nessus-update-plugins
+ rm -f /usr/local/sbin/nessus-mkcert
+ rm -f /usr/local/sbin/nessus-check-signature
+ rm -rf /usr/local/include/nessus
+ rm -f '/usr/local/lib/libhosts_gatherer.*'
+ rm -f '/usr/local/lib/libnasl.*'
+ rm -f '/usr/local/lib/libnessus.*'
+ rm -f '/usr/local/lib/libpcap-nessus.*'
+ rm -rf /usr/local/lib/nessus
+ rm -f /usr/local/man/man1/nasl-config.1
+ rm -f /usr/local/man/man1/nasl.1
+ rm -f /usr/local/man/man1/nessus-build.1
+ rm -f /usr/local/man/man1/nessus-config.1
+ rm -f /usr/local/man/man1/nessus.1
+ rm -f /usr/local/man/man1/nessus-mkrand.1
+ rm -f /usr/local/man/man1/nessus-mkcert-client.1
+ rm -f /usr/local/man/man8/nessus-mkcert.8
+ rm -f /usr/local/man/man8/nessus-adduser.8
+ rm -f /usr/local/man/man8/nessus-rmuser.8
+ rm -f /usr/local/man/man8/nessus-update-plugins.8
+ rm -f /usr/local/man/man8/nessusd.8
+ test -n ''
+ set +x
x -- Compiling
x -- Installing
x - Compiling the NASL interpretor
x -- Configuring the sources for your system
x -- Compiling
nasl_func.c: In function `insert_nasl_func':
nasl_func.c:100: warning: passing arg 4 of `qsort' from incompatible pointer type
nasl_init.c: In function `init_nasl_library':
```

```
nasl_init.c:430: warning: assignment discards qualifiers from pointer target type
x -- Installing
x - Compiling the core
x -- Configuring the sources for your system
x -- Compiling
monitor_dialog.c: In function `idle_socket':
monitor_dialog.c:108: warning: `soc' might be used uninitialized in this function
x -- Installing
x - Compiling the plugins
x -- Configuring the sources for your system
x -- Compiling
x – Installing
```

Após a compilação e instalação deve-se entrar com o código de ativação enviado para o e-mail, que foi cadastrado no processo de *download*, para registro e posterior atualização dos *plug-ins* de detecção de vulnerabilidades do *nessus*. Coloque o número de ativação e tecle Enter:

```
-------------------------------------------------------
                Nessus installation : Registration
-------------------------------------------------------

If you want to get a full plugin feed from http://www.nessus.org, you need
to register at http://www.nessus.org/plugins/.

If you already registered and received an activation code, please enter
it below if you want to activate it now :

Activation code : 9106-9953-7FDC-1B02-04F0
```

A instalação está agora finalizada, basta teclar enter como mostra a última tela da instalação, que nos dá algumas recomendações finais:

```
-------------------------------------------------------
                Nessus installation : Finished
-------------------------------------------------------

Congratulations ! Nessus is now installed on this host
```

```
. Create a nessusd certificate using /usr/local/sbin/nessus-mkcert
. Add a nessusd user use /usr/local/sbin/nessus-adduser
. Start the Nessus daemon (nessusd) use /usr/local/sbin/nessusd -D
. Start the Nessus client (nessus) use /usr/local/bin/nessus
. To uninstall Nessus, use /usr/local/sbin/uninstall-nessus

. Remember to invoke '/usr/local/sbin/nessus-update-plugins'
  periodically to update your list of plugins

. A step by step demo of Nessus is available at :
      http://www.nessus.org/demo/


Press ENTER to quit
```

As recomendações são para criar um certificado para o servidor, adicionar um usuário para autenticar-se no servidor do *nessus*, iniciar o serviço do *nessus* (servidor), executar um software cliente para o *nessus* e por fim algumas sugestões de como remover o *nessus* e manter-se atualizado com os *plug-ins*.

A seguir, cria-se um usuário no *nessus* com o comando *nessus-adduser*, seguindo os passos solicitados e escolhe-se uma senha. Por exemplo:

```
# nessus-adduser
Using /var/tmp as a temporary file holder

Add a new nessusd user
----------------------


Login : teste
Authentication (pass/cert) [pass] : pass
Login password :
Login password (again) :
Passwords do not match !
Login password :
Login password (again) :

User rules
----------
nessusd has a rules system which allows you to restrict the hosts
```

```
that teste has the right to test. For instance, you may want
him to be able to scan his own host only.

Please see the nessus-adduser(8) man page for the rules syntax

Enter the rules for this user, and hit ctrl-D once you are done :
(the user can have an empty rules set)
allow localhost
deny all


Login          : teste
Password       : ***********
DN             :
Rules          :
allow all


Is that ok ? (y/n) [y] y
user added.
```

Observe que o *nessus* permite escolher entre autenticação com certificado, ou com senha, seno que optamos por escolher a senha. O *nessus* permite ainda escolhermos regras de acesso para o usuário, dizendo qual ou quais hosts o usuário pode verificar e para este usuário escolhemos permissões totais com a entrada *allow all* acima.

APÊNDICE F – Instalação e configuração do proxy-cache *squid*

O *proxy-cache squid* é licenciado como software livre e possui binários para várias distribuições, inclusive para a distribuição *slackware*. A página oficial do *squid* é <www.squid-cache.org> onde seguimos os atalhos até os pacotes binários específicos para o *slackware* em: <http://www.merlinobbs.net/squid/squid-2.5.STABLE12-i386-1.tgz>.

Seguimos as instruções do *sítio* <http://www.merlinobbs.net/squid> para a instalação. Instalamos o *squid* no host de bastião para filtrar o tráfego de internet dos usuário da rede interna e assim desativar o roteamento para a rede externa ao host de bastião como recomenda ZWICKY (2000, p. 249). Para instalar o pacote do *squid* no *slackware* basta entrar com o seguinte comando no console:

```
#installpkg squid-2.5.STABLE12-i386-1.tgz
```

A seguir criamos o *script* de inicialização para o *squid* em */etc/rc.d/rc.squid*:

```
#touch /etc/rc.d/rc.squid
```

E copiamos o seguinte conteúdo para o arquivo:

```
#!/bin/sh
# nome del file: /etc/rc.d/rc.squid
echo -n ' squid '
case "$1" in
start)
/usr/sbin/squid -D
;;
stop)
/usr/sbin/squid -k shutdown
;;
restart)
/usr/sbin/squid -k reconfigure
;;
*)
echo "Usage: `basename $0` {start|stop|restart}"
;;
esac
#
# EOF (c) 2005 Stefano Tagliaferri - GPL license
```

A seguir copiamos o seguinte conteúdo para o arquivo /etc/rc.d/rc.inet2:

```
#inicialização do squid
if [ -x /etc/rc.d/rc.squid ]; then
  . /etc/rc.d/rc.squid start
fi
```

Agora editamos o arquivo */etc/squid/squid.conf.default* conforme o apêndice D e renomeamos o arquivo com o comando *mv* e o arquivo *mime.conf.default* apenas renomearemos para *mime.conf*:

```
#mv /etc/squid/squid.conf.default /etc/squid/squid.conf

#mv /etc/squid/mime.conf.default /etc/squid/mime.conf
```

A seguir colocamos o arquivo */etc/rc.d/rc.squid* com permissão de execução para que o *squid* possa inicializar quando o sistema iniciar:

```
#chmod 700 /etc/rc.d/rc.squid
```

Para finalizar e para reiniciar, quando alteramos o arquivo de configuração por exemplo, utilizamos os parâmetros *stop* e *restart* respectivamente. Antes de iniciar o *squid* devemos configurar os diretórios do *squid* criá-los. Para isto adicionaremos as seguintes linhas no arquivo */etc/squid/squid.conf*:

```
cache_dir ufs /var/cache/proxy 100 16 128
cache_access_log /var/log/squid-access.log
```

A seguir criamos um usuário chamado *squid* e configuramos no arquivo *squid.conf* na linha:

```
cache_effective_user squid
```

A seguir criamos o arquivo */var/log/squid-access.log* e configuramos as permissões dos diretórios de *log* e de *cahce*:

```
#mkdir /var/log/squid
#mkdir /var/log/squid/logs
#touch /var/log/squid/squid-access.log
#touch /var/log/squid/logs/store.log
#chown squid /var/log/squid/logs/store.log
#chown squid /var/log/squid/squid-access.log
#chown -R squid /var/cache/proxy
#chown -R squid /var/log/squid
```

A seguir iniciaremos os arquivos de *cache* do *squid* com o comando:

```
#squid -z
```

Para iniciar o *squid* basta executar o *script rc.squid* com o parâmetro *start*:

```
#cd /etc/rc.d
#./rc.squid start
```

Agora basta configurar os softwares de internet, como os navegadores e outros softwares, para conectar-se através de um *proxy http* com o endereço

do *host* executando o *squid*, no caso 192.168.1.1, e com a *porta* 3128 configurada.

Os arquivos de registros são importantes e estão localizados em */var/log/squid* e são:

- squid-access.log: contém os registros de acesso *http* realizadas pelo *squid* para os usuários;
- Logs/store.log: contém os registros do que foi armazenado em *cache* pelo *squid*;
- Logs/cache.log: contém as informações de armazenamento dos arquivos em *cache* e em *swap*. É voltada ao funcionamento interno do *squid* para fins de *debug*. Este arquivo aumenta demasiadamente e deve ser limpo a intervalos regulares de no máximo 14 dias;
- Logs/squid.pid: contém o número do processo do *squid* em execução.

Os arquivos de registros não estão sendo limpos periodicamente pelo utilitário *logrotate* e após um tempo prolongado de utilização do *squid* estes arquivos começam a ficar grandes, podendo inclusive lotar a capacidade da partição. Recomenda-se limpar os arquivos de registros a cada 14 dias sendo o máximo em 21 dias para evitar problemas de disco cheio em partições pequenas. Para limpar o arquivo *cache.log*, por exemplo, basta digitar o

seguinte comando:

```
#echo “” > /var/log/squid/logs/cache.log
```

Caso tenha problemas na inicialização do *squid* pode-se iniciá-lo com parâmetros de *debug*, os quais informam o problema de inicialização do *squid*. O comando é:

```
#squid -Ncd1
```

O *proxy* já está configurado e executando. Agora podemos desabilitar o encaminhamento de pacotes IP do *kernel* do sistema *firewall* e desabilitar o serviço de tradução de nomes (*NAT*) do *iptables* e utilizarmos apenas *proxy* na interface externa, evitando assim contato direto com a rede externa. O *iptables* porém continua sendo válido para bloquear, verificar e registrar informações tanto da rede interna quanto da rede externa, não deixando de ser utilizado quando necessário.

Para desabilitar o encaminhamento no *kernel* do linux digite no sistema *firewall*:

```
#echo 0 > /proc/sys/net/ipv4/ip_forward
```

ou ainda:

```
#sysctl net.ipv4.ip_forward=0
```

Para desabilitar o serviço NAT do iptables basta digitar o comando:

```
#iptables -t nat -F
```

Para verificar a tabela *NAT* do *iptables* digite:

```
#iptables -t nat -L -v
```

Agora o sistema *firewall* se baseia em serviço de *proxy* para a rede interna, evitando contato direto da rede interna com a rede *extena*. Caso algum serviço interno necessite de acesso direto à rede externa (tanto à internet quanto à *rede de perímetro*) sugerimos que faça-se como a sugestão de ZWICKY (2000, p.249) de utilizar um tipo especial de host de bastião chamado por ele de *máquina vítima*, a qual é implementada para servir estes tipos de

serviços que não podem ser oferecidos de forma segura.

Sugere-se colocar este computador na *rede de perímetro*, com um backup constante. Um exemplo desta abordagem seria um servidor *web* de *portfólio* para clientes externos, por exemplo, e negar qualquer tráfego para a rede interna vindas desta máquina.

Para bloquear determinado tipo de tráfego *http* dos clientes devemos criar uma *Access Control List* ou *ACL* e aplicar uma regra à esta *ACL*. As *ACLs* são inseridas dentro do arquivo */etc/squid/squid.conf* e são lidas seqüencialmente de cima para baixo. Por exemplo para bloquearmos o tráfego de *MSN messenger* e do serviço de e-mail *hotmail* devemos inserir as seguintes linhas no arquivo *squid.conf*:

```
acl hotmail dstdomain messenger.hotmail.com
acl msn dstdomain loginnet.passport.com
http_access deny all msn hotmail
```

A seguir devemos reiniciar o processo do squid:

```
#cd /etc/rc.d
#./rc.squid restart
```

Cada nova *ACL* inserida só entrará em vigor após reiniciar o *proxy*.

APÊNDICE G – Utilização do Open SSH

Os sistemas que se utilizam do *OpenSSH* procuram por inicializar o serviço na primeira inicialização do sistema e já criam as chaves para o padrão *ssh1* utilizando o padrão criptográfico *RSA* para as chaves, *ssh2* utilizando o *RSA* para as chaves e criam chaves para o padrão *ssh2* utilizando o algoritmo criptográfico *DSA* para gerar as chaves.

As chaves situam-se no diretório */etc/ssh*. Segue uma lista dos arquivos contidos neste diretório:

- ssh_config: Este é o arquivo de configuração do cliente *ssh*;
- sshd_config: Este é o arquivo de configuração do servidor *ssh*;
- ssh_host_dsa_key: Chave privada do algoritmo criptográfico *DSA* para utilização do protocolo *ssh* versão 2;
- ssh_host_key: arquivo contendo a chave do host, utilizado pelo protocolo *ssh* versão 1;
- ssh_host_rsa_key: Chave privada do algoritmo criptográfico *RSA* para utilização do protocolo *ssh* versão 2;
- ssh_host_dsa_key.pub: Chave pública do algoritmo criptográfico *DSA* para utilização do protocolo *ssh* versão 2;
- ssh_host_key.pub: Arquivo contendo a chave pública do host, utilizado pelo protocolo *ssh* versão 1;
- ssh_host_rsa_key.pub: Chave pública do algoritmo criptográfico *RSA*

para utilização do protocolo *ssh* versão 2.

Observação: o *sshd* irá negar a utilização se os arquivos de chave estiverem configurados com acessibilidade para grupos de usuários ou acessível de forma global.

Normalmente as distribuições inicializam o serviço *ssh* por padrão, gerando as chaves *DSA* e RSA para o host utilizar o protocolo *ssh* versão 2. É sabido que o protocolo *ssh* versão 1 contém algumas vulnerabilidades de segurança então o configuraremos para aceitar apenas a versão 2 no tópico *Configurando o servidor ssh para aceitar apenas a versão 2.*

Executando comandos arbitrários com o OpenSSH

Para a utilização do *ssh* é necessário que haja um servidor na máquina a qual solicitaremos o serviço. A sintaxe do cliente *ssh* é:

ssh <usuário>@<host> <comandos>

onde:

- *ssh* – Programa cliente do serviço *ssh;*
- <usuário> - Usuário para se autenticar na máquina remota;
- <host> - Nome da máquina remota na qual desejamos nos conectar,

pode ser o endereço IP da máquina de destino;

- <comandos> - Comando a ser executado na máquina remota.

Caso as condições sejam satisfeitas no computador remoto ao qual estamos requisitando, o *ssh* solicitará a senha do usuário especificado no sistema remoto.

Faremos um exemplo da utilização do comando *ls /* no computador de endereço IP 192.168.9.12 com o usuário *root*:

#ssh root@192.168.9.12 ls /

root@192.168.9.12's password: (digite a senha do usuário remoto no sistema remoto, observe que ela não é mostrada no console)

```
bin
boot
dev
etc
home
initrd
lib
lost+found
misc
mnt
opt
proc
root
sbin
tmp
usr
var
```

O comando acima executa o comando *ls* no diretório raiz (/) na máquina de endereço IP 192.168.9.12 com o usuário *root*, mostrando o conteúdo do diretório raiz.

Utilizando o OpenSSH para acesso remoto

Existe ainda a opção de autenticação no sistema remoto com o *ssh*. Esta opção permite que se utilize um console na máquina remota a partir da máquina local, fazendo com que se tenha a ilusão de estar na própria máquina remota. Para utilizar o *ssh* para acesso remoto utilize a seguinte sintaxe:

```
ssh -l <usuário> <máquina remota>
```

Onde <usuário> é um usuário válido na máquina remota e <máquina remota> é o nome do host remoto ou endereço IP da máquina remota.

Por exemplo:

```
#ssh -l root 192.168.9.12
```

O comando acima solicitará a senha do usuário *root* do sistema que tem o endereço IP 192.168.9.12 como mostrado a seguir em um exemplo completo:

```
# ssh -l root 192.168.9.12
root@192.168.9.12's password: <digite a senha e 'enter'>
Last login: Fri Sep 30 15:39:57 2005 from 192.168.9.9
[root@jhouse05 root]#
```

Após o processo de autenticação o *shell* da máquina remota é exibido, como mostrado acima em *[root@jhouse05 root]#,* com a vantagem de utilizar

criptografia para a transferência dos dados entre os computadores.

Utilizando o OpenSSH para transferência de arquivos

O *Open Secure Shell* possibilita a transferência segura (criptografada) de arquivos através do comando *scp*.

A sintaxe do comando *scp* é:

scp (opções) <usuario>@<hostOrigem>:<caminho> <usuario>@<hostDestino>:<caminho>

Onde (opções) - significam as opções do comando *scp*. As opções mais usadas são:

- -r : Cópia recursiva de diretórios
- -1 : Força a utilização da versão 1 do protocolo SSH.
- -2 : Força a utilização da versão 2 do protocolo SSH.
- -4 : Força a utilização do IP versão 4.
- -6 : Força a utilização do IP versão 6.
- -l : Força a limitação da largura de banda em Kbits por segundo.
- -o : Opções do ssh. Pode ser usado o mesmo formato do arquivo ssh_config.
- <usuario> - Nome do usuário a acessar o sistema remoto.
- <hostOrigem> - Nome ou endereço IP do computador (host) de origem.

- <hostDestino> - Nome ou endereço IP do computador (host) de destino.

Faremos um exemplo de transferência de um arquivo da máquina local para a máquina remota de endereço IP 192.168.9.12. Lembramos que deve haver um servidor *ssh* (*sshd*) rodando na máquina na qual queremos transferir o arquivo.

Transferiremos um arquivo localizado no diretório do super-usuário (usuário *root*) chamado *fortune.txt* para a máquina remota supra-citada, colocaremos o arquivo no diretório do super-usuário da máquina remota (/root):

```
# scp /root/fortune.txt root@192.168.9.12:/root
root@192.168.9.12's password:
fortune.txt                   100%  655     0.6KB/s   00:00
```

O comando acima solicitou a senha do usuário na máquina remota e a seguir transferiu o arquivo, exibindo algumas informações relevantes à transferência. Cada arquivo transferido aparece com sua estatística de transferência. A transferência do arquivo acima foi completa, como indica a percentagem (100%), foram transferidos 655 Bytes, indicado pela segunda coluna que contém o número 655, a uma taxa de 0,6 KiloBytes por segundo (0.6KB/s) em um tempo menor do que um segundo.

Para a transferência de um diretório e seu conteúdo basta que utilizemos a opção de recursividade de diretórios, indicada pela opção *r*. Por exemplo:

```
$ scp -r /opt/fss-0.24 root@192.168.9.12:/opt
root@192.168.9.12's password:
CHANGES                   100%  1997    2.0KB/s   00:00
FAQ                       100%  1220    1.2KB/s   00:00
TODO                      100%  175     0.2KB/s   00:00
```

```
fss                    100%   11KB  10.9KB/s   00:00
fss.1                  100%  3348    3.3KB/s   00:00
fss.conf               100%  347     0.3KB/s   00:00
```

No comando acima transferimos o diretório *fss-0.24* localizado no diretório */opt* da máquina local e todo o seu conteúdo para o diretório */opt* da máquina remota de endereço IP 192.168.9.12.

Note que para logarmos no sistema remoto como super-usuário não é necessário que sejamos o super-usuário da máquina local, basta que saibamos a senha de algum usuário na máquina remota (e o nome de usuário é claro) e que tenhamos um servidor ssh rodando na máquina remota.

Configurando o servidor ssh para aceitar apenas a versão 2

Para fazer com que o servidor *ssh* aceite apenas conexões do protocolo *ssh* versão 2 teremos que editar o arquivo */etc/ssh/sshd_config*. Para isso devemos primeiro entrar no diretório */etc/ssh* e após, editar o arquivo *sshd_config*:

```
#cd /etc/ssh
#vi sshd_config
```

Ao entrar no editor *vi* tecle *i* para selecionar o modo de edição.

Procure pela linha *#Protocol 2,1* e remova o sustenido (#), a vírgula e o número 1.

A linha deve ficar assim:

```
Protocol 2
```

Tecle *esc* para sair do modo de edição e entrar no modo de comandos. Tecle *:wq* para salvar e sair do *vi*.

Agora devemos reiniciar o servidor, indo até o diretório */etc/rc.d*

e executar o script *rc.sshd* com o parâmetro *restart*:

```
# cd /etc/rc.d
#./rc.sshd restart

WARNING: killing listener process only.  To kill every sshd process, you must
use 'rc.sshd stop'.  'rc.sshd restart' kills only the parent sshd to allow an admin logged
in through sshd to use 'rc.sshd restart' without being cut off.  If sshd has been
upgraded, new connections will now use the new version, which should be a safe
enough approach.
```

A mensagem acima informa que apenas o processo que está aguardando conexões será interrompido e que as conexões utilizando as velhas configurações estarão ainda ativas. Para reiniciar todos os processos do servidor *ssh* deve-se usar o parâmetro *stop* para terminar o serviço e a seguir reiniciar o serviço utilizando o parâmetro *start*. Se houver conexões com o seu servidor *ssh* e quiser reiniciá-las façamos como o sugerido pela mensagem de aviso:

```
#cd /etc/rc.d
#./sshd stop
#./sshd start
```

O serviço *ssh* está agora funcionando apenas com a versão 2 do

protocolo *ssh*, com as funcionalidades de transferência de arquivos e acesso remoto.

O software *ssh* possibilita também a criação de túneis criptográficos para as conexões de rede, mas ficando aquém do nosso escopo. Cremos que as necessidades básicas já foram atingidas com a documentação acima e não tendo a necessidade de criação de túneis via *ssh*.

APÊNDICE H – Instalação do antivírus *clamAV*

O *clamav* é um software antivírus disponibilizado para os usuários de software livre, podendo ser utilizado para verificar arquivos à procura de vírus. O software é capaz de verificar por vírus em arquivos comprimidos e pode ter sua base de dados de vírus atualizada através da internet. O sotware pode ser obtido em www.clamav.com na seção *downloads*.

Para intalar o *clamAV* é necessário a criação de um grupo *clamav* contendo um usuário *clamav*, para isto utilize os comandos:

```
# groupadd clamav
## adduser clamav

Login name for new user: clamav

User ID ('UID') [ defaults to next available ]:

Initial group [ users ]: clamav

Additional groups (comma separated) []:

Home directory [ /home/clamav ]

Shell [ /bin/bash ] /bin/sh
- Warning: /bin/sh is not in /etc/shells (potential problem using FTP)
  Do you wish to change the shell ? (Y/n)  y
Shell [ /bin/bash ]

Expiry date (YYYY-MM-DD) []:

New account will be created as follows:

---------------------------------------
Login name.......:  clamav
UID..............:  [ Next available ]
Initial group....:  clamav
```

```
Additional groups:  [ None ]
Home directory...:  /home/clamav
Shell.....:  /bin/bash
Expiry date.....:  [ Never ]

This is it... if you want to bail out, hit Control-C.  Otherwise, press
ENTER to go ahead and make the account.

Creating new account...

Changing the user information for clamav
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Changing password for clamav
Enter the new password (minimum of 5, maximum of 127 characters)
Please use a combination of upper and lower case letters and numbers.
New password:
Re-enter new password:
Password changed.

Account setup complete.
```

Agora basta utilizar o usuário para instalar o clamAV. Para isto utilize os comandos, supondo que baixamos no diretório do usuário root o arquivo de instalação:

```
#cd /home/clamav
#tar xvfz /root/clamav-0.87.1.tar.gz
#chown -R clamav:clamav clamav-0.87.1/
#cd clamav-0.87.1/
#su clamav
$./configure
$make; make check
$exit
#make install
```

ANEXOS

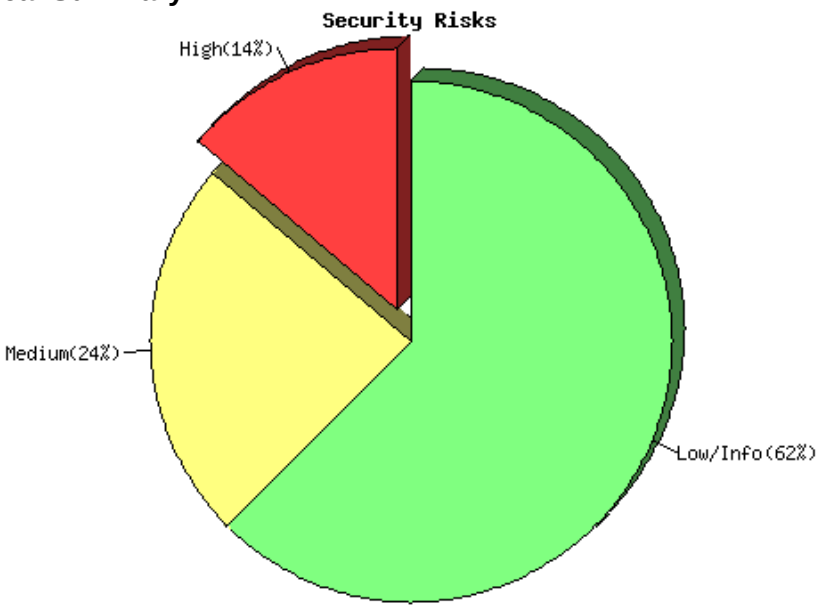
ANEXO A – Relatório geral do *nessus* no dia 15 de novembro de 2005.

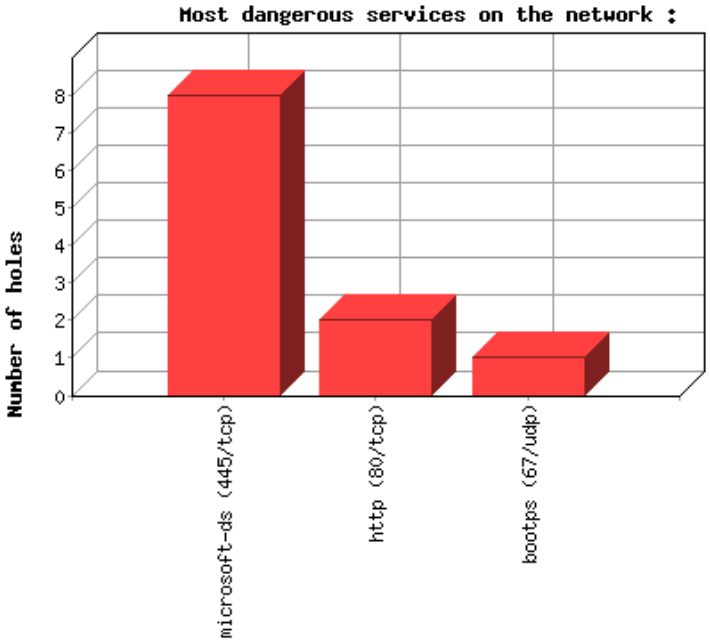
Nessus Report

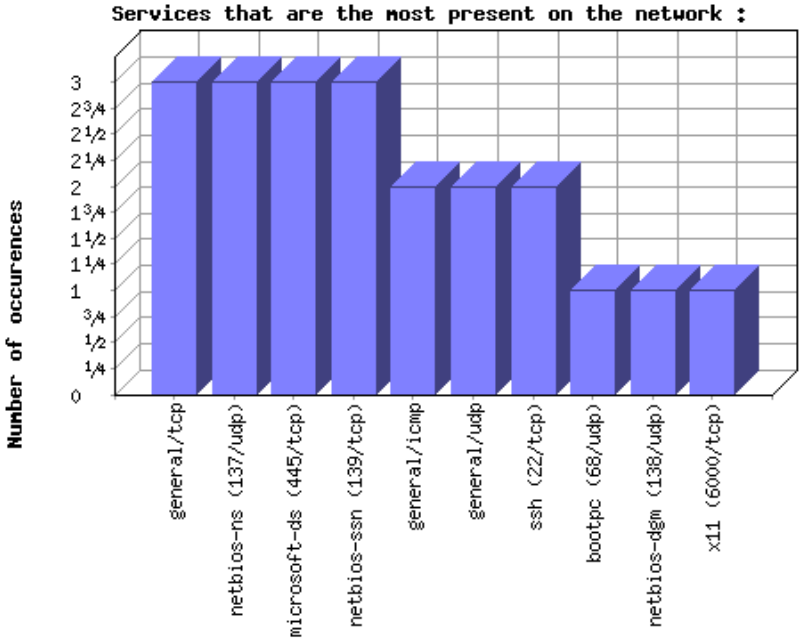
The Nessus Security Scanner was used to assess the security of 3 hosts

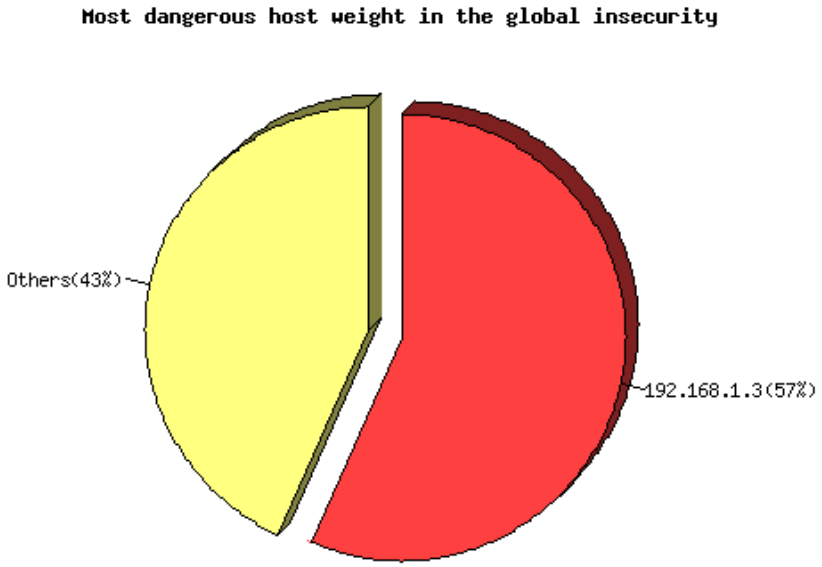
- **11 security holes have been found**
- **19 security warnings have been found**
- **50 security notes have been found**

Part I : Graphical Summary :




Most dangerous services on the network :
Number of holes
0
1
2
3
4
5
6
7
8
microsoft-ds (445/tcp)
http (80/tcp)
bootps (67/udp)


Services that are the most present on the network :
Number of occurences
0
1/4
1/2
3/4
1
1 1/4
1 1/2
1 3/4
2
2 1/4
2 1/2
2 3/4
3
general/tcp
netbios-ns (137/udp)
microsoft-ds (445/tcp)
netbios-ssn (139/tcp)
general/icmp
general/udp
ssh (22/tcp)
bootpc (68/udp)
netbios-dgm (138/udp)
x11 (6000/tcp)



Part II. Results, by host :

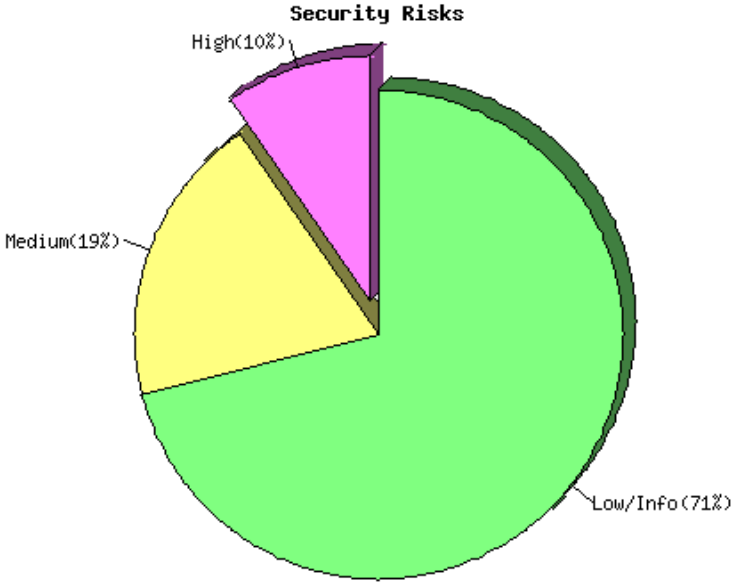
.1	192.168.1	(**found 3 security holes**)
.3	192.168.1	(**found 8 security holes**)
.4	192.168.1	(**found 4 security warnings**)

This file was generated by Nessus, the open-sourced security scanner.

ANEXO B – Relatório do nessus para o host de bastião no endereço IP 192.168.1.1 no dia 15 de novembro de 2005.

192.168.1.1

Repartition of the level of the security problems :



[Next host : 192.168.1.3]

List of open ports :

- *ssh (22/tcp)* *(Security warnings found)*
- *domain (53/tcp)* *(Security notes found)*
- *http (80/tcp)* *(Security hole found)*
- *netbios-ssn (139/tcp)* *(Security notes found)*
- *microsoft-ds (445/tcp)* *(Security warnings found)*
- *mysql (3306/tcp)* *(Security notes found)*
- *general/udp* *(Security notes found)*
- *netbios-ns (137/udp)* *(Security warnings found)*
- *general/tcp* *(Security warnings found)*
- *bootps (67/udp)* *(Security hole found)*
- *domain (53/udp)* *(Security notes found)*
- *general/icmp* *(Security warnings found)*

[back to the list of ports]

Warning found on port ssh (22/tcp)

The remote SSH daemon supports connections made using the version 1.33 and/or 1.5 of the SSH protocol. These protocols are not completely cryptographically safe so they should not be used. Solution : If you use OpenSSH, set the option 'Protocol' to '2' If you use SSH.com's set the option 'Ssh1Compatibility' to 'no' Risk factor : Low Nessus ID : 10882

[back to the list of ports]

Information found on port ssh (22/tcp)

An ssh server is running on this port Nessus ID : 10330

[back to the list of ports]

Information found on port ssh (22/tcp)

Remote SSH version : SSH-1.99-OpenSSH_4.2 Remote SSH supported authentication : publickey,password,keyboard-interactive Nessus ID : 10267

[back to the list of ports]

Information found on port ssh (22/tcp)

The remote SSH daemon supports the following versions of the SSH protocol : . 1.33 . 1.5 . 1.99 . 2.0 SSHv1 host key fingerprint : b9:4e:09:32:f9:f5:2c:c4:80:68:b2:3d:91:36:59:68 SSHv2 host key fingerprint : 9e:e6:3f:e8:47:1b:59:48:97:2b:8e:15:ca:1f:29:a9 Nessus ID : 10881

[back to the list of ports]

Information found on port domain (53/tcp)

BIND 'NAMED' is an open-source DNS server from ISC.org. Many proprietary DNS servers are based on BIND source code. The BIND based NAMED servers (or DNS servers) allow remote users to query for version and type information. The query of the CHAOS TXT record 'version.bind', will typically prompt the server to send the information back to the querying source. The remote bind version is : 9.3.1 Solution : Using the 'version' directive in the 'options' section will block the 'version.bind' query, but it will not log such attempts. Nessus ID : 10028

[back to the list of ports]

Information found on port domain (53/tcp)

A DNS server is running on this port. If you do not use it, disable it. Risk factor : Low Nessus ID : 11002

[back to the list of ports]

Vulnerability found on port http (80/tcp)

The following URLs seem to be vulnerable to various SQL injection techniques : /acid_stat_class.php?num_result_rows='UNION'
/acid_stat_class.php?num_result_rows='
/acid_stat_class.php?num_result_rows='%22
/acid_stat_class.php?num_result_rows=9%2c+9%2c+9
/acid_stat_class.php?num_result_rows='bad_bad_value
/acid_stat_class.php?num_result_rows=bad_bad_value'
/acid_stat_class.php?num_result_rows='+OR+'
/acid_stat_class.php?num_result_rows='WHERE
/acid_stat_class.php?num_result_rows=%3B
/acid_stat_class.php?num_result_rows='OR An attacker may exploit this flaws to bypass authentication or to take the control of the remote database. Solution : Modify the relevant CGIs so that they properly escape arguments Risk factor : High See also : http://www.securiteam.com/securityreviews/5DP0N1P76E.html Nessus ID : 11139

[back to the list of ports]

Vulnerability found on port http (80/tcp)

The following URLs seem to be vulnerable to various SQL injection techniques : /acid_stat_sensor.php?num_result_rows='UNION' /acid_stat_sensor.php?num_result_rows=' /acid_stat_sensor.php?num_result_rows='%22 /acid_stat_sensor.php?num_result_rows=9%2c+9%2c+9 /acid_stat_sensor.php?num_result_rows='bad_bad_value /acid_stat_sensor.php?num_result_rows=bad_bad_value' /acid_stat_sensor.php?num_result_rows='+OR+' /acid_stat_sensor.php?num_result_rows='WHERE /acid_stat_sensor.php?num_result_rows=%3B /acid_stat_sensor.php?num_result_rows='OR An attacker may exploit this flaws to bypass authentication or to take the control of the remote database. Solution : Modify the relevant CGIs so that they properly escape arguments Risk factor : High See also : http://www.securiteam.com/securityreviews/5DP0N1P76E.html Nessus ID : 11139

[back to the list of ports]

Warning found on port http (80/tcp)

Your webserver supports the TRACE and/or TRACK methods. TRACE and TRACK are HTTP methods which are used to debug web server connections. It has been shown that servers supporting this method are subject to cross-site-scripting attacks, dubbed XST for "Cross-Site-Tracing", when used in conjunction with various weaknesses in browsers. An attacker may use this flaw to trick your legitimate web users to give him their credentials. Solution: Disable these methods. If you are using Apache, add the following lines for each virtual host in your configuration file : RewriteEngine on RewriteCond %{REQUEST_METHOD} ^(TRACE|TRACK) RewriteRule .* - [F] If you are using Microsoft IIS, use the URLScan tool to deny HTTP TRACE requests or to permit only the methods needed to meet site requirements and policy. If you are using Sun ONE Web Server releases 6.0 SP2 and later, add the following to the default object section in obj.conf: <Client method="TRACE"> AuthTrans fn="set-variable" remove-headers="transfer-encoding" set-headers="content-length: -1" error="501" </Client> If you are using Sun ONE Web Server releases 6.0 SP2 or below, compile the NSAPI plugin located at: http://sunsolve.sun.com/pub-cgi/retrieve.pl?doc=fsalert%2F50603 See http://www.whitehatsec.com/press_releases/WH-PR-20030120.pdf http://archives.neohapsis.com/archives/vulnwatch/2003-q1/0035.html http://sunsolve.sun.com/pub-cgi/retrieve.pl?doc=fsalert%2F50603 http://www.kb.cert.org/vuls/id/867593 Risk factor : Medium BID : 9506, 9561, 11604 Nessus ID : 11213

[back to the list of ports]

Information found on port http (80/tcp)

A web server is running on this port Nessus ID : 10330

[back to the list of ports]

Information found on port http (80/tcp)

The following directories were discovered: /cgi-bin, /docs, /icons, /session, /tests While this is not, in and of itself, a bug, you should manually inspect these directories to ensure that they are in compliance with company security standards Nessus ID : 11032

[back to the list of ports]

Information found on port http (80/tcp)

The following CGI have been discovered : Syntax : cginame (arguments [default value]) /acid_stat_ports.php (port_type [1] num_result_rows [0] sort_order [last_d] caller [last_ports] current_view [0] proto [-1]) /acid_stat_class.php (num_result_rows [0] submit [ALL on Screen] action [] sort_order [class_a] caller [] current_view [0] action_arg []) /acid_stat_alerts.php (time%5B0%5D%5B7%5D [] num_result_rows [0] submit [ALL on Screen] action [] sort_order [last_d] time%5B0%5D%5B8%5D [+] caller [last_alerts] time%5B0%5D%5B0%5D [+] time%5B0%5D%5B9%5D [+] time_cnt [1] time%5B0%5D%5B1%5D [%3D] current_view [0] time%5B0%5D%5B2%5D [11] time%5B0%5D%5B3%5D [15] action_arg [] time%5B0%5D%5B4%5D [2005] time%5B0%5D%5B5%5D [] time%5B0%5D%5B6%5D []) /acid_stat_uaddr.php (time%5B0%5D%5B7%5D [] addr_type [1] num_result_rows [0] time%5B0%5D%5B8%5D [+] sort_order [occur_d] caller [] time%5B0%5D%5B0%5D [+] time%5B0%5D%5B9%5D [+] time_cnt [1] time%5B0%5D%5B1%5D [%3D] current_view [0] time%5B0%5D%5B2%5D [11] time%5B0%5D%5B3%5D [15] time%5B0%5D%5B4%5D [2005] time%5B0%5D%5B5%5D [] time%5B0%5D%5B6%5D []) /acid_ag_main.php (num_result_rows [-1] ag_action [list] sort_order [] caller [] current_view [-1] action_arg []) /acid_qry_main.php (sig_priority[0] [] data_encode[0] [] time%5B0%5D%5B7%5D [] num_result_rows [-1] submit [Query+DB] time%5B0%5D%5B8%5D [+] sort_order [time_d] time%5B0%5D%5B0%5D [+] caller [last_any] time_cnt [1] time%5B0%5D%5B9%5D [+] time%5B0%5D%5B1%5D [%3D] current_view [-1] sig[2] [] time%5B0%5D%5B2%5D [11] sig[0] [] sig_priority[1] [] data_encode[1] [] time%5B0%5D%5B3%5D [15] new [1] layer4 [TCP] time%5B0%5D%5B4%5D [2005] sig%5B0%5D [LIKE] ag [] sensor [] time%5B0%5D%5B5%5D [] sig%5B1%5D [spp_portscan] 0 [] sig_class [] time%5B0%5D%5B6%5D [] sig_type [0] sig[1] []) /acid_stat_iplink.php (num_result_rows [0] submit [ALL on Screen] action [] sort_order [sip_a] caller [] current_view [0] action_arg []) /acid_stat_sensor.php (num_result_rows [0] submit [ALL on Screen] action [] sort_order [sid_a] caller [] current_view [0] x=x [] action_arg []) Nessus ID : 10662

[back to the list of ports]

Information found on port http (80/tcp)

The remote web server type is : Apache/1.3.33 (Unix) mod_ssl/2.8.24 OpenSSL/0.9.7g PHP/4.4.0 Solution : You can set the directive 'ServerTokens Prod' to limit the information emanating from the server in its response headers. Nessus ID : 10107

[back to the list of ports]

Information found on port http (80/tcp)

An information leak occurs on Apache based web servers whenever the UserDir module is enabled. The vulnerability allows an external attacker to enumerate existing accounts by requesting access to their home directory and monitoring the response. Solution: 1) Disable this feature by changing 'UserDir public_html' (or whatever) to 'UserDir disabled'. Or 2) Use a RedirectMatch rewrite rule under Apache -- this works even if there is no such entry in the password file, e.g.: RedirectMatch ^/~(.*)$ http://my-target-webserver.somewhere.org/$1 Or 3) Add into httpd.conf: ErrorDocument 404 http://localhost/sample.html ErrorDocument 403 http://localhost/sample.html

(NOTE: You need to use a FQDN inside the URL for it to work properly). Additional Information: http://www.securiteam.com/unixfocus/5WP0C1F5FI.html Risk factor : Low CVE : CAN-2001-1013 BID : 3335 Nessus ID : 10766

[back to the list of ports]

Information found on port netbios-ssn (139/tcp)

An SMB server is running on this port Nessus ID : 11011

[back to the list of ports]

Warning found on port microsoft-ds (445/tcp)

Here is the browse list of the remote host : SERVER (os: 0.0) This is potentially dangerous as this may help the attack of a potential hacker by giving him extra targets to check for Solution : filter incoming traffic to this port Risk factor : Low Nessus ID : 10397

[back to the list of ports]

Information found on port microsoft-ds (445/tcp)

A CIFS server is running on this port Nessus ID : 11011

[back to the list of ports]

Information found on port microsoft-ds (445/tcp)

- NULL sessions are enabled on the remote host CVE : CAN-1999-0504, CAN-1999-0506, CVE-2000-0222, CAN-1999-0505, CAN-2002-1117 BID : 494, 990, 11199 Nessus ID : 10394

[back to the list of ports]

Information found on port microsoft-ds (445/tcp)

The remote native lan manager is : Samba 3.0.20 The remote Operating System is : Unix The remote SMB Domain Name is : TABAJARA Nessus ID : 10785

[back to the list of ports]

Information found on port mysql (3306/tcp)

An unknown service is running on this port. It is usually reserved for MySQL Nessus ID : 10330

[back to the list of ports]

Information found on port general/udp

For your information, here is the traceroute to 192.168.1.1 : 192.168.1.4 192.168.1.1 Nessus ID : 10287

[back to the list of ports]

Warning found on port netbios-ns (137/udp)

The following 7 NetBIOS names have been gathered : SERVER = This is the computer name registered for workstation services by a WINS client. SERVER = This is the current logged in user registered for this workstation. SERVER = Computer name __MSBROWSE__ TABAJARA = Workgroup / Domain name TABAJARA TABAJARA = Workgroup / Domain name (part of the Browser elections) . This SMB server seems to be a SAMBA server (this is not a security risk, this is for your information). This can be told because this server

claims to have a null MAC address If you do not want to allow everyone to find the NetBios name of your computer, you should filter incoming traffic to this port. Risk factor : Medium CVE : CAN-1999-0621 Nessus ID : 10150

[back to the list of ports]

Warning found on port general/tcp

The remote host does not discard TCP SYN packets which have the FIN flag set. Depending on the kind of firewall you are using, an attacker may use this flaw to bypass its rules. See also : http://archives.neohapsis.com/archives/bugtraq/2002-10/0266.html http://www.kb.cert.org/vuls/id/464113 Solution : Contact your vendor for a patch Risk factor : Medium BID : 7487 Nessus ID : 11618

[back to the list of ports]

Information found on port general/tcp

The remote host is running Clark Connect Firewall Nessus ID : 11936

[back to the list of ports]

Information found on port general/tcp

192.168.1.1 resolves as localhost. Nessus ID : 12053

[back to the list of ports]

Vulnerability found on port bootps (67/udp)

The remote host is running a DHCP server. If the remote server is ISC-DHCPd, make sure you are running the latest version, as several flaws affect older versions and may allow an attacker to gain root on this host *** Note that Nessus did not check for the presence of the *** flaws, so this might be a false positive See also : http://www.cert.org/advisories/CA-2003-01.html http://www.cert.org/advisories/CA-2002-12.html Risk factor : High CVE : CAN-2003-0026, CAN-2002-0702, CAN-2003-0039 BID : 4701, 6627, 6628, 11591 Other references : RHSA:RHSA-2003:034-01 Nessus ID : 11312

[back to the list of ports]

Information found on port bootps (67/udp)

Here is the information we could gather from the remote DHCP server. This allows an attacker on your local network to gain information about it easily : Master DHCP server of this network : 0.0.0.0 IP address the DHCP server would attribute us : 192.168.1.6 DHCP server(s) identifier = 192.168.1.1 netmask = 255.255.255.0 router = 192.168.1.1 domain name server(s) = 192.168.1.1 domain name = juarezdc.org broadcast address = 192.168.1.255 Solution : remove the options that are not in use in your DHCP server Risk factor : Low Nessus ID : 10663

[back to the list of ports]

Information found on port domain (53/udp)

A DNS server is running on this port. If you do not use it, disable it. Risk factor : Low Nessus ID : 11002

[back to the list of ports]

Information found on port domain (53/udp)

The remote name server could be fingerprinted as being : ISC BIND 9.2.3

Nessus ID : 11951

[back to the list of ports]

Information found on port domain (53/udp)

The remote DNS server answers to queries for third party domains which do not have the recursion bit set. This may allow a remote attacker to determine which domains have recently been resolved via this name server, and therefore which hosts have been recently visited. For instance, if an attacker was interested in whether your company utilizes the online services of a particular financial institution, they would be able to use this attack to build a statistical model regarding company usage of aforementioned financial institution. Of course, the attack can also be used to find B2B partners, web-surfing patterns, external mail servers, and more... For a much more detailed discussion of the potential risks of allowing DNS cache information to be queried anonymously, please see: http://community.sidestep.pt/~luis/DNS-Cache-Snooping/DNS_Cache_Snooping_1.1.pdf Risk factor : Low Nessus ID : 12217

[back to the list of ports]

Warning found on port general/icmp

The remote host answers to an ICMP timestamp request. This allows an attacker to know the date which is set on your machine. This may help him to defeat all your time based authentication protocols. Solution : filter out the ICMP timestamp requests (13), and the outgoing ICMP timestamp replies (14). Risk factor : Low CVE : CAN-1999-0524 Nessus ID : 10114

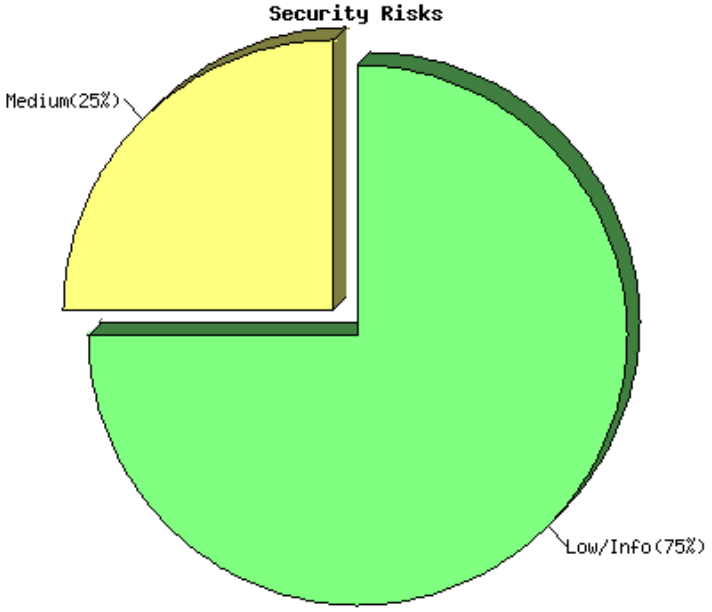
[Next host : 192.168.1.3]

This file was generated by Nessus, the open-sourced security scanner.

ANEXO C – Relatório do nessus para o servidor no endereço IP 192.168.1.4 no dia 15 de novembro de 2005.

192.168.1.4

Repartition of the level of the security problems :



[Back to the index]

List of open ports :

- *ssh (22/tcp) (Security warnings found)*
- *netbios-ssn (139/tcp) (Security notes found)*
- *microsoft-ds (445/tcp) (Security warnings found)*
- *nessus (1241/tcp) (Security warnings found)*
- *x11 (6000/tcp) (Security notes found)*
- *netbios-ns (137/udp) (Security warnings found)*
- *netbios-dgm (138/udp)*
- *bootpc (68/udp)*
- *general/tcp (Security notes found)*

[back to the list of ports]

Warning found on port ssh (22/tcp)

The remote SSH daemon supports connections made using the version 1.33 and/or 1.5 of the SSH protocol. These protocols are not completely cryptographically safe so they should not be used. Solution : If you use OpenSSH, set the option 'Protocol' to '2' If you use SSH.com's set the option 'Ssh1Compatibility' to 'no' Risk factor : Low Nessus ID : 10882

[back to the list of ports]

Information found on port ssh (22/tcp)

An ssh server is running on this port Nessus ID : 10330

[back to the list of ports]

Information found on port ssh (22/tcp)

Remote SSH version : SSH-1.99-OpenSSH_4.2 Remote SSH supported authentication : publickey,password,keyboard-interactive Nessus ID : 10267

[back to the list of ports]

Information found on port ssh (22/tcp)

The remote SSH daemon supports the following versions of the SSH protocol : . 1.33 . 1.5 . 1.99 . 2.0 SSHv1 host key fingerprint : 69:5c:cb:44:bc:3e:0b:3a:9b:d3:5a:05:f0:b6:fb:74 SSHv2 host key fingerprint : 34:bd:64:6d:fe:a4:50:4b:1e:03:90:76:a7:07:51:a7 Nessus ID : 10881

[back to the list of ports]

Information found on port netbios-ssn (139/tcp)

An SMB server is running on this port Nessus ID : 11011

[back to the list of ports]

Warning found on port microsoft-ds (445/tcp)

Here is the browse list of the remote host : CAVERINHA (os: 0.0) This is potentially dangerous as this may help the attack of a potential hacker by giving him extra targets to check for Solution : filter incoming traffic to this port Risk factor : Low Nessus ID : 10397

[back to the list of ports]

Information found on port microsoft-ds (445/tcp)

A CIFS server is running on this port Nessus ID : 11011

[back to the list of ports]

Information found on port microsoft-ds (445/tcp)

- NULL sessions are enabled on the remote host CVE : CAN-1999-0504, CAN-1999-0506, CVE-2000-0222, CAN-1999-0505, CAN-2002-1117 BID : 494, 990, 11199 Nessus ID : 10394

[back to the list of ports]

Information found on port microsoft-ds (445/tcp)

The remote native lan manager is : Samba 3.0.20 The remote Operating System is : Unix The remote SMB Domain Name is : MYGROUP Nessus ID : 10785

[back to the list of ports]

Warning found on port nessus (1241/tcp)

A Nessus Daemon is listening on this port. Nessus ID : 10147

[back to the list of ports]

Information found on port nessus (1241/tcp)

A TLSv1 server answered on this port Nessus ID : 10330

[back to the list of ports]

Information found on port nessus (1241/tcp)

Here is the TLSv1 server certificate: Certificate: Data: Version: 3 (0x2) Serial Number: 1 (0x1) Signature Algorithm: md5WithRSAEncryption Issuer: C=BR, ST=RS, L=POA, O=Nessus Users United, OU=Certification Authority for caverinha, CN=caverinha/emailAddress=ca@caverinha Validity Not Before: Nov 15 19:32:01 2005 GMT Not After : Nov 15 19:32:01 2006 GMT Subject: C=BR, ST=RS, L=POA, O=Nessus Users United, OU=Server certificate for caverinha, CN=caverinha/emailAddress=nessusd@caverinha Subject Public Key Info: Public Key Algorithm: rsaEncryption RSA Public Key: (1024 bit) Modulus (1024 bit): 00:d8:53:35:8d:df:20:71:6a:8c:a1:46:93:0c:7b: 27:37:7c:1c:d0:4b:37:f0:a6:b6:d0:97:4a:d5:9e: ac:01:9e:6d:55:1d:06:c7:28:97:99:1c:69:e1:d9: 42:65:1e:17:63:8f:11:99:2a:29:59:e1:88:94:d3: b9:c3:17:80:69:97:34:5c:8e:57:df:95:dd:24:5c: 13:ed:f9:93:a1:f6:a2:be:23:28:64:b3:40:75:43: 4a:19:df:40:ed:a1:76:09:49:7b:fc:0b:7c:ed:77: b9:9d:05:37:19:50:fe:e8:d8:08:7c:8a:70:ad:71: e0:95:81:5f:95:22:60:5d:6f Exponent: 65537 (0x10001) X509v3 extensions: Netscape Cert Type: SSL Server X509v3 Key Usage: Digital Signature, Non Repudiation, Key Encipherment Netscape Comment: OpenSSL Generated Certificate X509v3 Subject Key Identifier: 86:4D:47:83:E2:8F:D0:76:5A:66:79:82:55:3C:5C:4D:13:35:46:80 X509v3 Authority Key Identifier: keyid:91:A0:70:28:EE:1A:A1:1A:9A:03:57:F1:71:3C:D5:7E:46:86:45:CD DirName:/C=BR/ST=RS/L=POA/O=Nessus Users United/OU=Certification Authority for caverinha/CN=caverinha/emailAddress=ca@caverinha serial:A7:E8:5F:D7:90:E6:E2:7C X509v3 Subject Alternative Name: email:nessusd@caverinha X509v3 Issuer Alternative Name: <EMPTY> Signature Algorithm: md5WithRSAEncryption 42:59:a5:4f:fa:20:68:21:16:10:30:94:87:7d:19:4d:5a:6b: 80:35:9f:ca:1a:92:b8:f4:ed:c2:ee:33:35:98:09:ec:88:4b: 92:59:3c:28:83:d6:9a:9f:25:50:b3:3b:86:27:68:29:43:42: 24:82:3e:0a:05:12:93:e1:90:44:c9:6f:a4:8a:ef:7d:a4:57: 29:ee:f4:96:b7:1d:23:95:00:c0:49:e5:a5:7e:9f:9b:0e:65: 2c:60:cd:29:8f:98:d6:a0:5f:6b:ae:a7:c0:3d:e4:e9:6e:2c: 6e:ba:6a:be:77:d4:39:d4:3a:7e:61:ad:22:75:8d:f7:17:b6: 33:47 This TLSv1 server does not accept SSLv2 connections. This TLSv1 server does not accept SSLv3 connections. Nessus ID : 10863

[back to the list of ports]

Information found on port x11 (6000/tcp)

This X server does *not* allow any client to connect to it however it is recommended that you filter incoming connections to this port as attacker may send garbage data and slow down your X session or even kill the server. Here is the server version : 11.0 Here is the message we received : No protocol specified Solution : filter incoming connections to ports 6000-6009 Risk factor : Low Nessus ID : 10407

[back to the list of ports]

Warning found on port netbios-ns (137/udp)

The following 7 NetBIOS names have been gathered : CAVERINHA =

This is the computer name registered for workstation services by a WINS client. CAVERINHA = This is the current logged in user registered for this workstation. CAVERINHA = Computer name __MSBROWSE__ MYGROUP = Workgroup / Domain name MYGROUP MYGROUP = Workgroup / Domain name (part of the Browser elections) . This SMB server seems to be a SAMBA server (this is not a security risk, this is for your information). This can be told because this server claims to have a null MAC address If you do not want to allow everyone to find the NetBios name of your computer, you should filter incoming traffic to this port. Risk factor : Medium CVE : CAN-1999-0621 Nessus ID : 10150

[back to the list of ports]

Information found on port general/tcp

192.168.1.4 resolves as juarezdc.juarezdc.org. Nessus ID : 12053

[back to the list of ports]

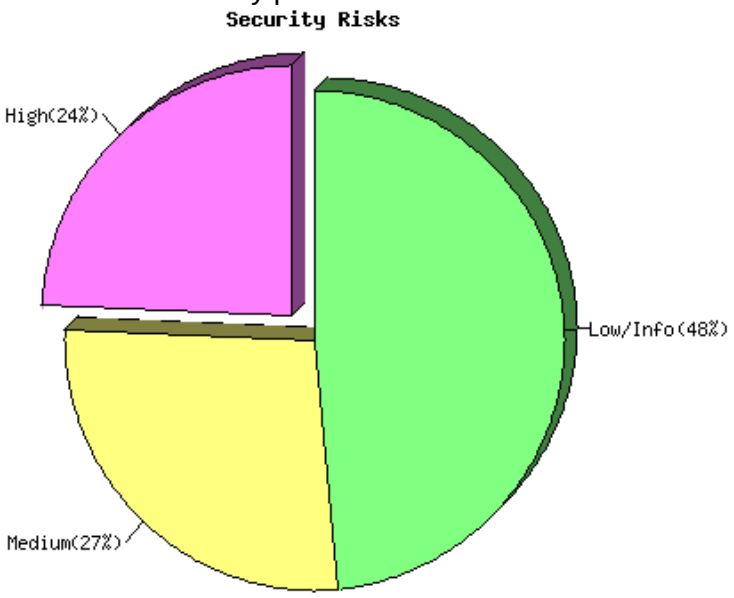
Information found on port general/tcp

The output of "uname -a" is : Linux caverinha 2.6.13.4 #5 SMP Wed Oct 26 22:22:36 BRST 2005 i686 unknown unknown GNU/Linux The remote Linux distribution is not supported, therefore local security checks have not been enabled Nessus ID : 12634

ANEXO D – Relatório do *nessus* para o cliente no endereço IP 192.168.1.3 no dia 15 de novembro de 2005.

192.168.1.3

Repartition of the level of the security problems :



[Next host : 192.168.1.4]

List of open ports :

- *loc-srv (135/tcp)* *(Security warnings found)*
- *netbios-ssn (139/tcp)* *(Security notes found)*
- *microsoft-ds (445/tcp)* *(Security hole found)*
- *blackjack (1025/tcp)* *(Security notes found)*
- *general/udp* *(Security notes found)*
- *general/tcp* *(Security warnings found)*
- *netbios-ns (137/udp)* *(Security warnings found)*
- *general/icmp* *(Security warnings found)*

[back to the list of ports]

Warning found on port loc-srv (135/tcp)

Distributed Computing Environment (DCE) services running on the remote host can be enumerated by connecting on port 135 and doing the appropriate queries. An attacker may use this fact to gain more knowledge about the remote host. Solution : filter incoming traffic to this port. Risk factor : Low Nessus ID : 10736

[back to the list of ports]

Information found on port netbios-ssn (139/tcp)

An SMB server is running on this port Nessus ID : 11011

[back to the list of ports]

Vulnerability found on port microsoft-ds (445/tcp)

The remote host is running either Windows Media Player 9 or MSN Messenger. There is a vulnerability in the remote version of this sofware which may allow an attacker to execute arbitrary code on the remote host. To exploit this flaw, one attacker would need to set up a rogue PNG image and send it to a victim on the remote host. Solution : http://www.microsoft.com/technet/security/bulletin/ms05-009.mspx Risk factor : High CVE : CAN-2004-1244, CAN-2004-0597 BID : 12485, 12506 Nessus ID : 16328

[back to the list of ports]

Vulnerability found on port microsoft-ds (445/tcp)

The remote host has a version of Outlook express which has a bug in its MHTML URL processor, which may allow an attacker to execute arbitrary code on this host. To exploit this flaw, an attacker would need to send a malformed email to a user of this host using Outlook, or would need to lure him into visiting a rogue website. Solution : http://www.microsoft.com/technet/security/bulletin/ms04-013.mspx Risk factor : High CVE : CAN-2004-0380 BID : 9105, 9107, 9658 Other references : IAVA:2004-A-0009 Nessus ID : 12208

[back to the list of ports]

Vulnerability found on port microsoft-ds (445/tcp)

The remote host is running a version of Internet Explorer 6 SP1 which is vulnerable to a vulnerability which may allow an attacker to execute arbitrary code on the remote host. To exploit this flaw, an attacker would need to lure a victim on the remote system into visiting a rogue website. See http://www.microsoft.com/technet/security/bulletin/ms04-040.mspx Risk factor : High CVE : CAN-2004-1050 Other references : IAVA:2004-A-0020 Nessus ID : 15894

[back to the list of ports]

Vulnerability found on port microsoft-ds (445/tcp)

We were able to determine that you are running IE Version 6.0000 CVE : CAN-2004-0842, CAN-2004-0727, CAN-2004-0216, CAN-2004-0839, CAN-2004-0844, CAN-2004-0843, CAN-2004-0841, CAN-2004-0845, CAN-2003-0814, CAN-2003-0815, CAN-2003-0816, CAN-2003-0817, CAN-2003-0823, CAN-2004-0549, CAN-2004-0566, CAN-2003-1048, CAN-2001-1325, CAN-2001-0149, CAN-2001-0727, CAN-2001-0875, CVE-2001-1325, CVE-2001-0149, CVE-2001-0727, CVE-2001-0875, CVE-2001-0339, CVE-2001-0002, CAN-2002-0190, CVE-2002-0026, CAN-2003-1326, CVE-2002-0027, CVE-2002-0022, CAN-2003-1328, CAN-2002-1262, CAN-2002-0193, CAN-1999-1016, CVE-2003-0344, CAN-2003-0233, CAN-2003-0309, CAN-2003-0113, CAN-2003-0114, CAN-2003-0115, CAN-2003-0116, CAN-2003-0531, CAN-2003-0809, CAN-2003-0530, CAN-2003-1025, CAN-2003-1026, CAN-2003-1027 BID : 11388, 11385, 11383, 11381, 11377, 11367, 11366, 10473, 8565, 9009, 9012, 9013, 9014, 9015, 9182, 9663, 9798, 12477, 12475, 12473, 12530 Other references : IAVA:2003-A-0014, IAVA:2004-A-0016 Nessus ID : 10861

[back to the list of ports]

Vulnerability found on port microsoft-ds (445/tcp)

The remote host contains a version of the Windows kernel which is vulnerable to a security flaw in the way that cursors and icons are handleld. An attacker may be able to execute arbitrary code on the remote host by constructing a malicious web page and entice a victim to visit this web page. An attacker may send a malicious email to the victim to exploit this flaw too. Solution : http://www.microsoft.com/technet/security/bulletin/ms05-002.mspx Risk factor : High CVE : CAN-2004-1305, CAN-2004-1049 BID : 12233 Nessus ID : 16124

[back to the list of ports]

Vulnerability found on port microsoft-ds (445/tcp)

The remote host is subject to two vulnerabilities in the HTML Help and showHelp modules, which could allow an attacker to execute arbitrary code on the remote host. To exploit this flaw, an attacker would need to set up a rogue website containing a malicious showHelp URL, and would need to lure a user on the remote host to visit it. Once the user visits the web site, a buffer overflow would allow the attacker to execute arbitrary commands with the privileges of the victim user. Solution : http://www.microsoft.com/technet/security/bulletin/ms04-023.mspx Risk factor : High CVE : CAN-2004-0201, CAN-2003-1041 BID : 10705, 9320 Other references : IAVA:2004-A-0012 Nessus ID : 13641

[back to the list of ports]

Vulnerability found on port microsoft-ds (445/tcp)

The remote version of Windows contains a flaw in the Windows Shell which may allow an attacker to elevate his privileges and/or execute arbitrary code on the remote host. To exploit this flaw, an attacker would need to lure a victim into visiting a malicious website or into opening a malicious file attachment. Solution : http://www.microsoft.com/technet/security/bulletin/MS05-008.mspx Risk factor : High CVE : CAN-2005-0053 Nessus ID : 16324

[back to the list of ports]

Vulnerability found on port microsoft-ds (445/tcp)

The remote host contains a version of the HTML Help ActiveX control which is vulnerable to a security flaw which may allow an attacker to execute arbitrary code on the remote host by constructing a malicious web page and entice a victim to visit this web page. Solution : http://www.microsoft.com/technet/security/bulletin/ms05-001.mspx Risk factor : High CVE : CAN-2004-1043 Nessus ID : 16123

[back to the list of ports]

Warning found on port microsoft-ds (445/tcp)

Here is the browse list of the remote host : CAVERA (os: 5.0) This is potentially dangerous as this may help the attack of a potential hacker by giving him extra targets to check for Solution : filter incoming traffic to this port Risk factor : Low Nessus ID : 10397

[back to the list of ports]

Warning found on port microsoft-ds (445/tcp)

Here is the list of the SMB shares of this host : E$ IPC$ D$ ADMIN$ C$

This is potentially dangerous as this may help the attack of a potential hacker. Solution : filter incoming traffic to this port Risk factor : Medium Nessus ID : 10395

[back to the list of ports]

Warning found on port microsoft-ds (445/tcp)

The remote host is missing a cumulative security update for Outlook Express which fixes a denial of service vulnerability in the Outlook Express mail client. To exploit this vulnerability, an attacker would need to send a malformed message to a victim on the remote host. The message will crash her version of Outlook, thus preventing her from reading her e-mail. Solution : http://www.microsoft.com/technet/security/bulletin/ms04-018.mspx Risk factor : Medium CVE : CAN-2004-0215 BID : 10711 Nessus ID : 13643

[back to the list of ports]

Warning found on port microsoft-ds (445/tcp)

The remote version of Windows contains a flaw which may allow an attacker to cause it to disclose information over the use of a named pipe through a NULL session. An attacker may exploit this flaw to gain more knowledge about the remote host. Solution : http://www.microsoft.com/technet/security/bulletin/MS05-007.mspx Risk factor : Low CVE : CAN-2005-0051 BID : 12486 Nessus ID : 16337

[back to the list of ports]

Information found on port microsoft-ds (445/tcp)

A CIFS server is running on this port Nessus ID : 11011

[back to the list of ports]

Information found on port microsoft-ds (445/tcp)

- NULL sessions are enabled on the remote host - The SMB tests will be done as 'j'/'******' CVE : CAN-1999-0504, CAN-1999-0506, CVE-2000-0222, CAN-1999-0505, CAN-2002-1117 BID : 494, 990, 11199 Nessus ID : 10394

[back to the list of ports]

Information found on port microsoft-ds (445/tcp)

The remote native lan manager is : Windows 2000 LAN Manager The remote Operating System is : Windows 5.0 The remote SMB Domain Name is : GRUPO Nessus ID : 10785

[back to the list of ports]

Information found on port microsoft-ds (445/tcp)

The remote registry can be accessed remotely using the login / password combination used for the SMB tests. Nessus ID : 10400

[back to the list of ports]

Information found on port microsoft-ds (445/tcp)

Localizador de computadores [Browser] Cliente DHCP [Dhcp] Gerenciador de discos l gicos [dmserver] Cliente DNS [Dnscache] Log de eventos [Eventlog] Sistema de eventos do COM+ [EventSystem] Servidor [lanmanserver] Esta o de trabalho [lanmanworkstation] Servi o auxiliar NetBIOS TCP/IP [LmHosts] Mensageiro [Messenger] Armazenamento remov

vel [NtmsSvc] NVIDIA Display Driver Service [NVSvc] Plug and Play [PlugPlay] Agente de diretiva IPSEC [PolicyAgent] Armazenamento protegido [ProtectedStorage] Gerenciador de conex o de acesso remoto [RasMan] Servi o de registro remoto [RemoteRegistry] Chama de procedimento remoto (RPC) [RpcSs] Gerenciador de contas de seguran a [SamSs] Agendador de tarefas [Schedule] Servi o RunAs [seclogon] Notifica o de eventos de sistema [SENS] Spooler de impress o [Spooler] Telefonia [TapiSrv] Cliente de rastreamento de link distribu do [TrkWks] Testador de instrumenta o de gerenciamento do Windows [WinMgmt] Extens es de driver de instrum. gerenc. do Windows [Wmi] Atualiza es Autom ticas [wuauserv] You should turn off the services you do not use. This list is useful to an attacker, who can make his attack more silent by not portscanning this host. Solution : To prevent the listing of the services for being obtained, you should either have tight login restrictions, so that only trusted users can access your host, and/or you should filter incoming traffic to this port. Risk factor : Low Nessus ID : 10456

[back to the list of ports]

Information found on port microsoft-ds (445/tcp)

The remote Windows 2000 system has Service Pack 4 applied. CVE : CAN-1999-0662 BID : 7930, 8090, 8128, 8154 Nessus ID : 10531

[back to the list of ports]

Information found on port microsoft-ds (445/tcp)

The host Security Identifier (SID) can be obtained remotely. Its value is : 1-5-21-682003330-436374069-854245398 An attacker can use it to obtain the list of the local users of this host Solution : filter the ports 137-139 and 445 Risk factor : Low CVE : CVE-2000-1200 BID : 959 Nessus ID : 10859

[back to the list of ports]

Information found on port microsoft-ds (445/tcp)

The host SID could be used to enumerate the names of the local users of this host. (we only enumerated users name whose ID is between 1000 and 1200 for performance reasons) This gives extra knowledge to an attacker, which is not a good thing : - Administrator account name : Administrador (id 500) - Guest account name : Convidado (id 501) - j (id 1000) Risk factor : None Solution : filter incoming connections this port CVE : CVE-2000-1200 BID : 959 Nessus ID : 10860

[back to the list of ports]

Information found on port microsoft-ds (445/tcp)

The following shares can be accessed as j : - C$ - (readable,writable) + Content of this share : arcsetup.exe Arquivos de programas AUTOEXEC.BAT boot.ini Bootfont.bin CONFIG.SYS Dev-Cpp devicetable.log Documents and Settings IO.SYS j2sdk1.4.2_07 Janes MSDOS.SYS NTDETECT.COM ntldr pagefile.sys Program Files RECYCLER search.log System Volume Information temp WINNT WUTemp - ADMIN$ - (readable,writable) + Content of this share : .. $MSI31Uninstall_KB893803v2$ $NtUninstallKB329115$ $NtUninstallKB823182$ $NtUninstallKB823559$ $NtUninstallKB824105$ $NtUninstallKB826232$ $NtUninstallKB828035$ $NtUninstallKB828741$ $NtUninstallKB828749$ $NtUninstallKB835732$ $NtUninstallKB837001$ $NtUninstallKB839645$ $NtUninstallKB840987$ $NtUninstallKB841356$ $NtUninstallKB841533$ $NtUninstallKB841872$ $NtUninstallKB841873$

$NtUninstallKB842526$ $NtUninstallKB842773$ $NtUninstallKB871250$ $NtUninstallKB873333$ $NtUninstallKB873339$ $NtUninstallKB883939-IE6SP1-20050428.125228$ $NtUninstallKB885250$ $NtUninstallKB885835$ $NtUninstallKB885836$ $NtUninstallKB888113$ $NtUninstallKB890046$ $NtUninstallKB890859$ $NtUninstallKB891781$ $NtUninstallKB893066$ $NtUninstallKB893086$ $NtUninstallKB894320$ $NtUninstallKB896358$ $NtUninstallKB896422$ $NtUninstallKB897715-OE6SP1-20050503.210336$ $NtUninstallQ828026$ Active Setup Log.txt addins AppPatch Areia.bmp Arquivos de instala o do Windows Update Bolhas de sab o.bmp Bruma.bmp Cafezinho.bmp - D$ - (readable,writable) + Content of this share : NVIDIA fortune2.txt _Restore ficha_10x_fisl.pdf ficha_20x_fisl.pdf FlightGear-0.9.4-i486-1.tgz MSOCache NVIDIA fortune2.txt _Restore ficha_10x_fisl.pdf ficha_20x_fisl.pdf FlightGear-0.9.4-i486-1.tgz MSOCache tcc08 NVIDIA fortune2.txt _Restore ficha_10x_fisl.pdf ficha_20x_fisl.pdf FlightGear-0.9.4-i486-1.tgz MSOCache tcc08 NVIDIA fortune2.txt _Restore ficha_10x_fisl.pdf ficha_20x_fisl.pdf FlightGear-0.9.4-i486-1.tgz MSOCache tcc08 NVIDIA fortune2.txt _Restore ficha_10x_fisl.pdf ficha_20x_fisl.pdf FlightGear-0.9.4-i486-1.tgz MSOCache tcc08 NVIDIA fortune2.txt _Restore ficha_10x_fisl.pdf ficha_20x_fisl.pdf FlightGear-0.9.4-i486-1.tgz MSOCache tcc08 NVIDIA fortune2.txt _Restore ficha_10x_fisl.pdf ficha_20x_fisl.pdf FlightGear-0.9.4-i486-1.tgz MSOCache tcc08 NVIDIA fortune2.txt _Restore ficha_10x_fisl.pdf ficha_20x_fisl.pdf FlightGear-0.9.4-i486-1.tgz MSOCache tcc08 NVIDIA fortune2.txt - E$ - (readable,writable) + Content of this share : mp3 drivers jogos _Restore newFAZER.sxw tcc2-051-31-8-2k5-recup.doc tcc08 mp3 drivers jogos _Restore newFAZER.sxw tcc2-051-31-8-2k5-recup.doc tcc08 ~WRL3338.tmp mp3 drivers jogos _Restore newFAZER.sxw tcc2-051-31-8-2k5-recup.doc tcc08 ~WRL3338.tmp mp3 drivers jogos _Restore newFAZER.sxw tcc2-051-31-8-2k5-recup.doc tcc08 ~WRL3338.tmp mp3 drivers jogos _Restore newFAZER.sxw tcc2-051-31-8-2k5-recup.doc tcc08 ~WRL3338.tmp mp3 drivers jogos _Restore newFAZER.sxw tcc2-051-31-8-2k5-recup.doc tcc08 ~WRL3338.tmp mp3 drivers jogos _Restore newFAZER.sxw tcc2-051-31-8-2k5-recup.doc tcc08 ~WRL3338.tmp mp3 drivers jogos _Restore newFAZER.sxw tcc2-051-31-8-2k5-recup.doc tcc08 ~WRL3338.tmp mp3 drivers jogos _Restore newFAZER.sxw tcc2-051-31-8-2k5-recup.doc tcc08 ~WRL3338.tmp mp3 drivers jogos _Restore newFAZER.sxw tcc2-051-31-8-2k5-recup.doc CVE : CAN-1999-0519, CAN-1999-0520 BID : 8026 Nessus ID : 10396

[back to the list of ports]

Information found on port microsoft-ds (445/tcp)

The registry key HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\CachedLogonsCount is non-null. It means that the remote host locally caches the passwords of the users when they log in, in order to continue to allow the users to log in in the case of the failure of the PDC. Solution : use regedt32 and set the value of this key to 0 Risk factor : Low Nessus ID : 11457

[back to the list of ports]

Information found on port blackjack (1025/tcp)

Distributed Computing Environment (DCE) services running on the remote host can be enumerated by connecting on port 135 and doing the appropriate queries. An attacker may use this fact to gain more knowledge about the remote host. Here is the list of DCE services running on this port: UUID: 1ff70682-0a51-30e8-076d-740be8cee98b, version 1 Endpoint: ncacn_ip_tcp:192.168.1.3[1025] Named pipe : atsvc Win32 service or process : mstask.exe Description : Scheduler service UUID: 378e52b0-c0a9-11cf-822d-00aa0051e40f, version 1 Endpoint: ncacn_ip_tcp:192.168.1.3[1025] Solution : filter incoming traffic to this port. Risk factor : Low Nessus ID : 10736

[back to the list of ports]

Information found on port general/udp

For your information, here is the traceroute to 192.168.1.3 : 192.168.1.4 192.168.1.3 Nessus ID : 10287

[back to the list of ports]

Warning found on port general/tcp

The remote host does not discard TCP SYN packets which have the FIN flag set. Depending on the kind of firewall you are using, an attacker may use this flaw to bypass its rules. See also : http://archives.neohapsis.com/archives/bugtraq/2002-10/0266.html http://www.kb.cert.org/vuls/id/464113 Solution : Contact your vendor for a patch Risk factor : Medium BID : 7487 Nessus ID : 11618

[back to the list of ports]

Warning found on port general/tcp

The remote host accepts loose source routed IP packets. The feature was designed for testing purpose. An attacker may use it to circumvent poorly designed IP filtering and exploit another flaw. However, it is not dangerous by itself. Solution : drop source routed packets on this host or on other ingress routers or firewalls. Risk factor : Low Nessus ID : 11834

[back to the list of ports]

Information found on port general/tcp

192.168.1.3 resolves as luiza.juarezdc.org. Nessus ID : 12053

[back to the list of ports]

Information found on port general/tcp

The remote host is running Microsoft Windows 2000 Professional Service Pack 4 (Portugu s (Brasil)) Nessus ID : 11936

[back to the list of ports]

Information found on port general/tcp

The following users are in the local administrator group : . Administrador (User) . j (User) You should make sure that only the proper users are member of this group Risk factor : Low Nessus ID : 10902

[back to the list of ports]

Warning found on port netbios-ns (137/udp)

The following 7 NetBIOS names have been gathered : CAVERA = This is the computer name registered for workstation services by a WINS client.

GRUPO = Workgroup / Domain name CAVERA = Computer name GRUPO = Workgroup / Domain name (part of the Browser elections) CAVERA = This is the current logged in user registered for this workstation. GRUPO __MSBROWSE__ The remote host has the following MAC address on its adapter : 00:e0:7d:de:c9:61 If you do not want to allow everyone to find the NetBios name of your computer, you should filter incoming traffic to this port. Risk factor : Medium CVE : CAN-1999-0621 Nessus ID : 10150

[back to the list of ports]

Warning found on port general/icmp

The remote host answers to an ICMP timestamp request. This allows an attacker to know the date which is set on your machine. This may help him to defeat all your time based authentication protocols. Solution : filter out the ICMP timestamp requests (13), and the outgoing ICMP timestamp replies (14). Risk factor : Low CVE : CAN-1999-0524 Nessus ID : 10114

[Next host : 192.168.1.4]

This file was generated by Nessus, the open-sourced security scanner.

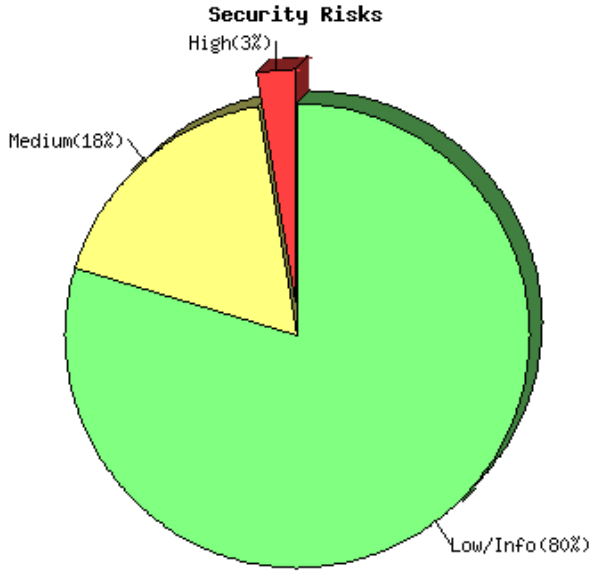
ANEXO E – Relatório geral do *nessus* no dia 22 de novembro de 2005.

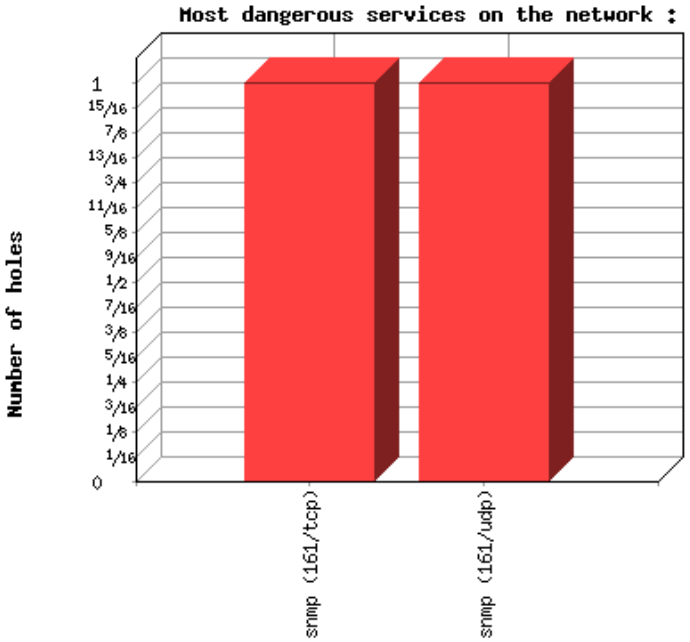
Nessus Report

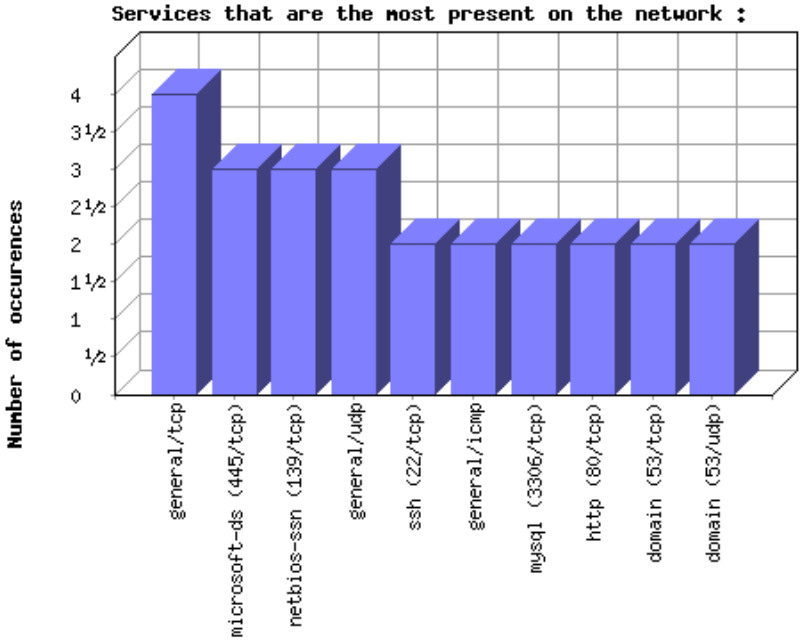
The Nessus Security Scanner was used to assess the security of 4 hosts

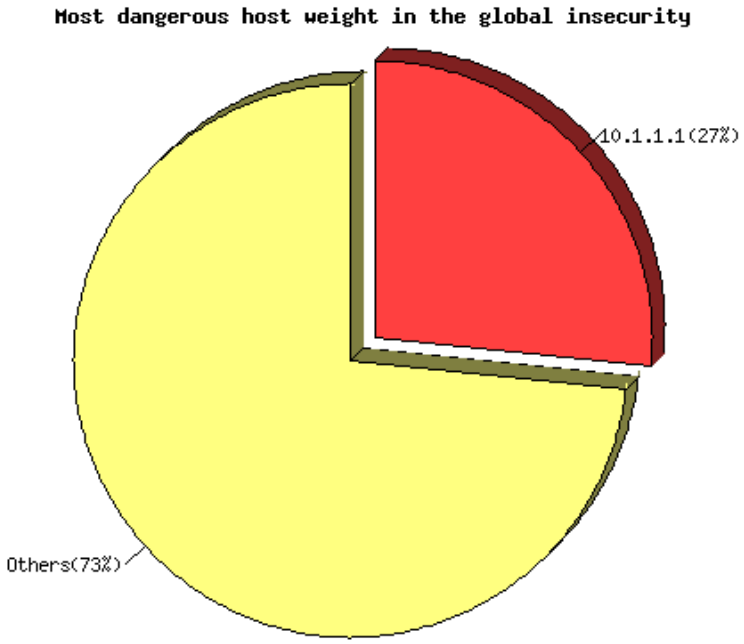
- **2 security holes have been found**
- **13 security warnings have been found**
- **59 security notes have been found**

Part I : Graphical Summary :




Most dangerous services on the network :
Number of holes
1
15/16
7/8
13/16
3/4
11/16
5/8
9/16
1/2
7/16
3/8
5/16
1/4
3/16
1/8
1/16
0
snmp (161/tcp)
snmp (161/udp)


Services that are the most present on the network :
Number of occurences
4
3 1/2
3
2 1/2
2
1 1/2
1
1/2
0
general/tcp
microsoft-ds (445/tcp)
netbios-ssn (139/tcp)
general/udp
ssh (22/tcp)
general/icmp
mysql (3306/tcp)
http (80/tcp)
domain (53/tcp)
domain (53/udp)



Part II. Results, by host :

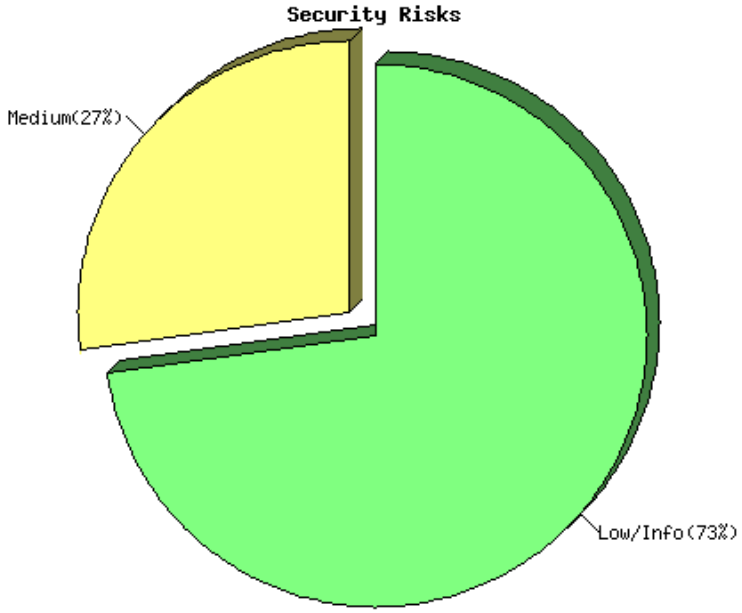
	10.1.1.1	**(found 2 security holes)**
	10.1.1.24	**(found 3 security warnings)**
.1	192.168.1	**(found 5 security warnings)**
.4	192.168.1	**(found 3 security warnings)**

This file was generated by Nessus, the open-sourced security scanner.

ANEXO F – Relatório do *nessus* para o host de bastião no endereço IP 10.1.1.24 no dia 22 de novembro de 2005.

10.1.1.24

Repartition of the level of the security problems :



[Next host : 192.168.1.1]

List of open ports :

- *domain (53/tcp) (Security warnings found)*
- *mysql (3306/tcp) (Security notes found)*
- *general/tcp (Security warnings found)*
- *general/udp (Security notes found)*
- *domain (53/udp) (Security notes found)*
- *general/icmp (Security warnings found)*

[back to the list of ports]

Warning found on port domain (53/tcp)

The remote name server allows recursive queries to be performed by the host running nessusd. If this is your internal nameserver, then forget this warning. If you are probing a remote nameserver, then it allows anyone to use it to resolve third parties names (such as www.nessus.org). This allows hackers to do cache poisoning attacks against this nameserver. If the host allows these recursive queries via UDP, then the host can be used to 'bounce' Denial of Service attacks against another network or system. See also : http://www.cert.org/advisories/CA-1997-22.html Solution : Restrict recursive queries to the hosts that should use this nameserver (such as those of the LAN connected to it). If you are using bind 8, you can do this by using the instruction 'allow-recursion' in the 'options' section of your named.conf If you are using bind 9, you can define a grouping of internal addresses using the 'acl' command Then, within the options block, you can explicitly state: 'allow-recursion {

hosts_defined_in_acl }' For more info on Bind 9 administration (to include recursion), see: http://www.nominum.com/content/documents/bind9arm.pdf If you are using another name server, consult its documentation. Risk factor : High CVE : CVE-1999-0024 BID : 136, 678 Nessus ID : 10539

[back to the list of ports]

Information found on port domain (53/tcp)

BIND 'NAMED' is an open-source DNS server from ISC.org. Many proprietary DNS servers are based on BIND source code. The BIND based NAMED servers (or DNS servers) allow remote users to query for version and type information. The query of the CHAOS TXT record 'version.bind', will typically prompt the server to send the information back to the querying source. The remote bind version is : 9.3.1 Solution : Using the 'version' directive in the 'options' section will block the 'version.bind' query, but it will not log such attempts. Nessus ID : 10028

[back to the list of ports]

Information found on port domain (53/tcp)

A DNS server is running on this port. If you do not use it, disable it. Risk factor : Low Nessus ID : 11002

[back to the list of ports]

Information found on port mysql (3306/tcp)

An unknown service is running on this port. It is usually reserved for MySQL Nessus ID : 10330

[back to the list of ports]

Warning found on port general/tcp

The remote host does not discard TCP SYN packets which have the FIN flag set. Depending on the kind of firewall you are using, an attacker may use this flaw to bypass its rules. See also : http://archives.neohapsis.com/archives/bugtraq/2002-10/0266.html http://www.kb.cert.org/vuls/id/464113 Solution : Contact your vendor for a patch Risk factor : Medium BID : 7487 Nessus ID : 11618

[back to the list of ports]

Information found on port general/tcp

The remote host is running Clark Connect Firewall Nessus ID : 11936

[back to the list of ports]

Information found on port general/udp

For your information, here is the traceroute to 10.1.1.24 : 192.168.1.4 10.1.1.24 Nessus ID : 10287

[back to the list of ports]

Information found on port domain (53/udp)

A DNS server is running on this port. If you do not use it, disable it. Risk factor : Low Nessus ID : 11002

[back to the list of ports]

Information found on port domain (53/udp)

The remote name server could be fingerprinted as being : ISC BIND 9.2.3 Nessus ID : 11951

[back to the list of ports]

Information found on port domain (53/udp)

The remote DNS server answers to queries for third party domains which do not have the recursion bit set. This may allow a remote attacker to determine which domains have recently been resolved via this name server, and therefore which hosts have been recently visited. For instance, if an attacker was interested in whether your company utilizes the online services of a particular financial institution, they would be able to use this attack to build a statistical model regarding company usage of aforementioned financial institution. Of course, the attack can also be used to find B2B partners, web-surfing patterns, external mail servers, and more... For a much more detailed discussion of the potential risks of allowing DNS cache information to be queried anonymously, please see: http://community.sidestep.pt/~luis/DNS-Cache-Snooping/DNS_Cache_Snooping_1.1.pdf Risk factor : Low Nessus ID : 12217

[back to the list of ports]

Warning found on port general/icmp

The remote host answers to an ICMP timestamp request. This allows an attacker to know the date which is set on your machine. This may help him to defeat all your time based authentication protocols. Solution : filter out the ICMP timestamp requests (13), and the outgoing ICMP timestamp replies (14). Risk factor : Low CVE : CAN-1999-0524 Nessus ID : 10114

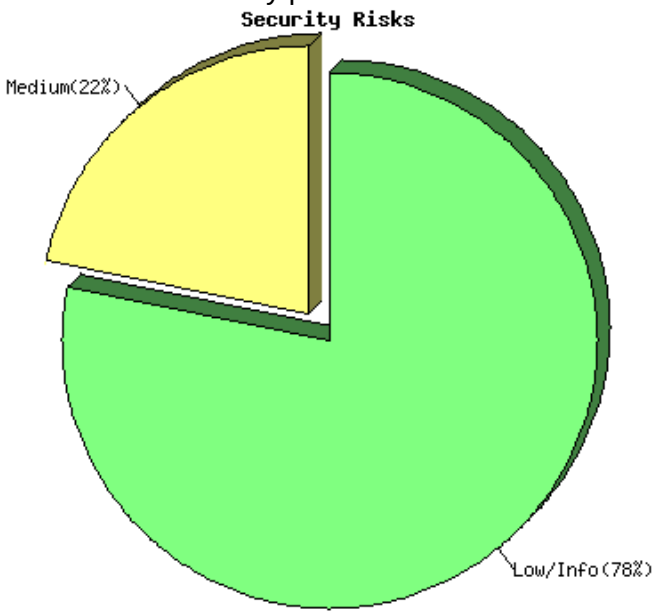
[Next host : 192.168.1.1]

This file was generated by Nessus, the open-sourced security scanner.

ANEXO G – Relatório do *nessus* para o host de bastião no endereço IP 192.168.1.1 no dia 22 de novembro de 2005.

192.168.1.1

Repartition of the level of the security problems :



[Next host : 192.168.1.4]

List of open ports :

- *ssh (22/tcp)* (Security notes found)
- *domain (53/tcp)* (Security notes found)
- *http (80/tcp)* (Security notes found)
- *netbios-ssn (139/tcp)* (Security notes found)
- *microsoft-ds (445/tcp)* (Security warnings found)
- *squid-http (3128/tcp)* (Security warnings found)
- *mysql (3306/tcp)* (Security notes found)
- *general/udp* (Security notes found)
- *general/tcp* (Security warnings found)
- *bootps (67/udp)* (Security notes found)
- *general/icmp* (Security warnings found)

[back to the list of ports]

Information found on port ssh (22/tcp)

An ssh server is running on this port Nessus ID : 10330

[back to the list of ports]

Information found on port ssh (22/tcp)

Remote SSH version : SSH-2.0-OpenSSH_4.2 Remote SSH supported

authentication : publickey,password,keyboard-interactive Nessus ID : 10267

[back to the list of ports]

Information found on port domain (53/tcp)

BIND 'NAMED' is an open-source DNS server from ISC.org. Many proprietary DNS servers are based on BIND source code. The BIND based NAMED servers (or DNS servers) allow remote users to query for version and type information. The query of the CHAOS TXT record 'version.bind', will typically prompt the server to send the information back to the querying source. The remote bind version is : 9.3.1 Solution : Using the 'version' directive in the 'options' section will block the 'version.bind' query, but it will not log such attempts. Nessus ID : 10028

[back to the list of ports]

Information found on port http (80/tcp)

A web server is running on this port Nessus ID : 10330

[back to the list of ports]

Information found on port http (80/tcp)

The following directories were discovered: /cgi-bin, /docs, /icons, /session, /tests While this is not, in and of itself, a bug, you should manually inspect these directories to ensure that they are in compliance with company security standards Nessus ID : 11032

[back to the list of ports]

Information found on port http (80/tcp)

The remote web server type is : Apache/1.3.33 (Unix) mod_ssl/2.8.24 OpenSSL/0.9.7g PHP/4.4.0 Solution : You can set the directive 'ServerTokens Prod' to limit the information emanating from the server in its response headers. Nessus ID : 10107

[back to the list of ports]

Information found on port netbios-ssn (139/tcp)

An SMB server is running on this port Nessus ID : 11011

[back to the list of ports]

Warning found on port microsoft-ds (445/tcp)

Here is the browse list of the remote host : SERVER (os: 0.0) This is potentially dangerous as this may help the attack of a potential hacker by giving him extra targets to check for Solution : filter incoming traffic to this port Risk factor : Low Nessus ID : 10397

[back to the list of ports]

Information found on port microsoft-ds (445/tcp)

A CIFS server is running on this port Nessus ID : 11011

[back to the list of ports]

Information found on port microsoft-ds (445/tcp)

- NULL sessions are enabled on the remote host CVE : CAN-1999-0504, CAN-1999-0506, CVE-2000-0222, CAN-1999-0505, CAN-2002-1117 BID : 494,

990, 11199 Nessus ID : 10394

[back to the list of ports]

Information found on port microsoft-ds (445/tcp)

The remote native lan manager is : Samba 3.0.20 The remote Operating System is : Unix The remote SMB Domain Name is : TABAJARA Nessus ID : 10785

[back to the list of ports]

Warning found on port squid-http (3128/tcp)

The misconfigured proxy accepts requests coming from anywhere. This allows attackers to gain some anonymity when browsing some sensitive sites using your proxy, making the remote sites think that the requests come from your network. Solution: Reconfigure the remote proxy so that it only accepts requests coming from inside your network. Risk factor : Low / Medium Nessus ID : 10195

[back to the list of ports]

Warning found on port squid-http (3128/tcp)

The proxy accepts gopher:// requests. Gopher is an old network protocol which predates HTTP and is nearly unused today. As a result, gopher-compatible software is generally less audited and more likely to contain security bugs than others. By making gopher requests, an attacker may evade your firewall settings, by making connections to port 70, or may even exploit arcane flaws in this protocol to gain more privileges on this host (see the attached CVE id for such an example). Solution : reconfigure your proxy so that it refuses gopher requests. Risk factor : Medium CVE : CAN-2002-0371 BID : 4930 Nessus ID : 11305

[back to the list of ports]

Information found on port squid-http (3128/tcp)

A web server is running on this port Nessus ID : 10330

[back to the list of ports]

Information found on port squid-http (3128/tcp)

An HTTP proxy is running on this port Nessus ID : 10330

[back to the list of ports]

Information found on port squid-http (3128/tcp)

There might be a caching proxy on the way to this web server Nessus ID : 11040

[back to the list of ports]

Information found on port mysql (3306/tcp)

An unknown service is running on this port. It is usually reserved for MySQL Nessus ID : 10330

[back to the list of ports]

Information found on port general/udp

For your information, here is the traceroute to 192.168.1.1 : 192.168.1.4

192.168.1.1 Nessus ID : 10287

[back to the list of ports]

Warning found on port general/tcp

The remote host does not discard TCP SYN packets which have the FIN flag set. Depending on the kind of firewall you are using, an attacker may use this flaw to bypass its rules. See also : http://archives.neohapsis.com/archives/bugtraq/2002-10/0266.html http://www.kb.cert.org/vuls/id/464113 Solution : Contact your vendor for a patch Risk factor : Medium BID : 7487 Nessus ID : 11618

[back to the list of ports]

Information found on port general/tcp

The remote host is running Clark Connect Firewall Nessus ID : 11936

[back to the list of ports]

Information found on port general/tcp

192.168.1.1 resolves as localhost. Nessus ID : 12053

[back to the list of ports]

Information found on port bootps (67/udp)

Here is the information we could gather from the remote DHCP server. This allows an attacker on your local network to gain information about it easily : Master DHCP server of this network : 0.0.0.0 IP address the DHCP server would attribute us : 192.168.1.4 DHCP server(s) identifier = 192.168.1.1 netmask = 255.255.255.0 router = 192.168.1.1 domain name server(s) = 192.168.1.1 host name = juarezdc domain name = juarezdc.org broadcast address = 192.168.1.255 Solution : remove the options that are not in use in your DHCP server Risk factor : Low Nessus ID : 10663

[back to the list of ports]

Warning found on port general/icmp

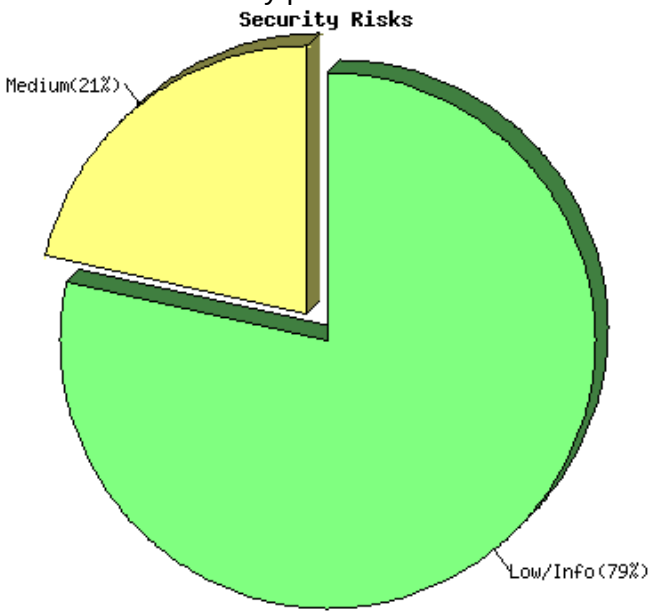
The remote host answers to an ICMP timestamp request. This allows an attacker to know the date which is set on your machine. This may help him to defeat all your time based authentication protocols. Solution : filter out the ICMP timestamp requests (13), and the outgoing ICMP timestamp replies (14). Risk factor : Low CVE : CAN-1999-0524 Nessus ID : 10114

[Next host : 192.168.1.4]

ANEXO H – Relatório do *nessus* para o servidor no endereço IP 192.168.1.4 no dia 22 de novembro de 2005.

192.168.1.4

Repartition of the level of the security problems :



[Back to the index]

List of open ports :

- *smtp (25/tcp) (Security notes found)*
- *netbios-ssn (139/tcp) (Security notes found)*
- *microsoft-ds (445/tcp) (Security warnings found)*
- *nessus (1241/tcp) (Security warnings found)*
- *x11 (6000/tcp) (Security notes found)*
- *netbios-ns (137/udp) (Security warnings found)*
- *netbios-dgm (138/udp)*
- *bootpc (68/udp)*
- *general/tcp (Security notes found)*

[back to the list of ports]

Information found on port smtp (25/tcp)

An SMTP server is running on this port Here is its banner : 220 caverinha.juarezdc.orgz ESMTP Sendmail 8.13.4/8.13.4; Tue, 22 Nov 2005 01:41:25 -0200 Nessus ID : 10330

[back to the list of ports]

Information found on port smtp (25/tcp)

This server could be fingerprinted as being Sendmail 8.11.6/8.11.6 -84-

Nessus ID : 11421

[back to the list of ports]

Information found on port smtp (25/tcp)

Remote SMTP server banner : 220 caverinha.juarezdc.orgz ESMTP Sendmail 8.13.4/8.13.4; Tue, 22 Nov 2005 01:42:00 -0200 This is probably: Sendmail version 8.13.4 Nessus ID : 10263

[back to the list of ports]

Information found on port netbios-ssn (139/tcp)

An SMB server is running on this port Nessus ID : 11011

[back to the list of ports]

Warning found on port microsoft-ds (445/tcp)

Here is the browse list of the remote host : CAVERINHA (os: 0.0) This is potentially dangerous as this may help the attack of a potential hacker by giving him extra targets to check for Solution : filter incoming traffic to this port Risk factor : Low Nessus ID : 10397

[back to the list of ports]

Information found on port microsoft-ds (445/tcp)

A CIFS server is running on this port Nessus ID : 11011

[back to the list of ports]

Information found on port microsoft-ds (445/tcp)

- NULL sessions are enabled on the remote host CVE : CAN-1999-0504, CAN-1999-0506, CVE-2000-0222, CAN-1999-0505, CAN-2002-1117 BID : 494, 990, 11199 Nessus ID : 10394

[back to the list of ports]

Information found on port microsoft-ds (445/tcp)

The remote native lan manager is : Samba 3.0.20 The remote Operating System is : Unix The remote SMB Domain Name is : MYGROUP Nessus ID : 10785

[back to the list of ports]

Warning found on port nessus (1241/tcp)

A Nessus Daemon is listening on this port. Nessus ID : 10147

[back to the list of ports]

Information found on port nessus (1241/tcp)

A TLSv1 server answered on this port Nessus ID : 10330

[back to the list of ports]

Information found on port nessus (1241/tcp)

Here is the TLSv1 server certificate: Certificate: Data: Version: 3 (0x2) Serial Number: 1 (0x1) Signature Algorithm: md5WithRSAEncryption Issuer: C=BR, ST=RS, L=POA, O=Nessus Users United, OU=Certification Authority for caverinha, CN=caverinha/emailAddress=ca@caverinha Validity Not Before: Nov 15 19:32:01 2005 GMT Not After : Nov 15 19:32:01 2006 GMT Subject:

C=BR, ST=RS, L=POA, O=Nessus Users United, OU=Server certificate for caverinha, CN=caverinha/emailAddress=nessusd@caverinha Subject Public Key Info: Public Key Algorithm: rsaEncryption RSA Public Key: (1024 bit) Modulus (1024 bit): 00:d8:53:35:8d:df:20:71:6a:8c:a1:46:93:0c:7b: 27:37:7c:1c:d0:4b:37:f0:a6:b6:d0:97:4a:d5:9e: ac:01:9e:6d:55:1d:06:c7:28:97:99:1c:69:e1:d9: 42:65:1e:17:63:8f:11:99:2a:29:59:e1:88:94:d3: b9:c3:17:80:69:97:34:5c:8e:57:df:95:dd:24:5c: 13:ed:f9:93:a1:f6:a2:be:23:28:64:b3:40:75:43: 4a:19:df:40:ed:a1:76:09:49:7b:fc:0b:7c:ed:77: b9:9d:05:37:19:50:fe:e8:d8:08:7c:8a:70:ad:71: e0:95:81:5f:95:22:60:5d:6f Exponent: 65537 (0x10001) X509v3 extensions: Netscape Cert Type: SSL Server X509v3 Key Usage: Digital Signature, Non Repudiation, Key Encipherment Netscape Comment: OpenSSL Generated Certificate X509v3 Subject Key Identifier: 86:4D:47:83:E2:8F:D0:76:5A:66:79:82:55:3C:5C:4D:13:35:46:80 X509v3 Authority Key Identifier: keyid:91:A0:70:28:EE:1A:A1:1A:9A:03:57:F1:71:3C:D5:7E:46:86:45:CD DirName:/C=BR/ST=RS/L=POA/O=Nessus Users United/OU=Certification Authority for caverinha/CN=caverinha/emailAddress=ca@caverinha serial:A7:E8:5F:D7:90:E6:E2:7C X509v3 Subject Alternative Name: email:nessusd@caverinha X509v3 Issuer Alternative Name: <EMPTY> Signature Algorithm: md5WithRSAEncryption 42:59:a5:4f:fa:20:68:21:16:10:30:94:87:7d:19:4d:5a:6b: 80:35:9f:ca:1a:92:b8:f4:ed:c2:ee:33:35:98:09:ec:88:4b: 92:59:3c:28:83:d6:9a:9f:25:50:b3:3b:86:27:68:29:43:42: 24:82:3e:0a:05:12:93:e1:90:44:c9:6f:a4:8a:ef:7d:a4:57: 29:ee:f4:96:b7:1d:23:95:00:c0:49:e5:a5:7e:9f:9b:0e:65: 2c:60:cd:29:8f:98:d6:a0:5f:6b:ae:a7:c0:3d:e4:e9:6e:2c: 6e:ba:6a:be:77:d4:39:d4:3a:7e:61:ad:22:75:8d:f7:17:b6: 33:47 This TLSv1 server does not accept SSLv2 connections. This TLSv1 server does not accept SSLv3 connections. Nessus ID : 10863

[back to the list of ports]

Information found on port x11 (6000/tcp)

This X server does *not* allow any client to connect to it however it is recommended that you filter incoming connections to this port as attacker may send garbage data and slow down your X session or even kill the server. Here is the server version : 11.0 Here is the message we received : No protocol specified Solution : filter incoming connections to ports 6000-6009 Risk factor : Low Nessus ID : 10407

[back to the list of ports]

Warning found on port netbios-ns (137/udp)

The following 7 NetBIOS names have been gathered : CAVERINHA = This is the computer name registered for workstation services by a WINS client. CAVERINHA = This is the current logged in user registered for this workstation. CAVERINHA = Computer name __MSBROWSE__ MYGROUP = Workgroup / Domain name MYGROUP MYGROUP = Workgroup / Domain name (part of the Browser elections) . This SMB server seems to be a SAMBA server (this is not a security risk, this is for your information). This can be told because this server claims to have a null MAC address If you do not want to allow everyone to find the NetBios name of your computer, you should filter incoming traffic to this

port. Risk factor : Medium CVE : CAN-1999-0621 Nessus ID : 10150

[back to the list of ports]

Information found on port general/tcp

192.168.1.4 resolves as juarezdc.juarezdc.org. Nessus ID : 12053

This file was generated by Nessus, the open-sourced security scanner.

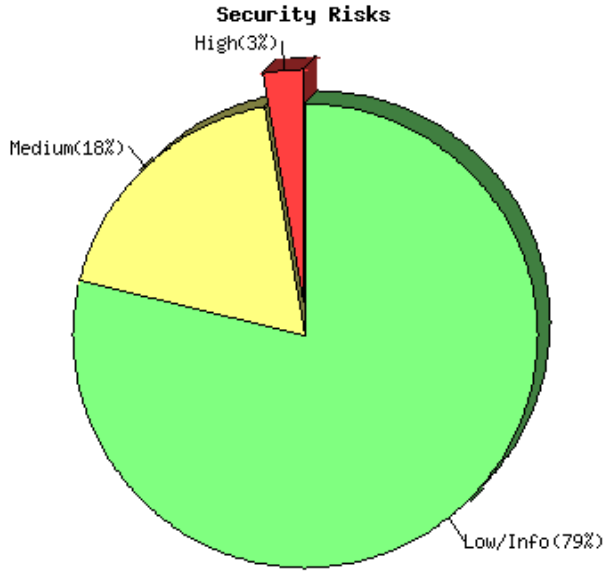
ANEXO I – Relatório geral do *nessus* no dia 10 de dezembro de 2005.

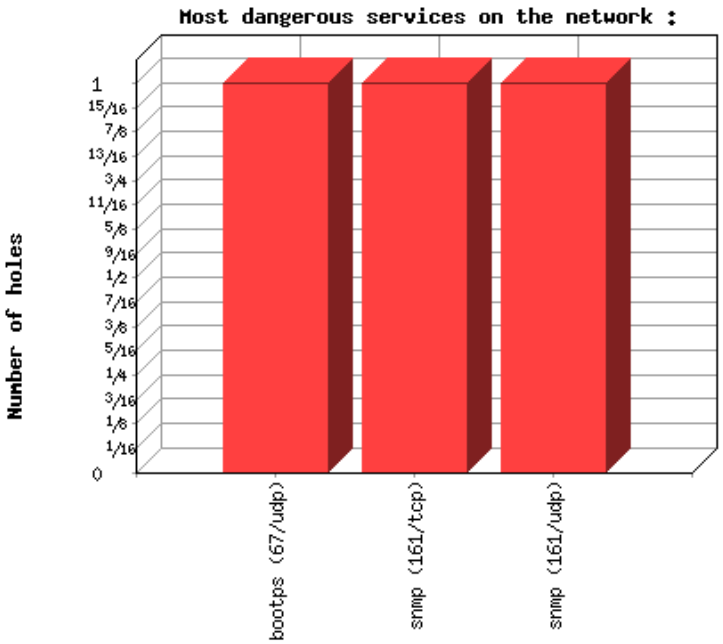
Nessus Report

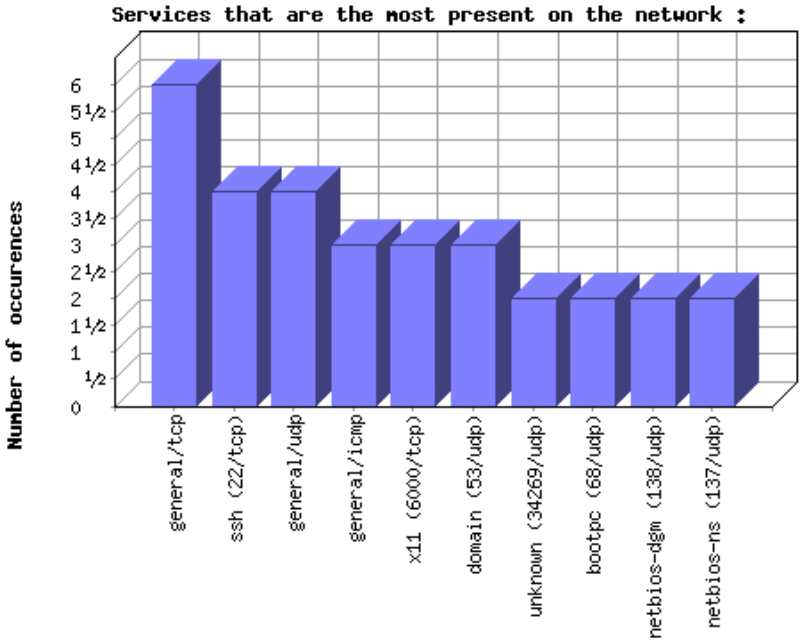
The Nessus Security Scanner was used to assess the security of 6 hosts

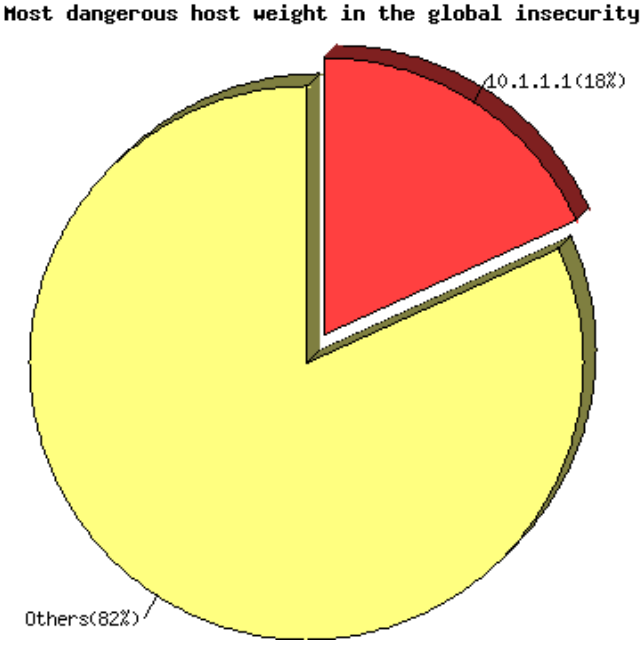
- **3 security holes have been found**
- **19 security warnings have been found**
- **82 security notes have been found**

Part I : Graphical Summary :









Part II. Results, by host :

10.1.1.1	**(found 2 security holes)**
10.1.1.5	**(found 3 security warnings)**
10.1.1.24	**(found 3 security warnings)**
127.0.0.1	**(found 3 security warnings)**
192.168.1.1	**(found 1 security hole)**

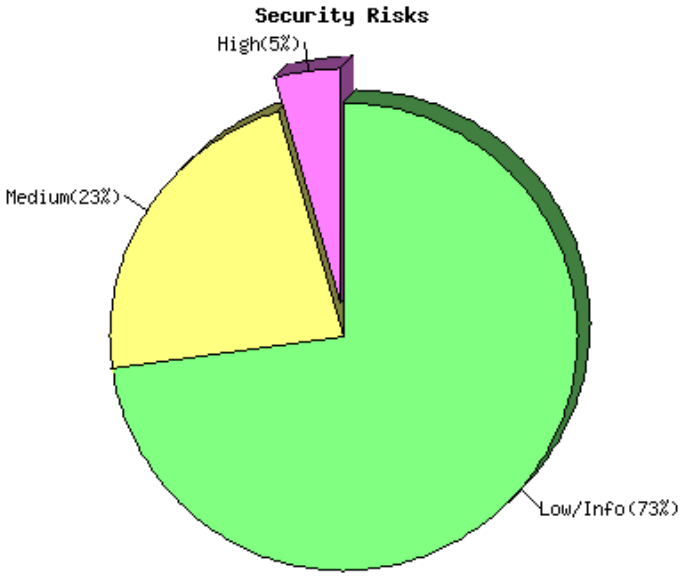
192.168.1.4 (**found 3 security warnings**)

This file was generated by Nessus, the open-sourced security scanner.

ANEXO J – Relatório do *nessus* para o host de bastião no endereço IP 192.168.1.1 no dia 10 de dezembro de 2005.

192.168.1.1

Repartition of the level of the security problems :



[Next host : 192.168.1.4]

List of open ports :

- *ssh (22/tcp)* *(Security notes found)*
- *domain (53/tcp)* *(Security warnings found)*
- *squid-http (3128/tcp)* *(Security warnings found)*
- *mysql (3306/tcp)* *(Security notes found)*
- *general/udp* *(Security notes found)*
- *domain (53/udp)* *(Security notes found)*
- *general/tcp* *(Security warnings found)*
- *bootps (67/udp)* *(Security hole found)*
- *general/icmp* *(Security warnings found)*

[back to the list of ports]

Information found on port ssh (22/tcp)

An ssh server is running on this port Nessus ID : 10330

[back to the list of ports]

Information found on port ssh (22/tcp)

Remote SSH version : SSH-2.0-OpenSSH_4.2 Remote SSH supported authentication : publickey,password,keyboard-interactive Nessus ID : 10267

[back to the list of ports]

Information found on port ssh (22/tcp)

The remote SSH daemon supports the following versions of the SSH protocol : . 1.99 . 2.0 SSHv2 host key fingerprint : 9e:e6:3f:e8:47:1b:59:48:97:2b:8e:15:ca:1f:29:a9 Nessus ID : 10881

[back to the list of ports]

Warning found on port domain (53/tcp)

The remote name server allows recursive queries to be performed by the host running nessusd. If this is your internal nameserver, then forget this warning. If you are probing a remote nameserver, then it allows anyone to use it to resolve third parties names (such as www.nessus.org). This allows hackers to do cache poisoning attacks against this nameserver. If the host allows these recursive queries via UDP, then the host can be used to 'bounce' Denial of Service attacks against another network or system. See also : http://www.cert.org/advisories/CA-1997-22.html Solution : Restrict recursive queries to the hosts that should use this nameserver (such as those of the LAN connected to it). If you are using bind 8, you can do this by using the instruction 'allow-recursion' in the 'options' section of your named.conf If you are using bind 9, you can define a grouping of internal addresses using the 'acl' command Then, within the options block, you can explicitly state: 'allow-recursion { hosts_defined_in_acl }' For more info on Bind 9 administration (to include recursion), see: http://www.nominum.com/content/documents/bind9arm.pdf If you are using another name server, consult its documentation. Risk factor : High CVE : CVE-1999-0024 BID : 136, 678 Nessus ID : 10539

[back to the list of ports]

Information found on port domain (53/tcp)

BIND 'NAMED' is an open-source DNS server from ISC.org. Many proprietary DNS servers are based on BIND source code. The BIND based NAMED servers (or DNS servers) allow remote users to query for version and type information. The query of the CHAOS TXT record 'version.bind', will typically prompt the server to send the information back to the querying source. The remote bind version is : 9.3.1 Solution : Using the 'version' directive in the 'options' section will block the 'version.bind' query, but it will not log such attempts. Nessus ID : 10028

[back to the list of ports]

Information found on port domain (53/tcp)

A DNS server is running on this port. If you do not use it, disable it. Risk factor : Low Nessus ID : 11002

[back to the list of ports]

Warning found on port squid-http (3128/tcp)

The misconfigured proxy accepts requests coming from anywhere. This allows attackers to gain some anonymity when browsing some sensitive sites using your proxy, making the remote sites think that the requests come from your network. Solution: Reconfigure the remote proxy so that it only accepts requests coming from inside your network. Risk factor : Low / Medium Nessus ID : 10195

[back to the list of ports]

Warning found on port squid-http (3128/tcp)

The proxy accepts gopher:// requests. Gopher is an old network protocol which predates HTTP and is nearly unused today. As a result, gopher-compatible software is generally less audited and more likely to contain security bugs than others. By making gopher requests, an attacker may evade your firewall settings, by making connections to port 70, or may even exploit arcane flaws in this protocol to gain more privileges on this host (see the attached CVE id for such an example). Solution : reconfigure your proxy so that it refuses gopher requests. Risk factor : Medium CVE : CAN-2002-0371 BID : 4930 Nessus ID : 11305

[back to the list of ports]

Information found on port squid-http (3128/tcp)

A web server is running on this port Nessus ID : 10330

[back to the list of ports]

Information found on port squid-http (3128/tcp)

An HTTP proxy is running on this port Nessus ID : 10330

[back to the list of ports]

Information found on port squid-http (3128/tcp)

There might be a caching proxy on the way to this web server Nessus ID : 11040

[back to the list of ports]

Information found on port squid-http (3128/tcp)

The remote web server type is : squid/2.5.STABLE11 Solution : We recommend that you configure (if possible) your web server to return a bogus Server header in order to not leak information. Nessus ID : 10107

[back to the list of ports]

Information found on port mysql (3306/tcp)

An unknown service is running on this port. It is usually reserved for MySQL Nessus ID : 10330

[back to the list of ports]

Information found on port general/udp

For your information, here is the traceroute to 192.168.1.1 : 192.168.1.4 192.168.1.1 Nessus ID : 10287

[back to the list of ports]

Information found on port domain (53/udp)

A DNS server is running on this port. If you do not use it, disable it. Risk factor : Low Nessus ID : 11002

[back to the list of ports]

Information found on port domain (53/udp)

The remote name server could be fingerprinted as being : ISC BIND 9.2.3 Nessus ID : 11951

[back to the list of ports]

Information found on port domain (53/udp)

The remote DNS server answers to queries for third party domains which do not have the recursion bit set. This may allow a remote attacker to determine which domains have recently been resolved via this name server, and therefore which hosts have been recently visited. For instance, if an attacker was interested in whether your company utilizes the online services of a particular financial institution, they would be able to use this attack to build a statistical model regarding company usage of aforementioned financial institution. Of course, the attack can also be used to find B2B partners, web-surfing patterns, external mail servers, and more... For a much more detailed discussion of the potential risks of allowing DNS cache information to be queried anonymously, please see: http://community.sidestep.pt/~luis/DNS-Cache-Snooping/DNS_Cache_Snooping_1.1.pdf Risk factor : Low Nessus ID : 12217

[back to the list of ports]

Warning found on port general/tcp

The remote host does not discard TCP SYN packets which have the FIN flag set. Depending on the kind of firewall you are using, an attacker may use this flaw to bypass its rules. See also : http://archives.neohapsis.com/archives/bugtraq/2002-10/0266.html http://www.kb.cert.org/vuls/id/464113 Solution : Contact your vendor for a patch Risk factor : Medium BID : 7487 Nessus ID : 11618

[back to the list of ports]

Information found on port general/tcp

The remote host is running Clark Connect Firewall Nessus ID : 11936

[back to the list of ports]

Information found on port general/tcp

192.168.1.1 resolves as server.juarezdc.org. Nessus ID : 12053

[back to the list of ports]

Vulnerability found on port bootps (67/udp)

The remote host is running a DHCP server. If the remote server is ISC-DHCPd, make sure you are running the latest version, as several flaws affect older versions and may allow an attacker to gain root on this host *** Note that Nessus did not check for the presence of the *** flaws, so this might be a false positive See also : http://www.cert.org/advisories/CA-2003-01.html http://www.cert.org/advisories/CA-2002-12.html Risk factor : High CVE : CAN-2003-0026, CAN-2002-0702, CAN-2003-0039 BID : 4701, 6627, 6628, 11591 Other references : RHSA:RHSA-2003:034-01 Nessus ID : 11312

[back to the list of ports]

Warning found on port general/icmp

The remote host answers to an ICMP timestamp request. This allows an attacker to know the date which is set on your machine. This may help him to defeat all your time based authentication protocols. Solution : filter out the ICMP timestamp requests (13), and the outgoing ICMP timestamp replies (14). Risk factor : Low CVE : CAN-1999-0524 Nessus ID : 10114

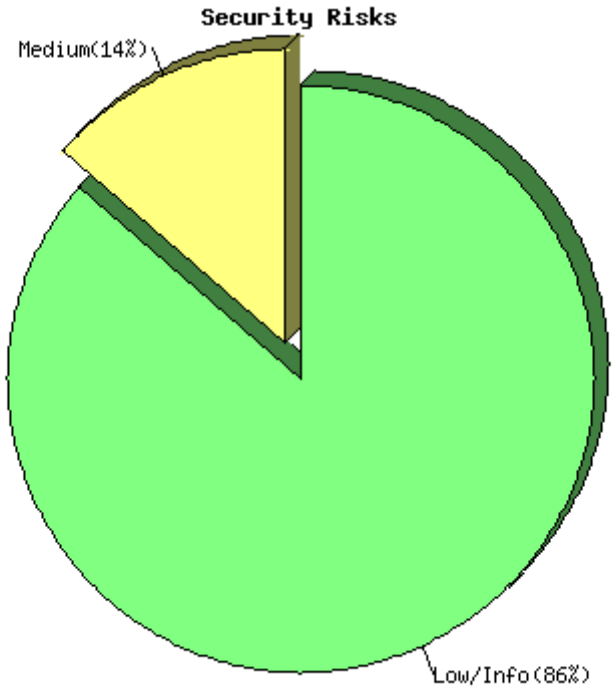
[Next host : 192.168.1.4]

This file was generated by Nessus, the open-sourced security scanner.

ANEXO K – Relatório do *nessus* para o servidor no endereço IP 192.168.1.4 no dia 10 de dezembro de 2005.

192.168.1.4

Repartition of the level of the security problems :



[Back to the index]

List of open ports :

- *ssh (22/tcp)* *(Security notes found)*
- *smtp (25/tcp)* *(Security notes found)*
- *pop3 (110/tcp)* *(Security notes found)*
- *netbios-ssn (139/tcp)* *(Security notes found)*
- *imap (143/tcp)* *(Security notes found)*
- *microsoft-ds (445/tcp)* *(Security warnings found)*
- *submission (587/tcp)* *(Security notes found)*
- *nessus (1241/tcp)* *(Security warnings found)*
- *x11 (6000/tcp)* *(Security notes found)*

- *biff (512/udp)*
- *netbios-ns (137/udp) (Security warnings found)*
- *netbios-dgm (138/udp)*
- *bootpc (68/udp)*
- *unknown (34269/udp)*
- *general/tcp (Security notes found)*

[back to the list of ports]

Information found on port ssh (22/tcp)

An ssh server is running on this port
Nessus ID : 10330

[back to the list of ports]

Information found on port ssh (22/tcp)

Remote SSH version : SSH-2.0-OpenSSH_4.2

Nessus ID : 10267

[back to the list of ports]

Information found on port ssh (22/tcp)

The remote SSH daemon supports the following versions of the SSH protocol :

. 1.99
. 2.0

SSHv2 host key fingerprint : 34:bd:64:6d:fe:a4:50:4b:1e:03:90:76:a7:07:51:a7

Nessus ID : 10881

[back to the list of ports]

Information found on port smtp (25/tcp)

An SMTP server is running on this port
Here is its banner :
220 juarezdc.juarezdc.org ESMTP Sendmail 8.13.4/8.13.4; Sun, 11 Dec 2005 15:08:13 -0200
Nessus ID : 10330

[back to the list of ports]

Information found on port smtp (25/tcp)

Remote SMTP server banner :
220 juarezdc.juarezdc.org ESMTP Sendmail 8.13.4/8.13.4; Sun, 11 Dec 2005 15:08:50 -0200

This is probably: Sendmail version 8.13.4

Nessus ID : 10263

[back to the list of ports]

Information found on port pop3 (110/tcp)

A pop3 server is running on this port
Nessus ID : 10330

[back to the list of ports]

Information found on port netbios-ssn (139/tcp)

An SMB server is running on this port
Nessus ID : 11011

[back to the list of ports]

Information found on port imap (143/tcp)

An IMAP server is running on this port
Nessus ID : 10330

[back to the list of ports]

Information found on port imap (143/tcp)

The remote imap server banner is :
* OK [CAPABILITY IMAP4REV1 LITERAL+ SASL-IR LOGIN-REFERRALS STARTTLS AUTH=LOGIN] juarezdc.juarezdc.org IMAP4rev1 2004.357 at Sun, 11 Dec 2005 15:08:13 -0200 (BRST)
Versions and types should be omitted where possible.
Change the imap banner to something generic.
Nessus ID : 11414

[back to the list of ports]

Warning found on port microsoft-ds (445/tcp)

Here is the browse list of the remote host :

JUAREZDC (os: 0.0)

This is potentially dangerous as this may help the attack
of a potential hacker by giving him extra targets to check for

Solution : filter incoming traffic to this port
Risk factor : Low

Nessus ID : 10397

[back to the list of ports]

Information found on port microsoft-ds (445/tcp)

A CIFS server is running on this port
Nessus ID : 11011

[back to the list of ports]

Information found on port microsoft-ds (445/tcp)

- NULL sessions are enabled on the remote host

CVE : CAN-1999-0504, CAN-1999-0506, CVE-2000-0222, CAN-1999-0505, CAN-2002-1117
BID : 494, 990, 11199
Nessus ID : 10394

[back to the list of ports]

Information found on port microsoft-ds (445/tcp)

The remote native lan manager is : Samba 3.0.20
The remote Operating System is : Unix
The remote SMB Domain Name is : MYGROUP

Nessus ID : 10785

[back to the list of ports]

Information found on port submission (587/tcp)

An SMTP server is running on this port
Here is its banner :
220 juarezdc.juarezdc.org ESMTP Sendmail 8.13.4/8.13.4; Sun, 11 Dec 2005 15:08:13 -0200
Nessus ID : 10330

[back to the list of ports]

Information found on port submission (587/tcp)

Remote SMTP server banner :
220 juarezdc.juarezdc.org ESMTP Sendmail 8.13.4/8.13.4; Sun, 11 Dec 2005 15:08:50 -0200

This is probably: Sendmail version 8.13.4

Nessus ID : 10263

[back to the list of ports]

Warning found on port nessus (1241/tcp)

A Nessus Daemon is listening on this port.
Nessus ID : 10147

[back to the list of ports]

Information found on port nessus (1241/tcp)

A TLSv1 server answered on this port

Nessus ID : 10330

[back to the list of ports]

Information found on port nessus (1241/tcp)

Here is the TLSv1 server certificate:
Certificate:
Data:
Version: 3 (0x2)
Serial Number: 1 (0x1)
Signature Algorithm: md5WithRSAEncryption
Issuer: C=BR, ST=RS, L=POA, O=Nessus Users United, OU=Certification Authority for caverinha, CN=caverinha/emailAddress=ca@caverinha
Validity
Not Before: Nov 15 19:32:01 2005 GMT
Not After : Nov 15 19:32:01 2006 GMT
Subject: C=BR, ST=RS, L=POA, O=Nessus Users United, OU=Server certificate for caverinha, CN=caverinha/emailAddress=nessusd@caverinha
Subject Public Key Info:
Public Key Algorithm: rsaEncryption
RSA Public Key: (1024 bit)
Modulus (1024 bit):
00:d8:53:35:8d:df:20:71:6a:8c:a1:46:93:0c:7b:
27:37:7c:1c:d0:4b:37:f0:a6:b6:d0:97:4a:d5:9e:
ac:01:9e:6d:55:1d:06:c7:28:97:99:1c:69:e1:d9:
42:65:1e:17:63:8f:11:99:2a:29:59:e1:88:94:d3:
b9:c3:17:80:69:97:34:5c:8e:57:df:95:dd:24:5c:
13:ed:f9:93:a1:f6:a2:be:23:28:64:b3:40:75:43:
4a:19:df:40:ed:a1:76:09:49:7b:fc:0b:7c:ed:77:
b9:9d:05:37:19:50:fe:e8:d8:08:7c:8a:70:ad:71:
e0:95:81:5f:95:22:60:5d:6f
Exponent: 65537 (0x10001)
X509v3 extensions:
Netscape Cert Type:
SSL Server
X509v3 Key Usage:
Digital Signature, Non Repudiation, Key Encipherment
Netscape Comment:
OpenSSL Generated Certificate
X509v3 Subject Key Identifier:
86:4D:47:83:E2:8F:D0:76:5A:66:79:82:55:3C:5C:4D:13:35:46:80
X509v3 Authority Key Identifier:
keyid:91:A0:70:28:EE:1A:A1:1A:9A:03:57:F1:71:3C:D5:7E:46:86:45:CD
DirName:/C=BR/ST=RS/L=POA/O=Nessus Users United/OU=Certification Authority for caverinha/CN=caverinha/emailAddress=ca@caverinha
serial:A7:E8:5F:D7:90:E6:E2:7C

X509v3 Subject Alternative Name:
email:nessusd@caverinha
X509v3 Issuer Alternative Name:
<EMPTY>

Signature Algorithm: md5WithRSAEncryption
42:59:a5:4f:fa:20:68:21:16:10:30:94:87:7d:19:4d:5a:6b:
80:35:9f:ca:1a:92:b8:f4:ed:c2:ee:33:35:98:09:ec:88:4b:
92:59:3c:28:83:d6:9a:9f:25:50:b3:3b:86:27:68:29:43:42:
24:82:3e:0a:05:12:93:e1:90:44:c9:6f:a4:8a:ef:7d:a4:57:
29:ee:f4:96:b7:1d:23:95:00:c0:49:e5:a5:7e:9f:9b:0e:65:
2c:60:cd:29:8f:98:d6:a0:5f:6b:ae:a7:c0:3d:e4:e9:6e:2c:

6e:ba:6a:be:77:d4:39:d4:3a:7e:61:ad:22:75:8d:f7:17:b6:
33:47
This TLSv1 server does not accept SSLv2 connections.
This TLSv1 server does not accept SSLv3 connections.

Nessus ID : 10863

[back to the list of ports]

Information found on port x11 (6000/tcp)

This X server does *not* allow any client to connect to it
however it is recommended that you filter incoming connections
to this port as attacker may send garbage data and slow down
your X session or even kill the server.

Here is the server version : 11.0
Here is the message we received : No protocol specified

Solution : filter incoming connections to ports 6000-6009
Risk factor : Low
Nessus ID : 10407

[back to the list of ports]

Warning found on port netbios-ns (137/udp)

The following 7 NetBIOS names have been gathered :
JUAREZDC = This is the computer name registered for workstation services by
a WINS client.
JUAREZDC = This is the current logged in user registered for this workstation.
JUAREZDC = Computer name
__MSBROWSE__
MYGROUP = Workgroup / Domain name
MYGROUP
MYGROUP = Workgroup / Domain name (part of the Browser elections)

. This SMB server seems to be a SAMBA server (this is not a security
risk, this is for your information). This can be told because this server
claims to have a null MAC address

If you do not want to allow everyone to find the NetBios name
of your computer, you should filter incoming traffic to this port.

Risk factor : Medium
CVE : CAN-1999-0621
Nessus ID : 10150

[back to the list of ports]

Information found on port general/tcp

192.168.1.4 resolves as juarezdc.juarezdc.org.
Nessus ID : 12053

[back to the list of ports]

Information found on port general/tcp

The output of "uname -a" is :
Linux juarezdc 2.6.13.4 #5 SMP Wed Oct 26 22:22:36 BRST 2005 i686 unknown unknown GNU/Linux

The remote Linux distribution is not supported, therefore local security checks have not been enabled
Nessus ID : 12634

This file was generated by Nessus, the open-sourced security scanner.