Antonio Cesar Pazebão

Segurança de Perímetro - Projeto Colegio ADV - Jaú

Monografia de Pós-Graduação "*Lato Sensu*" apresentada ao Departamento de Ciência da Computação para obtenção do título de Especialista em "Administração em Redes Linux"

Orientador
Prof. Msc. Denilson Vedoveto Martins

Lavras
Minas Gerais - Brasil
2010

Antonio Cesar Pazebão

Segurança de Perímetro - Projeto Colegio ADV - Jaú

Monografia de Pós-Graduação "*Lato Sensu*" apresentada ao Departamento de Ciência da Computação para obtenção do título de Especialista em "Administração em Redes Linux"

Aprovada em *Abril 2010*

Prof. Dr. Joaquim Quinteiro Uchôa

Prof. Msc. Herlon Ayres Camargo

Prof. Msc. Denilson Vedoveto Martins
(Orientador)

Lavras
Minas Gerais - Brasil
2010

Aos meus pais Florindo e Aparecida, às minhas irmãs Irandi, Regina, Sueli e principalmente Maria de Lourdes, por apoiarem-me nessa empreitada. Ao também amigo de muitos anos, Prof. Mestre em Geografia Armstrong Machado, pela força para que eu continuasse com os meus estudos na área.

Agradecimentos

Agradeço primeiramente a Deus, em especial aos novos amigos da turma ARL 108 e corpo docente da Universidade Federal de Lavras - UFLA pelo apoio e companheirismo nessa jornada. A todo o Colégio ADV - Unidade II - Jaú SP., em especial Patricia Rebeca Nigro Rivera, pela enorme colaboração para o desenvolvimento desse trabalho. A todos o meu profundo agradecimento.

Sumário

Lista de Figuras

Lista de Tabelas

Resumo

O presente projeto tem por objetivo o controle do perímetro compreendido entre a rede de comunicação de dados da unidade escolar e a Internet. O mesmo fora implementado em ambiente GNU/Linux com aplicações livres, anteriormente exercidas por servidor único, utilizando-se de ferramentas proprietárias. A separação dos sistemas de segurança da rede de comunicação de dados visa ofertar melhora significativa no sistema de segurança, bem como, prover controle de acesso para o corpo administrativo, docente e discente da unidade escolar.

Como pontos relevantes no projeto proposto além da substituição de ferramenta de segurança proprietária por ferramentas livres, implantou-se também a rede *Wireless* para uso do corpo docente e administrativo. Para tanto, otimizou-se além de outras aplicações, um *firewall* bastante restritivo e o uso do *squid* autenticado onde permite-se o monitoramento dos acessos a Internet de todo o corpo administrativo, docente e discente da unidade escolar, com intuito de análisar possíveis incidentes que por ventura possam ocorrer.

Palavras-Chave: Perímetro, segurança, servidor, *iptables*, *squid* e rede *wireless.*

Capítulo 1

Introdução

Notoriamente nos dias atuais, se torna necessário a proteção do perímetro de uma rede local afim de garantir a disponibilidade, integridade e confidencialidade dos dados gerados, tentando-se evitar ao extremo atos não condizentes com o modelo de negócio de uma instituição, seja por agentes externos ou internos. Trabalhos e estudos hoje se atentam para fontes de proteção do perímetro de uma rede local como um todo, incluindo-se sistema de detecção de intrusão, *firewall, proxy* autenticado/transparente, sistema de *caching* e um sistema de *hardening* aplicado ao *hardware* usado como fronteira entre a rede local e a Internet.

O projeto em questão, propõe a substituição e comparação dos sistemas de segurança do perímetro da rede local do Colegio ADV - Unidade II Jaú SP, existentes hoje, por aplicações livres. Tal mudança é motivada por diversos fatores destacando-se entre eles significativa melhora na segurança de acesso aos dados e às informações, uma vez que todos os serviços da rede local, segurança de perímetro e acesso a Internet estão centralizados em um único servidor.

Por essa nova implementação se espera obter o uso da rede local e Internet de forma segura, separando-se os serviços de rede local e segurança do perímetro, controlando o acesso individual envolvendo toda área administrativa, corpo docente e discente da unidade de ensino, bem como a flexibilização da rede local para a implantação de rede *wireless.*

Por essas necessidades encontradas na unidade de ensino o autor, de forma perceptiva otimizou a mudança na rede local, visando o aprimoramento de seus conhecimentos em *software* livre, podendo o mesmo estudo servir como base e experiência para novas implementações em unidades de ensino público ou privado

e também em empresas que operam em negócios comerciais. Como objeto de pré-otimização, o autor elaborou a simulação da implantação em ambiente virtualizado, afim de se criar uma situação mais próxima possível da realidade da unidade de ensino. Esta monografia fora elaborada com base em consultas à bibliografia especializada, bem como artigos e textos mantidos por organizações responsáveis pelas aplicações utilizadas.

O estudo de caso encontra-se organizado como segue. O Capítulo 2 apresenta conceitos de segurança de perimetro, análise do perímetro da rede da unidade escolar existente e sua topologia. Traz ainda inventário dos ativos de rede, sistemas operacionais utilizados, aplicações, análise de tráfego e vulnerabilidades. O Capítulo 3 mostra o projeto proposto para a rede local e implantação da rede *wireless*, equipamentos utilizados, a distribuição GNU/Linux, os serviços, as aplicações utilizadas e a configuração utilizada no roteador *wireless.* No Capítulo 4, demonstra-se a implantação do projeto onde tem-se o *hardening* proposto para o servidor em questão, configuração de interfaces de rede, aplicações e serviços bem como, análise de possíveis vulnerabilidades, e o resultado do monitoramento do trafego da rede após o servidor ser colocado em produção. No Capítulo 5 apresenta-se a conclusão do projeto em questão com análises e observações do autor acerca da implementação do projeto.

Capítulo 2

Segurança de perímetro

2.1 Conceito de segurança de perímetro

Para (CHESWICK; BELLOVIN; RUBIN, 2005) entende-se por segurança de perímetro a dificuldade em manter seguro cada *host* em uma rede. Conceitua-se como sendo de suma importância a segurança de perímetro na Internet. Trata-se como componentes básicos para devida implementação, um *firewall,* onde se permite a realização dos trabalhos com uma boa dosagem de segurança. Controla-se o fluxo de entrada e saída de dados, como também asseguram-se todos os dispositivos de rede, sanando-se possíveis falhas em softwares e portas dos fundos aberta.

> *"Se for muito difícil manter segura cada casa em uma vizinhança, talvez os residentes possam associar-se para construir um muro em torno da cidade. Assim, as pessoas precisam temer somente a elas próprias e a uma força invasora suficientemente forte que rompa o muro. Guardas alertas e bem treinados podem ser posicionados nos portões enquanto as pessoas ocupam-se de seus negócios."(CHESWICK; BELLOVIN; RUBIN, 2005)*

De acordo com (NORTHCUTT *et al.*, 2005), toda rede é avaliada diariamente, pois sempre há pessoas envolvidas na avaliação da eficácia de suas defesas. No entanto, sistemas de computadores com endereços IP, podem ser alcançados entre milhares de tentativas de ataque, como muitos sendo simples *scans* ou ameaças de maior sofisticação acarretando-se grandes incidentes.

Diante do exposto, o que é exatamente o perímetro? Perímetro é conceituado como sendo uma fronteira fortificada de rede onde se pode incluir roteadores de fronteira que direcionam o fluxo de dados para dentro e para fora dessa rede, sendo o último roteador controlador de entrada e saída de dados antes de uma rede insegura como a Internet.

Trabalhando junto com roteadores de fronteira tem-se os *firewalls,* dispositivos embasados em regras especificando o aceite ou não de pacotes de dados para dentro ou fora da rede. Geralmente, aloca-se o mesmo em ponto estratégico de forma a interagir com roteadores de fronteira, desenvolvendo filtragem bastante rigorosa dos pacotes de dados.

Conceitua-se também como defesa de perímetro, a prevenção de possíveis ataques por sistemas de detecção de intrusão. Tais sistemas podem ser definidos como alarmes anti-roubo para uma rede, desempenhando as funções de detecção e alertas a eventos maliciosos, com sensores IDS colocados em pontos estratégicos na rede. Pode-se usar com base em rede (NDIS), tendo-se a exemplo o *Snort*[1] ou Cisco *Secure* IDS[2], para o monitoramento de atividades suspeitas.

Pode-se também fazer o uso de IDS baseados no anfitrião[3], como *Tripwire*[4] ou *ISS BlackICE*[5], os quais monitoram-se anfitriões, desempenham análises estatísticas e de anomalias. Havendo-se a detecção de eventos suspeitos pode-se alertar de várias maneiras ou apenas registrando-se a ocorrência. Pode-se também reportar a uma base de dados central a qual correlaciona suas informações para verificar a rede de muitos pontos. Tem-se também os sistemas de prevenção de intrusões os quais automaticamente detectam e impedem-se ataques de computador contra recursos protegidos.

Esforça-se para automaticamente defender o alvo sem o envolvimento direto do administrador. Nessa proteção, pode-se envolver o uso de técnicas baseadas em assinaturas ou de comportamento a fim de identificar ataques para bloqueio de fluxo malicioso, ou então uma chamada de sistema, a fim de evitar-se danos. Pode-

[1] Disponível em http://www.snort.org/

[2] Disponível em http://www.cisco.com/warp/public/cc/pd/sqsw/sqidsz/index.shtml

[3] *Host-Based* IDS - analisa diversas áreas afim de determinar desvios ou intrusão. Consultam vários tipos de arquivos de *log* (*kernel*, sistema, servidor de rede, *firewall* e etc) e compara os logs internos contra uma base de dados de assinaturas comuns.

[4] Ferramenta de verificação de assinatura de arquivos e diretórios de todo um sistema procurando alterações não autorizadas.

[5] Executa verficação de assinatura de arquivos e diretórios de todo um sistema, desenvolvido pela IBM

se combinar funcionalidade de um *firewall* e IDS a fim de ofertar uma solução automática para bloqueio de ações ofensoras quando se detecta um ataque.

Conceituando-se a proteção de perímetro de redes, devem-se referenciar também aplicações sediadas na rede das organizações, definindo-se como estruturadas, pois desempenha papel importante e significante na infra-estrutura de segurança, onde tem-se como propósito primário da rede, proteger-se os dados e serviços do aplicativo.

2.2 Análise do perímetro da rede

O estudo de caso fora realizado para o Colégio ADV – Unidade II, localizado na cidade de Jaú – SP. A rede encontra-se com todo o tráfego de pacotes passando diretamente por um único servidor, o qual disponibiliza os serviços de banco de dados, servidor de arquivos, funções de *gateway* para a Internet, *firewall e proxy.*

2.2.1 *Layout* da rede atual

Analisando-se a topologia da rede existente, pode-se notar a entrada de dados provenientes do modem ADSL[6], roteado e conectado diretamente ao servidor de arquivos, o qual também é *proxy/firewall.* O servidor, bem como todas as estações de trabalho, são conectadas diretamente a um *switch* central. A análise pode ser melhor elucidada conforme a figura 2.1.

Tem-se então, dezenove computadores distribuídos em topologia estrela, sendo doze equipamentos para uso do corpo discente, dois equipamentos para os serviços de biblioteca, dois equipamentos para os serviços de secretaria, um equipamento para a coordenação, um equipamento para armazenamento de imagens de circuito fechado de TV e o servidor principal.

2.2.2 Inventário dos ativos da rede.

Inventariando-se os ativos da rede local existente hoje conforme a figura 2.1, tem-se as seguintes configurações de *hardware:*

[6] *Assymetric Digital Subscriber Line* - Tecnologia para transferência digital de dados em alta velocidade por meio de linhas telefônicas comuns.

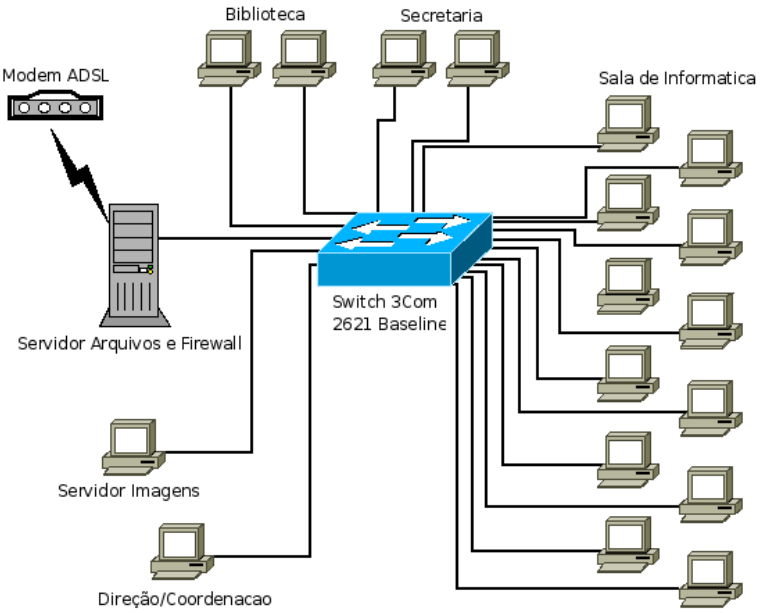


Figura 2.1: *Layout* da rede atual

- Servidor de arquivos, banco de dados e *firewall/proxy:* Equipamento com processador Intel *Core 2 Duo* E 4500, com HD 160 GB, 1 GB memória RAM, sistema RAID 1.
- Servidor de imagens para circuito fechado de TV: Equipamento com processador Celeron 2.53 Ghz, 1.0 Gb memória RAM e HD 80 Gb.
- Biblioteca: Equipamento com processador Celeron 2.53 Ghz, 256 Mb memória RAM e HD 40 Gb. Equipamento com processador Pentium Dual E2180 – 2.0 Ghz, HD 80 Gb e memória RAM 1.0 Gb.
- Coordenação: Equipamento com processador Celeron 2.53 Ghz, 256 Mb memória RAM e HD 40 Gb.
- Secretaria: Equipamento com processador Pentium Dual E2180 – 2.0 Ghz, HD 80 Gb e memória RAM 1.0 Gb.
- Laboratório de Informática: Equipamento com processador Celeron 2.53 Ghz, 256 Mb memória RAM e HD 40 Gb.
- Modem ADSL: Modem D-Link 500B, com entrada WAN e saída Ethernet 10/100 Mb com velocidade de banda contratada de 500 Kbps.

- Switch: 3Com, *baseline* 2126 – 24 portas 10/100 e 2 portas 10/100/1000, não gerenciável e configuração fixa de camada [7]

2.2.3 Sistema Operacional e aplicações no servidor

O sistema operacional instalado atualmente é o *Microsoft Windows 2000 Server,* com os seguintes serviços e aplicações:

- *Active directory:* centralização de todos os usuarios, computadores, dispositivos de impressão em um único domínio.
- DHCP - *domain host control protocol:* escopo criado para distribuir dinamicamente endereços IP para toda rede local, resguardando endereços fixos para serem usados em determinados equipamentos.
- DNS - *Domain name system:* como o próprio nome já diz, sistemas de nome de domínios dá-se pela conversão de endereços IP em nomes. O *Windows 2000 Server* possui nativamente um servidor Dns, onde é de fundamental importância também no *active directory.*
- *Microsoft Access:* sistema de banco de dados relacional (SGBDR) utilizado como aplicação para banco de dados do sistema de gerenciamento escolar utilizado pela unidade de ensino.
- Bedel 5.0 - Sistema de gerenciamento escolar: desenvolvido pela *SpeedSoft*[8] Soluções Educacionais, utilizado para gerenciamento do controle do corpo discente. Utiliza-se alem da aplicação, o *Microsoft Access* como sistema de banco de dados.
- *Microsoft Isa-server 2000* - Como forma de proteger o perímetro da rede local do Colégio, objeto de estudo de caso, utiliza-se hoje o *Microsoft ISA Server*[9] 2000. Tal aplicação desenvolvida pela *Microsoft,* tem-se como sendo sua última versão a 2006. Destina-se a controle de tráfego de entrada e saída de rede, bem como serviço de controle de banda, acesso a Internet, *cache* e *firewall.* Conforme abordado por (BADDINI, 2003), o *ISA Server 2000* possui uma série de recursos voltados para segurança Internet. O mesmo

[7] Modelo de referência OSI - camada de enlace, controle do link logico.

[8] Disponível em *www.speedsoft.com.br*

[9] *MS Isa Server* - Aplicação para controle de acesso, fluxo de dados *firewall* e IDS - desenvolvido pela *Microsoft*.

disponibiliza um *firewall* multi-camadas, onde controla o tráfego com base em regras aplicadas ao subsistema de rede contemplando várias camadas existentes no modelo OSI[10]. Tais regras podem ser criadas com intuito de negar ou permitir o fluxo de dados partindo de uma determinada porta e um protocolo definido ou ainda, partir de filtros de aplicativos.

O mesmo pode fazer ainda o *SecureNat,* onde implementa a publicação de recursos *online* através de requisições vindas da *Web,* redirecionado para servidores internos. Possui ainda, IDS *(Intrusion Detection System)* onde é analisado todo o tráfego que é requisitado ao *firewall* detectar que uma intrusão está ocorrendo ou tentando ocorrer. Em relação a *web cache,* o mesmo armazena em disco rígido objetos HTTP, HTTPS e FTP, onde os quais podem ser oriundos da *Web* ou do servidor *Web* interno de uma instituição. O mesmo trata ainda de *download* programado de conteúdo, onde pode-se agendar *download* de *Web Sites* e armazenamento em *cache,* afim de oferecer uma rápida entrega do conteúdo a usuários da rede.

De acordo com (SHINDER; SHINDER; GRASDAL, 2001) o *MS ISA Server* pode ser usado para configurar uma VPN ou Acesso remoto entre um cliente e um *gateway* ou um membro de vários túneis VPN de servidor para servidor. Políticas de acesso e informações de configuração do *ISA Server,* podem ser integradas com o *active directory* para uma administração mais fácil e segura. Afim de otimizar o tráfego de rede, pode-se ainda fazer controle de largura de banda a ser atribuído a um determinado usuário, comunicação ou cliente de destino através da qualidade de serviço (QoS).

Pode-se tambem disponibilizar acesso para servidores de publicação interna e a clientes específicos protegendo-os contra acessos não autorizados. O mesmo prevê ainda, análise de conteúdo de e-mail por palavra-chave afim de permitir aplicação e implementação de rigorosas políticas de segurança, a funcionalidade do H.323 *Gatekeeper*[11] para utilização de *software* para vídeo-conferência através de *proxy.*

O mesmo executa a geração e acompanhamento de relatórios permitindo monitoramento de desempenho e segurança detalhada, além de criar arquivos de *log's* de acesso e gerar relatórios gráficos, onde os mesmos podem ser programados. Enfim, pode-se dizer que o mesmo fora projetado para uma maior escalabilidade com foco no mercado corporativo, permitindo de-

[10] OSI - Open System Interconect - Modelo de referencia para conexão de sistemas

[11] H.323 *Gatekeeper* - A recomendação H.323 tem o objetivo de especificar sistemas de comunicação multimídia em redes baseadas em pacotes e que não provêem uma Qualidade de Serviço (QoS) garantida.

finir políticas de uso e acesso a nível empresarial, de uma forma um tanto centralizada.

2.2.4 Monitoramento da rede

A aplicação escolhida para o monitoramento da rede local fora o MRTG[12]. De fácil operação e otimização, a motivação pelo seu uso se deu por ser uma aplicação livre e também suportada em sistemas operacionais proprietários. Com larga utilização para monitoramento de tráfego em interfaces de redes, seu desenvolvimento está embasado na aplicação *Perl*[13].

O prazo total para coleta de dados e geração de gráficos referente ao monitoramento, foi de aproximadamente quarenta dias, com inicio em 01 de abril de 2009 e encerramento em 10 de maio de 2009. Durante o monitoramento, foram analisadas as duas interfaces de rede existentes no servidor, sendo a interface com IP 10.1.1.2 conectada diretamente com o servidor e o modem ADSL. A interface cujo IP 192.168.0.1 conecta o servidor com o *switch* da rede local. O sistema de conexão do servidor com o modem ADSL e a rede local pode ser melhor visualizado no *layout*, na figura 2.2.

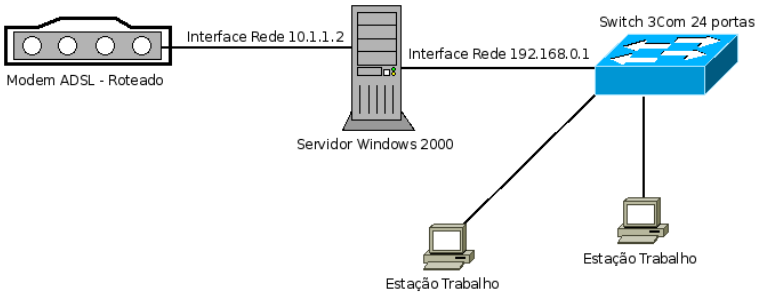


Figura 2.2: *Layout* da conexão com servidor

[12] *Multi Router Traffic Grapher* - desenvolvido por *Tobi Oetiker's* usado para análise de rede. Disponível em *http://oss.oetiker.ch/mrtg/*

[13] *Practical Extraction And Report Language* - linguagem de programação estável e multiplataforma, usada em aplicações de missão crítica em todos os setores com destaque para desenvolvimento de aplicações *Web* de todos os tipos.

2.2.4.1 Análise do tráfego gerado nas interfaces de rede

Com a análise do tráfego gerado nas interfaces de rede, obteve-se os seguintes resultados:

De acordo com a figura 2.3, observa-se que o fluxo máximo de entrada de pacotes fora de 53,3 kB/s, obtendo-se 42,7% (pontos percentuais). O fluxo máximo de saída de pacotes fora de 2599.0 B/s, equivalente a 2,1% (pontos percentuais) do total ofertado. O fluxo médio de entrada de pacotes em *bytes* por segundo, fora de 1984.0 B/s com média de 1,6% (pontos percentuais). O fluxo médio de saída de pacotes fora de 239.0 B/s, obtendo-se em média 0,2% (pontos percentuais) do total de pacotes entrantes. Os percentuais ora apurados mostram que em relação ao uso da Internet obteve-se maior entrada de pacotes de dados em relação a pacotes saintes.

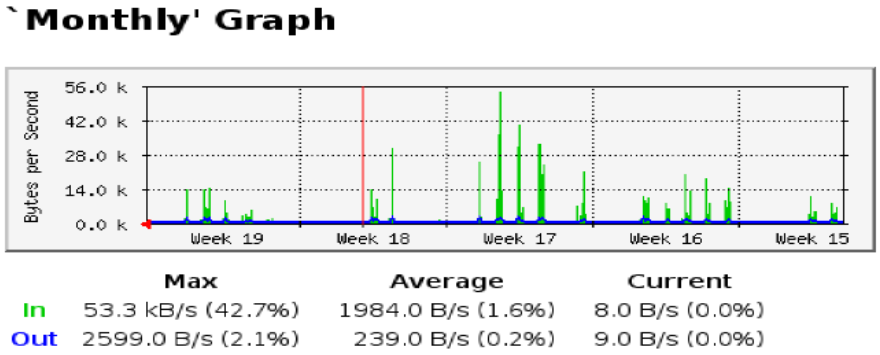


Figura 2.3: Monitoramento interface rede acesso Internet IP 10.1.1.2

De acordo com a figura 2.4, nota-se uma saída maior de pacotes de dados, devido o acesso aos dados contidos no servidor por usuários da rede local. Em média, a cada 514,0 B/s (0,4%) de entrada de dados, tem-se 3709,0 B/s (3,0%) de saída de pacotes de dados. Analisando-se o fluxo máximo da rede local, tem-se 7537,0 B/s (6,0%) de entrada de dados e 58,9 kB/s (47,2%) de saída de dados.

Por essa análise, através dos gráficos gerados pelo MRTG, o fluxo de acesso ao servidor bem como a troca de pacotes é consideravelmente maior que o acesso à Internet, haja visto o fluxo médio de acesso aos dados produzido pela rede local na ordem de 3.0% contra o fluxo de acesso à Internet, na ordem de 0.2%.

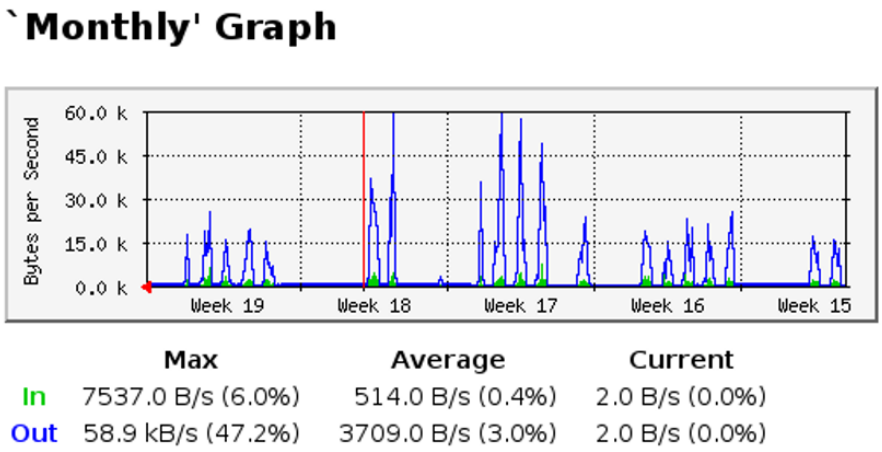


Figura 2.4: Monitoramento interface rede local IP 192.168.0.1

2.2.5 Análise de vulnerabilidades no servidor

Para auditoria e análise de possíveis vulnerabilidades, a aplicação escolhida fora o Nessus[14]. A mesma realiza uma varredura de portas, detecta servidores ativos e simula invasões a fim de detectar vulnerabilidades. Tem como importância, a procura não apenas em portas padrão, mas em todas as portas TCP, sendo capaz de detectar qualquer vulnerabilidade em qualquer aplicação e em qualquer porta que o mesmo estiver ativo.

A ferramenta *Nessus* é lider mundial em *scanners* ativos, sendo referência para *software open source* de análise de vulnerabilidade. Apresenta alta velocidade de descoberta, configuração de auditoria, perfis, descoberta de dados sensíveis e análise do perfil de vulnerabilidade de segurança. Durante a auditoria executada no servidor, além das vulnerabilidades, constatou-se um enorme volume de portas abertas, o que caracteriza uma configuração não adequada para o servidor em questão. Através do uso da aplicação Nessus, obteve-se as seguintes vulnerabilidades:

[14]Ferramenta para uso em análise de vulnerabilidades em redes. Disponível em http://www.nessus.org/nessus/

- Porta *Epmap*(135/tcp): vulnerabilidade com fator de risco alto, tem como *ID Nessus 11808.* Codificada pelo BID[15] 8205, CVE[16] CAN-2003-0352. Entende-se por uma ameaça devido a uma falha na interface *Dcom Rpc,* o qual permite a um atacante executar um código arbitrário e obter privilégios no sistema. Para essa vulnerabilidade, existe ao menos um *Worm*[17], que poderá explora-la, denominado de *MsBlaster4.* Escuta-se essa vulnerabilidade na porta 135 TCP/UDP, onde o mesmo pode ser estendido a outras portas tais como TCP 139, 445 e 593. Encontram-se vários *exploits* que podem ser utilizados por essa vulnerabilidade.

- Porta *Snmp*(161/udp) e desconhecida(32789/udp): conforme se explanou através do relatório do *Nessus ID 10264,* para essa vulnerabilidade o mesmo não mostrou o fator de risco. Codificada pelo BID 11237, 10576, 177, 2112, 6825, 7081, 7212, 7317, 9681, 986. CVE CAN-1999-0517, CAN-1999-0186, CAN-1999-0254, CAN-1999-0516. Confirma-se nessa vulnerabilidade que o agente *Snmp*[18] respondeu como esperado com nome de comunidade pública. Pode-se, por meio dessa vulnerabilidade, alavancar um problema de negação de serviço a fim de que o equipamento afetado pare de responder exigindo-se um desligamento para trazer de volta a sua funcionalidade. Pode-se alterar também o conjunto de dispositivos afetados por padrão, facilitando a alteração das configurações dos dispositivos. Para resolver o problema dessa vulnerabilidade, o fabricante disponibilizou uma atualização do firmware a fim de lidar com essas questões.

[15] *Bugtraq ID* - Identificação de defeito em aplicações.

[16] CVE - *Common Vulnerabilities and Exposures* - Exposições e vulnerabilidades comuns, o qual utiliza o CAN, registro de vulnerabilidade do cve.mitre.org.

[17] *Worm* - verme em português. Sistema auto-replicante, semelhante a vírus de computador.

[18] SNMP - *Simple Network Management Protocol* - protocolo simples de gerência de redes. Protocolo típico para gerência de redes TCP/IP.

Capítulo 3

Projeto proposto, equipamentos e aplicações utilizadas

3.1 *Layout* proposto para o perímetro

A proposta do novo projeto visa a separação dos serviços de segurança de perímetro entre a rede local e a Internet ora desenvolvido pelo servidor de arquivos e banco de dados da Unidade Escolar. Para a elaboração do projeto proposto, houve a necessidade de aquisição de novos equipamentos de *hardware*, onde o mesmo, após sua otimização, será conectado diretamente ao modem ADSL, servindo como *gateway* para a Internet envolvendo a rede local e a rede *wireless*. O novo projeto do perímetro da rede local é melhor explicitado na figura 3.1.

3.2 Equipamentos utilizados

Conforme o *layout* do projeto proposto na figura 3.1, incluiu-se dois equipamentos na rede local sendo um computador otimizado para executar os serviços de segurança de perímetro da rede local e da rede wireless e, um roteador *wireless* para disponibilizar Internet sem fio para uso nas salas de aula.

Após a reestruturação, haverá um aumento nos ativos de rede, acrescentado-se um micro-computador e um roteador *wireless.* O novo *layout* da rede local pode ser melhor compreendido na figura 3.2. Os equipamentos destinados à otimização dos serviços foram:

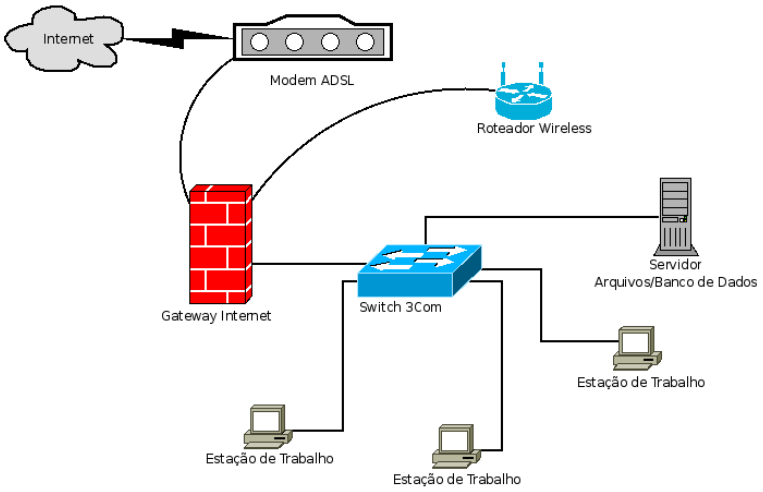


Figura 3.1: *Layout* proposto para o perímetro

- Equipamento com processador Celeron(R) 2.53 GHz, memória DDR 1024 GB - PC 400 MHz, disco rígido SATA 160.0 GB, *mainboard Gigabyte* GA-8I865GME-775 com três placas de rede 3Com - *Chipset* 3Com 920-ST03 10/100 Mb.

- Roteador *wireless* marca TP-Link 2.4 GHz - 802.11 b/g, modelo No. TL-WR642G, 108M *Wireless G Router.*

3.3 Distribuição GNU/Linux utilizada

Para esse trabalho, o autor utilizou-se da distribuição Debian[1]. Seu uso se deu por motivo da mesma ser de uso pessoal aproximadamente há dois anos, pela simplicidade na instalação de pacotes, por ser reconhecida como uma das mais seguras, ser atualizada frequentemente e principalmente por ser uma das únicas sem fins comerciais além da mesma ter suporte para mais de 12 arquiteturas e 15 sub-arquiteturas. Suas versões são lançadas após rigorosos testes de segurança e correção de falhas tornando-a como uma das distribuições mais seguras.

[1] http://www.debian.org/

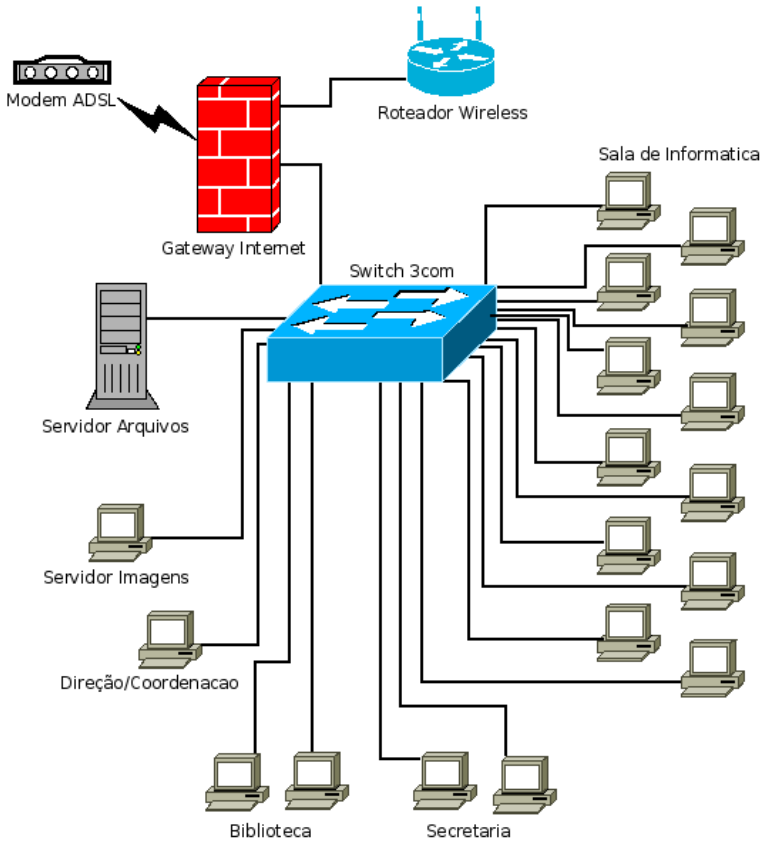


Figura 3.2: *Layout* da rede local com novo perímetro

3.4 Serviços e aplicações utilizadas no servidor

Para otimização do servidor em questão, foram disponibilizados diversos serviços entre eles Dns-cache com intuito de resolver problemas relacionados a negação de serviços originados pela operadora de telefonia. Também fora otimizado serviço Dhcp para fornecer acesso a Internet para as estações de trabalho do Laboratório de Informática e da rede *wireless.*

Em conformidade com a proposta do projeto, este servidor irá oferecer serviços de *proxy, firewall* para controle de pacotes de dados embasados no endereço/porta de origem/destino do pacote, prioridade e etc, controle de detecção de intrusões e um sistema de *hardening,* otimizado no servidor em questão. Para otimização do mesmo, as aplicações utilizadas foram:

3.4.1 DNS - *Domain Name System*

Para otimização do protocolo DNS, utilizou-se a aplicação *Bind*[2]. O mesmo tem seu nome padrão como *Berkeley Internet Name Domain,* onde originou-se no inicio de 1980 na Universidade de *Berkeley,* California. O protocolo DNS é parte integrante do nucleo da Internet, onde especifica o processo pelo qual um computador é embasado em um nome. O *Bind* contem todo *software* necessário para resolver questões relacionadas a serviços de nomes e responder a tais perguntas. É composto por 3 partes:

- *Domain Name System server* - programa denominado *"named",* o qual responde por questões enviadas a ele em concordância com as regras especificadas no protocolo Dns padrão.
- *Domain Name System "resolver library",* o qual resolve questões sobre nomes, envia perguntas aos servidores adequados e responde apropriadamente a tais servidores.
- *Software tools for testing servers,* ferramentas que são utilizadas para testes com intuito de certificar se o servidor está funcionando corretamente.

Devido a Unidade Escolar possuir uma rede de pequeno porte, a mesma utiliza-se dos serviços de DNS da operadora de telefonia. A aplicação *Bind* fora utilizada tão somente para serviços de cache DNS, afim de se evitar problemas de conexão com Internet por motivos relacionados a negação de serviço dos servidores DNS da operadora de telefonia. Para (NEMETH; SNYDER; HEIN, 2007), a implementação do serviço de cache também tem o propósito de aumentar a eficiência das pesquisas:

> *"Uma resposta armazenada em cache é praticamente livre e normalmente está correta porque em geral os mapeamentos de nome de host para endereço raramente mudam. Uma resposta é salva por um período de tempo chamado "Time to Live"(TTL), que é especificado pelo proprietário do registro de dados em questão. A maior parte das consultas é para hosts locais e pode ser solucionada rapidamente. Os usuários também ajudam inadvertidamente a eficiência, pois repetem muitas consultas: após a primeira ocorrência de uma consulta, o restante fica "livre"."(NEMETH; SNYDER; HEIN, 2007)*

[2]https://www.isc.org/software/bind/whatis

3.4.2 *DHCP - Dynamic Host Configuration Protocol*

Conforme (NEMETH; SNYDER; HEIN, 2007) *"as distribuições Linux são historicamente distribuídas com vários servidores e clientes DHCP diferentes. Em dias atuais, as mesmas passaram por um processo de padronização na implementação de referência do ICS*[3] *- Internet System Consortium".* O ISC DHCP é *software open-source* que implementa o *Dynamic Host Configuration Protocol* para a conexão a uma rede local.

O mesmo pode economizar tempo de trabalho do Administrador de Redes, se implementado. Assim que o mesmo estiver operante, os clientes utilizam-no para obter automaticamente suas configurações em tempo de inicialização. O servidor DHCP fora otimizado afim de oferecer serviços de rede para os computadores do Laboratório de Informática. Esse serviço é disponibilizado por motivo dos mesmos serem utilizados pelo corpo discente somente para acesso a Internet, pesquisas escolares, consultas a boletins de notas escolares, entre outros.

O serviço DHCP será otimizado com IP fixo. Conforme (MORIMOTO, 2008), *"esse recurso é usado em redes de terminais leves, para que o servidor "reconheça"os terminais e possa enviar a configuração adequada a cada um, mas pode ser usado também em outras situações, como em uma pequena rede, onde alguns micros compartilham impressoras e arquivos e por isso não podem ficar mudando de endereço IP a cada reboot."*

O serviço DHCP, além de ser otimizado com IP fixo, também terá como controle o endereço MAC de cada estação de trabalho do Laboratório de Informática. O propósito do uso é dificultar uma possivel inclusão de qualquer *hardware* para fazer uso da rede local partindo-se de uma tomada de área de trabalho, a fim de evitar possíveis problemas relacionados a ataques originados dentro da rede local.

Com o intuito de disponibilizar o *gateway* para Internet para a rede *wireless,* a mesma também operará com o serviço Dhcp. Notoriamente essa rede, difere da rede local, pois a mesma irá disponibilizar apenas uma range de endereços IPs sem controle de endereço MAC. Tal medida, deve-se ao uso da configuração da mesma em modo infra-estrutura e consequentemente as estações de trabalho móveis terão seu endereço MAC cadastrado diretamente no AP.

[3] https://www.isc.org/software/dhcp

3.4.3 *NTP - Network Time Protocol*

O NTP[4] é um protocolo para sincronização dos relógios dos computadores onde define uma forma para um grupo de computadores conversar entre si e acertar seus relógios, baseados em alguma fonte confiável de tempo como os relógios atômicos do Observatório Nacional, que definem a Hora Legal Brasileira.

O serviço NTP fora otimizado com intuito de controlar o horário dos servidores e estações de trabalho da rede local do Colégio, sincronizados com o servidor de perímetro em questão e este, com os servidores NTP nacionais. Conforme (RNP - REDE NACIONAL DE ENSINO E PESQUISA, 2000), o benefício do uso do protocolo NTP, engloba usuários e administradores de rede

> *"Os benefícios da utilização do NTP atingem tanto usuários quanto administradores de rede. Pelo lado dos usuários, a sincronização dos relógios de computadores pode ser vital em certas operações. Atrasos de até um ou dois minutos são bastante freqüentes quando não se usa um esquema de NTP. Do ponto de vista da administração de redes, a utilização do NTP é muito vantajosa, pois possibilita a sincronização automática de todos os equipamentos conectados em rede. Ou seja, o administrador não precisa ir de máquina em máquina acertando o relógio local."(RNP - REDE NACIONAL DE ENSINO E PESQUISA, 2000)*

O uso do NTP deu-se por motivos de aplicações relacionadas a segurança da informação, implantadas no presente estudo de caso da rede em questão e, as mesmas utilizarem-se desse meio, afim de corroborar com alguma prova cabal em algum incidente que por ventura possa vir a acontecer. Tal assertiva motivacional para o uso do mesmo pode ser melhor elucidada, conforme segue:

> *"Além disso, a questão da segurança é reforçada com a adoção da sincronização dos relógios dos equipamentos em rede pois a investigação de eventos de ataques em computadores depende da verificação de logs em diversos equipamentos. A inconsistência dos horários registrados inviabiliza esse trabalho."(RNP - REDE NACIONAL DE ENSINO E PESQUISA, 2000)*

[4] http://www.ntp.br/

3.4.4 SSH - Security Shell

Para devida administração do servidor, uma vez que o mesmo não sera acessado diretamente, houve a necessidade de uma aplicação destinada a tal serviço onde para tanto o autor optou por usar o `ssh`. O mesmo é bastante seguro em relação a rlogin[5], rcp[6] e telnet[7] e utiliza-se de criptografia para interagir com outro host. Tal assertiva pode ser melhor explanada por (NEMETH; SNYDER; HEIN, 2007) que detalha comentários a respeito do mesmo.

> *"Ele usa autenticação criptografica para confirmar a identidade de um usuário e criptografa todo o fluxo de comunicações entre os dois hosts. O protocolo utilizado pelo SSH é concebido para resistir a uma série de ataques potenciais. O protocolo está documentado pelas RFCs 4250 e 4256 e é agora um padrão proposto para o IETF."(NEMETH; SNYDER; HEIN, 2007)*

O mesmo sofreu mudanças onde, a partir de projeto de código-fonte aberto com distribuição gratuita, tornou-se produto comercial onde utiliza protocolo diferente - o SSH2. Para a comunidade de código-fonte aberto, em contrapartida disponibilizou o *OpenSSH,* onde implementa os dois protocolos SSH e SSH2 e é mantido pela *OpenBSD.* Criado por *Tatu Ylonen,* tem como principais componentes um *daemon* de servidor - o `sshd` que pode autenticar *logins* de usuários de diversas maneiras diferentes, além de dois comandos para nível de usuários onde o **ssh** torna-se o responsável pelo acesso remoto a outro *host* e o **scp** para executar cópias de arquivos, de grande utilidade para administradores de sistemas.

O mesmo disponibiliza ainda outros componentes como o comando ***ssh-keygen*,** onde gera pares de chaves públicas e diversos utilitários auxiliares no suporte a X *Window* seguro.

[5]rlogin - Aplicativo para Unix que permite fazer login em outro host por meio de uma rede, comunicando-se via TCP, porta 513

[6]rcp - Comando utilizado para cópia de arquivos entre máquinas.

[7]Telnet - Protocolo cliente-servidor usado para permitir a comunicação entre computadores ligados numa rede, porém sem criptografia de dados

3.4.5 *PAM - Pluggable Authentication Modules*

O *PAM - Pluggable Authentication Modules* foi desenvolvido pela *Sun*[8] no intuito de flexibilizar autenticação de usuários. O mesmo no linux acompanha todas as distribuições e tem como conceito onde os programas que necessitam de autenticação só precisam saber que um módulo está disponivel para efetuar a autenticação. É configurado de tal forma que pode ser removido e reconfigurado a qualquer hora sem necessidade de lincagem dos mesmos no momento em que há a compilação de um aplicativo.

O mesmo tem como campos tipo-do-módulo que pode assumir valores `auth, account, session ou password`, *flags* de controle onde possui quatro valores possíveis ou seja, `required, requisite, sufficient e optional`, além do terceiro e quarto campos que oferta o nome de caminho bem como argumentos para o objeto de módulo carregável dinamicamente.

Seu principal uso deu-se por motivo de ofertar uma maior proteção ao acesso ssh. Para tanto, o módulo implementado é pam_abl, o qual consiste em criar um *front-end* de segurança onde evita acessos indevidos por bisbilhoteiros, evitando assim abusos a partir de *hosts* remotos realizando ataques de *brute-force* baseados em dicionários.

Tem-se também a configuração incorreta de um servidor onde permite-se que *hosts* mal intencionados, realizem ataques por um periodo indefinido, causando problemas relacionados a segurança. Conforme (TAMBORIM, 2007) *"pam_abl (Auto Blocking List) efetua uma blacklist de hosts que efetuem logins mal sucedidos no sistema de acordo com a política configurada e evita que um ataque remoto seja realizado, indisponibilizando o acesso ao serviço para o atacante, removendo então o perigo de acesso remoto."*. Ataques *brute-force* envolvem várias tentativas para autenticar-se em um serviço usando um dicionário de senhas comum, onde embora seja necessário a utilização de senhas seguras nem sempre é possivel e por conseguinte o uso de senhas fracas, ataques *brute-force* podem ser eficazes. Os argumentos especificados pela pam_abl, citados por (ARMSTRONG,) podem ser definidos como:

- `host_db, user_db` - Parâmetro onde especifica a base de dados responsável por registrar tentativas de falha de autenticação, registrando o *hostname* responsável. A base de dados dos usuários é usada para registro do nome de

[8] *Sun* - Originario de *Stanford University Network* é uma empresa fabricante de computadores, semicondutores e *software* com sede em Santa Clara, Califórnia, no *Silicon Valley*

usuário solicitado. Para o caso do `host_db e do user_db` serem omitidos, a lista negra correspondente será desativada.

- `host_purge, user_purge` - Especifica o tempo que as tentativas deverão ser mantidas em base de dados, onde para um bom funcionamento deve ser um período mais longo possível, podendo ser aplicadas em até dias da semana.

- `host_rule, user_rule` - Tais regras determinam em quais circunstâncias as contas são *auto-blacklisted,* onde o `host_rule` bloqueia o acesso a maquinas responsáveis por falhas de autenticação em excesso enquanto o parâmetro `user_rule` desativa contas as quais houve excesso de falhas de autenticação.

3.4.6 *Firewall Iptables*

Notoriamente o conceito e uso de *firewall* provem do objetivo de proteger uma rede local ou um *host* contra acessos indesejados e proteção de serviços sendo executados na rede local. A motivação pela escolha do *Iptables* deu-se primeiramente por oferecer maior segurança para a rede local, a rede *wireless* em questão e ser totalmente *open-source.*

É considerado *firewall* que trabalha a nível de pacotes, como porta/endereço de origem/destino, estado de conexão e outros parâmetros de pacote. Para (NETO, 2004), o mesmo tem suas funções agregadas ao kernel.

> *"No Linux, as funções de firewall são agregadas a propria arquitetura do Kernel, isso o torna, sem dúvida, muito superior em relação a seus concorrentes. Equanto a maioria dos produtos firewall pode ser definida como sub-sistema, o Linux possui a capacidade de transformar o Firewall no próprio."(NETO, 2004)*

O *Iptables* fora implantado a partir do Kernel 2.4 em substituição ao *Ipchains,* referentes aos *kernels 2.2.* É chamado também de *Netfilter,* oferece muita flexibilidade para desenvolvimento de regras, controle de tráfego, controle independente de tráfego da rede local, entre redes e interfaces devido a nova organização das etapas de roteamento de pacotes. Tem seu uso para monitoramento de trafego de rede. Conforme o (GUIA FOCA GNU/LINUX,), o *Iptables* tem como caracteristicas:

- Especificação de portas/endereço de origem/destino

- Suporte a protocolos TCP/UDP/ICMP (incluindo tipos de mensagens ICMP)
- Suporte a interfaces de origem/destino de pacotes
- Manipula serviços de *proxy* na rede
- Tratamento de trafego dividido em *chains* (para melhor controle do tráfego que entra/sai da máquina e tráfego redirecionado
- Permite um número ilimitado de regras por *chain*
- Muito rápido, estável e seguro
- Possui mecanismos internos para rejeitar automaticamente pacotes duvidosos ou mal formados
- Suporte a modulos externos para expansão das funcionalidades padrões oferecidas pelo código de *firewall*
- Suporte completo a roteamento de pacotes, tratadas em uma área diferente de tráfegos padrões
- Suporte a especificação de tipo de serviço para priorizar o tráfego de determinados tipos de pacotes
- Permite especificar exceções para as regras ou parte das regras
- Suporte a detecção de fragmentos
- Permite enviar alertas personalizados ao *syslog* sobre o tráfego aceito/bloqueado
- Redirecionamento de portas
- Suporte a *Masquerading*
- Suporte a *SNAT* (modificação do endereço de origem das máquinas para um único IP ou faixa de IP's)
- Suporte a *DNAT* (modificação do endereço de destino das máquinas para um único IP ou fixa de IP's)
- Contagem de pacotes que atravessaram uma interface/regra
- Limitação de passagem de pacotes/conferência de regra (muito útil para criar proteções contra *syn flood, ping flood,* DoS, etc).

3.4.7 *Proxy Squid*

Conceituando-se os serviços de *proxy,* pode-se dizer que o mesmo efetua a complementação de um *firewall* para segurança de uma rede e também, no armazenamento de dados economizando-se assim o uso de largura de banda. Tal assertiva é melhor descrita por (BALL; PITTS; et al., 2002) "*um firewall de filtragem de pacotes básico pode ser complementado com um proxy para melhorar a sua segurança e, em alguns casos, para armazenar dados para diminuir o uso da largura de banda da rede."* Serviços de proxy, tomam decisões acima da camada de pacote ou camada de transporte. Conforme (CHESWICK; BELLOVIN; RUBIN, 2005) destaca.

> *"Os proxies podem ser utilizados para tomar decisões de filtragem com base nas informações acima da camada do pacote ou da camada de transporte inteira. Eles também são utilizados para definir regras de filtragem de pacotes muito simples, enquanto passam a complexidade para outra pessoa."(CHESWICK; BELLOVIN; RUBIN, 2005)*

O *proxy* mais conhecido e utilizado no *GNU/Linux* é o *Squid*[9], onde o mesmo tem suporte aos protocolos HTTP, HTTPS, FTP via HTTP e GOPHER. Reduz uso de largura de banda e melhora os tempos de resposta ao reutilizar frequentemente páginas *Web,* solicitadas ao *caching.* É usado por provedores de Internet, com o intuito de melhora no acesso a Internet. Para uma rede local, permite o compartilhamento da conexão de Internet entre vários micros. Conforme (MORIMOTO, 2008), as vantagens de utilizar-se um serviço proxy são:

- É possível impor restrições de acesso com base no horário, login, endereço IP da máquina e outras informações, além de bloquear páginas com conteúdo indesejado. É por isso que quase todos os softwares de filtro de conteúdo envolvem o uso de algum tipo de *proxy,* muitas vezes o próprio *Squid* (já que, como o *software* é aberto, você pode incluí-lo dentro de outros aplicativos, desde que respeitando os termos da GPL).

- O *squid* funciona também como um *cache* de páginas e arquivos, armazenando informações já acessadas. Quando alguém acessa uma página que já foi carregada, o *squid* envia os dados que guardou no *cache,* sem precisar acessar a mesma página repetidamente. Isso acaba economizando bastante

[9] http://www.squid-cache.org/

banda, tornando o acesso mais rápido. Hoje em dia, os sites costumam usar páginas dinâmicas, onde o conteúdo muda a cada visita, mas, mesmo nesses casos, o *squid* dá uma boa ajuda, pois embora o HTML seja diferente a cada visita e realmente precise ser baixado de novo, muitos componentes da página como ilustrações, banners e animações em *flash,* podem ser aproveitados do *cache,* diminuindo o tempo total de carregamento. Dependendo da configuração, o *squid* pode apenas acelerar o acesso às páginas ou servir como um verdadeiro *cache* de arquivos, armazenando atualizações do *Windows Update, downloads* diversos e pacotes instalados através do apt-get, por exemplo. Em vez de ter que baixar o último *Service Pack* do *Windows* ou a última atualização do *Firefox* nos 10 micros da rede, o usuário vai precisar baixar apenas no primeiro, pois os outros 9 vão baixar a partir do *cache* do *Squid.*

- Uma terceira vantagem de usar um *proxy* é que ele loga todos os acessos realizados através dele. O usuário pode visualizar os acessos posteriormente usando o *Sarg,* um gerador de relatórios que transforma as longas listas de acessos dos logs em arquivos Html bem organizados.

3.4.8 *SARG - Squid Analysis Report Generator*

Para análise do conteúdo gerado pelo *proxy-squid,* o autor utilizou-se da ferramenta *Sarg*[10]. A mesma fora desenvolvida por Pedro Orso e tem como função gerar relatórios de analise de acesso através do squid, onde mostra informações acerca do comportamento de usuários, IPs, *bytes, sites* visitados e periodicidade dos mesmos. Bastante útil e de fácil utilização, busca informações contidas no *access.log* e cria páginas Html com o intuito de aprimorar a apresentação dos dados. Através do mesmo pode-se ver quais sites foram acessados, mais visitados, bloqueados, acesso negado entre outros. A respeito da organização dos *log's* (MORIMOTO, 2008) descreve sua organização quanto a arquivos de log novos e arquivos de log antigos.

> *"Os logs são inicialmente organizados por período, sendo que os relatórios antigos são mantidos quando o relatório é atualizado (com o tempo o relatório acaba armazenando um volume muito grande de informações). Dentro do relatório de cada período, você tem a lista dos endereços IP e/ou dos usuários autenticados que utilizaram o proxy*

[10] SARG - *Squid Analysis Report Generator* - http://sarg.sourceforge.net/pt-sarg.php

e, dentro do relatório referente a cada um, você pode acompanhar o log das páginas acessadas e outras informações, de forma bastante detalhada."(MORIMOTO, 2008)

Para que o mesmo seja executado automaticamente, há a necessidade do mesmo ser incluído no *cron.* A partir da sua instalação baseada na distribuição Debian, ele cria automaticamente um *script* o qual ele é automatizado para ser executado todos os dias as 6:25 horas em qualquer instalação.

3.4.9 *nmap* - Scanneamento de portas

Para análise de possíveis vulnerabilidades de segurança, tem como principal função, a verificação de *hosts* ou conjunto de *hosts* com o intuito de averiguar quais portas possuem servidores escutando nas mesmas, portas essas TCP e/ou UDP.

Possui um repertório de maneiras dissimuladas onde faz a sondagem de portas sem reiniciar conexão e, na maioria dos casos investiga pacotes que parecem provenientes do meio de uma conversa TCP, aguardando logicamente o envio de retorno dos pacotes de diagnóstico.

São sondagens indetectáveis e capazes de passar por um *firewall* ou evitar a detecção por monitores de segurança de rede na busca por varreduras de portas. Seu uso deu-se por motivo de analisar portas utilizadas no projeto em questão, bem como avaliar alguma anomalia na mesma. Tem a capacidade ainda, de advinhar qual sistema operacional esta sendo utilizado em um sistema remoto e analisando detalhes da sua implantação TCP/IP. Sua utilização é bastante prática e tem sempre como sintaxe de uso o comando `nmap <parâmetros> <alvo> -p <portas>` onde:

- nmap - Comando propriamente para varredura de rede ou *host*
- parâmetros - parâmetros passados para o comando nmap tais como: sT - onde é feito um escaneamento através de tentativas de conexão TCP.
- alvo - endereço IP do *host* ou rede que deseja-se escanear.
- portas - especificação de portas ou faixas de portas para análise.

3.4.10 *Snort* - Sistema de detecção de intrusão

Acerca da administração de um IDS[11], há necessidade significativa de recursos, onde os mesmos têm obrigatoriedade de serem instalados em pontos estratégicos, serem adequadamente otimizados e monitorados. Na maior parte dos ambientes instalados, lidam com quantidade surpreendentemente grande de tráfego irregular de rede. Para o trabalho em questão, o autor escolheu o uso do Snort[12].

Notoriamente, o mesmo é uma aplicação com intuito de detectar e prever anomalias nos pacotes de dados de uma rede. Foi desenvolvido por *Martin Roesch* e tem seu codigo fonte aberto. Bastante popular por sua flexibilidade nas configurações de regras e constante atualizações, além de funcionar em várias plataformas onde o libpcap[13] possa ser executado. Conforme (CHESWICK; BELLOVIN; RUBIN, 2005) destaca:

> *"O Snort pode ser utilizado de diversas maneiras: pode farejar uma rede e produzir saida formatada em tcpdump[14], bem como ser utilizado para registrar pacotes, de modo que ferramentas de exploração de dados e programas de terceiros podem fazer análise do tráfego da rede após uma ocorrência. O recurso mais interessante dele é sua capacidade de definir um conjunto de regras que reconhece certos padrões de tráfego. Muitas dessas regras estão disponíveis para o Snort e frequentemente são compartilhadas entre os usuários e postadas na Internet."(CHESWICK; BELLOVIN; RUBIN, 2005)*

3.4.11 Tripwire

Notoriamente em um projeto para segurança de perímetro de uma rede local, há sempre a preocupação com eminentes ataques os quais surgem dentro da própria rede local, ou vindo pela Internet. Em tais circunstâncias, é de suma importância garantir a integridade dos arquivos de dados utilizados no servidor.

Em um primeiro momento listar todos os arquivos de dados do mesmo e sua localização dentro do sistema operacional. O processo para detecção de possíveis

[11] *IDS - Intrusion Detection System* (sistema de detecção de intrusão)

[12] http://www.snort.org/

[13] *libpcap* - consiste de uma interface de programação de aplicativos (API) para capturar o tráfego da rede

[14] http://www.tcpdump.org/ - é uma ferramenta utilizada para monitorar os pacotes trafegados em uma rede de computadores

alterações é feito por amostragem comparando-se o relatório gerado no início de sua implantação, com relatórios gerados em tempos futuros.

A aplicação escolhida para esse serviço foi *Open-source Tripwire*[15], onde *"a mesma mantem a funcionalidade do núcleo original de integridade de dados para os servidores. Utilizando uma interface de linha de comando, o mesmo irá detectar alterações em cada servidor no qual está sendo instalado, alertando os usuários para invasões e mudanças inesperadas". (WWW.TRIPWIRE.COM,)*

3.4.12 *Hardening*

O conceito básico de *Hardening*, como a tradução do proprio nome expõe, é tornar o acesso a uma determinado sistema operacional o mais restritivo possível, envolvendo o máximo possivel de segurança aplicada no *hardware e software,* onde os mesmos são fortalecidos com intuito de serem acessados somente por administradores do sistema.

Para *hardware,* a melhor forma de assegurar sua integridade é evitar-se ao máximo o acesso físico ao mesmo por pessoas não autorizadas e tentar a proteção de suas configurações atraves de senhas de acesso. Para sistema operacional, se deve usar sistema de senhas a fim de restringir-se o acesso a determinadas áreas, permitir acesso somente por administradores responsáveis, desativar aplicações sem utilidade para o propósito do servidor, bem como remover grupos e usuários não necessários para o servidor em questão. Conceituações a respeito, tem o propósito de minimizar o máximo possível o risco de acessos indevidos, ocasionados por agentes invasores na rede local ou Internet. Conforme (NORTHCUTT *et al.*, 2005) expõe:

> *"É preciso modificar a configuração afim de expor o host o minimo possível diminuindo sua exposição a ameaças. O grau de hardening aplicado, depende muito do serviço que o sistema executa. Uma vez o hospedeiro devidamente otimizado, pode atuar como um colaborador eficaz para um perímetro de segurança de rede confiável". (NORTHCUTT* et al.*, 2005)*

[15] http://www.tripwire.org/

3.5 Uso de Roteador *Wireless*

Afim de ofertar apoio pedagógico ao corpo docente da Unidade Escolar, fora encampado à proposta do novo projeto da rede o sinal *wireless,* uma vez que a totalidade do corpo docente possui dispositivos móveis. Conforme (JARDIM, 2007), as redes Wi-Fi *"que vem do termo Wireless Fidelity ou (fidelidade sem fios), tornou-se a tecnologia de mais rápida adoção no mundo wireless dos ultimos anos e já se encontra em muitos dispositivos computacionais".* Wi-Fi é o nome comercial usado para designar o conjunto de padrões de rede *wireless* desenvolvido pelo comitê 802.11 do IEEE[16].

Tornou-se a de mais rapida adoção no mundo computacional nos ultimos quatro anos e é dividida em três principais padrões: 801.11b, 802.11a e 802.11g onde o padrão 802.11b foi o primeiro a ganhar força no mercado nacional a partir do ano de 2002. Para o projeto em questão, o padrão utilizado foi 802.11g, conhecido como Wi-Fi2 com solidificação do seu uso no final do ano de 2003 e inicio de 2004. O mesmo possui taxa homologada padrão de 54 Mbps e frequênca de 2,4 Ghz. No inicio do ano de 2005 os fabricantes de dispositivos enquadrados no padrão 802.11g começaram lançar no mercado novo recurso com apelido de *Speedbuster* ou super G, o qual permite que o padrão 802.11g passe a trabalhar a 108 Mbps, sendo necessário a aquisição do equipamento que comporte tal funcionalidade. Porém em setembro do ano de 2006, grande parte dos dispositivos Wi-Fi com padrão 802.11g ja tinham tal recurso. Diante do exposto, nota-se que a rede utilizando padrão 802.11g pode ultrapassar a rede cabeada, utilizando o padrão 802.3.

O modo de operação adotado para o projeto em questão, é de infra-estrutura, onde tem-se um dispositivo centralizador, denominado de AP *(Access Point)* ou ponto de acesso). Tal modo, utiliza o conceito de BSA (*Basic Service Area*) que representa a área na qual os dispositivos móveis podem trocar informações. A área abrangida por um AP é denominada BSS *(Basic Service Set)*, onde possui um identificador SSID *(Service Set Identifier)*.

Para tanto, com o intuito de oferta-la com segurança e qualidade na distribuição do sinal, o roteador *wireless* será alocado em ponto estratégico, ou seja, no centro do recinto escolar, tornando o acesso por terceiros não autorizados dificultoso e, com o intuito de amenizar problemas relacionados a segurança fisica.

[16] *IEEE - Institute of Engineering of Electrical and Electronics*

> *"Administradores tendem a cuidar muito da segurança lógica e, em geral, dão pouca atenção à segurança física, até por que, geralmente, nas organizações a área de segurança física esta vinculada a outros departamentos não subordinados à área de tecnologia da informação, o que, em geral, é um erro estratégico."(RUFINO, 2007)*

Comparando-se o risco da segurança física de redes cabeadas e equipamentos *wireless,* (RUFINO, 2007) mostra que *"se a segurança física é um importante componente de risco quando se trata de redes cabeadas, em redes sem fio esse aspecto é ainda mais relevante, visto que a área de abrangência "física"aumenta substancialmente."* Com relação à questão de segurança, os dispositivos móveis a serem utilizados na rede *wireless,* terão cadastrados no roteador seu endereço MAC, configurado através filtro de endereçamento MAC.

O protocolo de autenticaçao utilizado será o WPA2-PSK, onde devido a pequena quantidade de dispositivos moveis utilizados na rede diariamente, o autor optou pelo não uso de servidor *radius.* O uso do mesmo, deu-se por motivos de ofertar maior segurança, pois utiliza como sistema de encriptação o AES, bastante seguro e baseia-se no uso de chaves com 128 a 256 bits.

Capítulo 4

Implementação do projeto

4.1 *Hardening* proposto para o servidor

Afim de torná-lo o mais seguro possível o autor, após suas pesquisas, iniciou o trabalho de *hardening* alterando a montagem de alguns dos dispositivos. Para (GOMES *et al.*, 2001), mudanças no particionamento torna certos diretórios diferenciados, oferecendo melhor segurança aos mesmos. As alterações aplicadas estão explicitadas na figura 4.1 e, as opções especiais utilizadas foram:

- *noexec* - Onde não pode ser executado o conteúdo a partir do ponto de montagem. O mesmo fora aplicado respectivamente para /boot e /tmp.
- *nosuid* - Não permite operação de *bits suid.* Aplicados para /boot e /tmp respectivamente.
- *nodev* - Não permite dispositivos especiais de bloco ou caractere do sistema de arquivos. Aplicado em /boot, /tmp e /usr
- *ro* - Especifica partições que serão somente leitura. Aplicado em /boot e /usr.

Com intuito de aumentar a segurança do servidor, aplicativos não necessários ao propósito do projeto, foram desinstalados e também não inicializados durante o boot do sistema. Para melhor gerenciamento do diretório *init.d*[1], utilizou-se o

[1] *init.d* - diretório que contém scripts para a inicialização de serviços da máquina

```
squalidus:/home/suporte# cat /etc/fstab
# /etc/fstab: static file system information.
#
# <file system> <mount point> <type> <options>      <dump> <pass>
proc            /proc         proc   defaults            0   0
/dev/sda2       /             ext3   errors=remount-ro   0   1
/dev/sda1       /boot         ext3   noexec,nosuid,nodev,ro 0   2
/dev/sda7       /tmp          ext3   noexec,nosuid,nodev 0   2
/dev/sda5       /usr          ext3   nodev,ro            0   2
/dev/sda6       /var          ext3   defaults            0   2
/dev/sda3       none          swap   sw                  0   0
```

Figura 4.1: Saída do fstab com alterações

file-rc.[2] Para o diretório `init.d`, foram aplicados os comandos *chown* e *chmod* respectivamente, afim de disponibilizar o acesso somente a `root`.

Listando-se o sistema operacional, encontrou-se diversas aplicações não necessárias ao projeto proposto tais como o servidor de e-mail *Exim4* e suas relações, *Portmap e NFS* as quais foram removidas. Em conformidade com a proposta do projeto, houve algumas alterações no arquivo *inittab*, onde desabilitou-se o uso do Ctrl+Alt+Del e restringiu-se o uso a dois terminais. A figura 4.2 traz as linhas alteradas no arquivo *inittab*.

De acordo com (TURNBULL, 2005), deve atentar-se para o arquivo `/etc/securetty`, limitando-se o acesso ao servidor através dos terminais `tty1 e tty2`. Tal configuração juntamente com permissões somente para escrita e leitura permitidas ao *root,* visa dificultar qualquer outro tipo de acesso por outros meios ao servidor em questão. Muitos usuários e grupos já definidos por padrão no sistema operacional, também não são necessários. Para tanto, no intuito de aumentar a segurança do servidor em questão, removeu-se alguns usuários e grupos. Para a lista de usuários removidos tem-se games, gnats, irc, list, lp, mail, news, nobody, proxy, sync, uucp, www-data; para a lista de grupos removidos tem-se adm, operator, src, staff, users. Após essas configurações, finalizou-se o serviço de hardening ora executado no servidor em questão.

[2]File-rc: Aplicativo alternativo onde usa um único arquivo de configuração, de facil administração e bastante flexível.

```
# What to do when CTRL-ALT-DEL is pressed.
#ca:12345:ctrlaltdel:/sbin/shutdown -t1 -a -r now

1:2:respawn:/sbin/getty 38400 tty1
2:2:respawn:/sbin/getty 38400 tty2
```

Figura 4.2: Alteraçoes executadas no *inittab*

4.2 Configurando interfaces de rede

De acordo com o projeto proposto utilizou-se três interfaces de rede, onde a interface eth0 cujo IP é 10.1.1.2, é responsável por conectar-se com o modem ADSL, a interface eth1 cujo seu endereço IP é 192.168.0.254, é responsável pela conexão com a rede local, o qual disponibiliza endereçamento ip fixo iniciando em 192.168.0.1 até 192.168.0.10, além de uma range de endereços por `DHCP`, concatenado com o endereço MAC de cada interface de rede, visando dificultar o uso da rede local por qualquer outra estação de trabalho.

A interface de rede eth2, cujo seu endereço IP é 192.168.3.254, responde pela conexão da rede wireless, onde disponibiliza o acesso por `DHCP`. Essa outra faixa de rede, fora implementada afim de propor separação entre a rede local e a rede wireless. A Figura 4.3 mostra a configuração das interfaces de rede, onde após sua configuração restartou-se o serviço de rede.

4.3 Configurando protocolo NTP

O protocolo NTP fora configurado com base nos servidores nacionais. Na Figura 4.4 tem-se o arquivo de configuração dos servidores que respondem ao serviço `ntp`. Após a instalação do ntp criou-se o arquivo ntp.drift o qual é considerado "memória responsável para o escorregamento de frequencia do micro". Após edita-se o arquivo ntp.conf e reinicia-se o serviço NTP. Na Figura 4.5, se tem a saida dos servidores `NTP`, listados através do comando `ntpq -c pe`.

```
squalidus:/home/suporte# cat /etc/network/interfaces

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug eth0
iface eth0 inet static
        address 10.1.1.2
        netmask 255.255.255.0
        network 10.1.1.0
        broadcast 10.1.1.255
        gateway 10.1.1.1

# The secondary network interface
allow-hotplug eth1
iface eth1 inet static
        address 192.168.0.254
        netmask 255.255.255.0
        network 192.168.0.0
        broadcast 192.168.0.255

# The terciary network interface
allow-hotplug eth2
iface eth2 inet static
        address 192.168.3.254
        netmask 255.255.255.0
        network 192.168.3.0
        broadcast 192.168.3.255
```

Figura 4.3: Configuração Interfaces de Rede

4.4 Otimizando DNS-cache com *BIND*

A aplicação utilizada para o serviço de DNS-cache foi o *BIND*[3]. A versão instalada é 9.5.1. Para o propósito de DNS-cache somente será necessário a edição do arquivo `/etc/bind/named.conf.options`, incluindo os servidores DNS alternativos à operadora de telefonia, as opções de segurança e desabilitando-se o IPv6. Para o funcionamento do DNS-cache, alterou-se também a configuração

[3] *BIND - Berkeley Internet Name Domain* originado no inicio de 1980 na Universidade *Berkeley* - Califórnia

```
squalidus:/etc# cat ntp.conf
driftfile /etc/ntp.drift

# estatisticas do ntp que permitem verificar o historico
# de funcionamento e gerar graficos
statsdir /var/log/ntpstats/
statistics loopstats peerstats clockstats
filegen loopstats file loopstats type day enable
filegen peerstats file peerstats type day enable
filegen clockstats file clockstats type day enable

# servidores publicos do projeto ntp.br
server a.ntp.br iburst
server b.ntp.br iburst
server c.ntp.br iburst

# configuracoes de restricao de acesso
restrict default kod notrap nomodify nopeer
```

Figura 4.4: Arquivo de configuração do serviço ntp

```
squalidus:/etc# ntpq -c pe
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
*a.ntp.br  200.160.7.186   2 u   22   64   37   30.299 40.159  4.834
+b.ntp.br  200.20.186.76   2 u   19   64   37   34.482 40.325  2.628
+c.ntp.br  200.160.7.186   2 u   16   64   37   43.614 39.298  2.271
```

Figura 4.5: Servidores ntp listados

dos arquivos `hosts e resolv.conf`. A Figura 4.6 elucida a configuração dos arquivos.

4.5 Otimizando serviço DHCP

Notoriamente quse toda rede usa serviços `DHCP` em dias atuais. Para o projeto em questão, serão disponibilizados para a rede local uma faixa com 12 endereços `DHCP` com IP fixo, concatenados com endereço MAC. De acordo com (MORIMOTO, 2008) *"torna-se uma opção de grande interesse, pois o mesmo obtem o mesmo endereço a partir do servidor DHCP, atribuindo somente aos endereços*

```
squalidus:/home/suporte# cat /etc/bind/named.conf.options
options {
        directory "/var/cache/bind";
        forwarders {
                // Servidor DNS Root Master
                4.2.2.2;
                // Servidores DNS Fapesp
                208.67.222.222;
                200.160.0.10;
        };
        // Security Options - Rede Local e Rede Wireless
        listen-on { 127.0.0.1; 192.168.0.254; 192.168.3.254; };
        allow-query { 127.0.0.1; 192.168.0.0/24; 192.168.3.0/24; };
        allow-recursion { 127.0.0.1; 192.168.0.0/24; 192.168.3.0/24; };
        allow-transfer { none; };
        auth-nxdomain no;    # conform to RFC1035
        //      listen-on-v6 { any; };
};

squalidus:/home/suporte# cat /etc/resolv.conf
nameserver 127.0.0.1

squalidus:/home/suporte# cat /etc/hosts
127.0.0.1       localhost.localdomain   localhost
```

Figura 4.6: Configuração named.conf.options, resolv.conf e hosts

MAC[4] *cadastrados com o endereço IP."* A faixa de endereços disponivel incia-se em 192.168.0.21 e termina em 192.168.0.32. Para a rede *wireless,* fora disponibilizado o serviço `DHCP` simples, sem configuração alguma. Após a instalação do serviço `DHCP` configurou-se o serviço, conforme a Figura 4.7, após sua configuração, editou-se o arquivo `/etc/default/dhcp3-server` informando-se as interfaces de rede que irão responder ao serviço `DHCP`. Após restartou-se o serviço.

4.6 *Firewall - Iptables*

O *Iptables,* rotulado como o responsável por todo o controle de entrada e saída de pacotes da rede local e também da rede *wireless,* teve como ponto de partida o princípio onde o que não é permitido, está sumariamente proibido. O *Iptables*

[4] *MAC - Media access control* - controle de acesso ao meio, onde o dispositivo efetua a comunicação com outro dispositivo

```
squalidus:/home/suporte# cat /etc/dhcp3/dhcpd.conf
# Configuração dhcp server
# Responder as maquinas lab. informatica com MAC-ADDRESS - Rede Local
# Responder a rede Wireles

ddns-update-style none;
default-lease-time 600;
max-lease-time 7200;
authoritative;

# Rede Local
        subnet 192.168.0.0 netmask 255.255.255.0{
        range 192.168.0.21 192.168.0.32;
        option routers 192.168.0.254;
        option domain-name-servers 192.168.0.254;
        option broadcast-address 192.168.0.255;

# Micros concatenados por MAC Address
        host lab_micro1 {
        hardware ethernet 00:0F:B0:55:EA:13;
        fixed-address 192.168.0.21;
        }

# Rede Wireless
        subnet 192.168.3.0 netmask 255.255.255.0{
        range 192.168.3.0 192.168.0.34;
        option routers 192.168.3.254;
        option domain-name-servers 192.168.3.254;
        option broadcast-address 192.168.3.255;
        }

squalidus:/home/suporte# cat /etc/default/dhcp3-server

INTERFACES="eth1 eth2"
```

Figura 4.7: Otimização do serviço dhcp

fora otimizado como um *script* no padrão *system V*[5], juntamente com controle de versões afim de analisar alterações feitas no *firewall* durante sua operação e uso no servidor em questão. Conforme tabela 4.1, pode-se notar as portas utilizadas

[5] *System V* - define como deve ser a inicialização dos serviços do sistema

para o projeto em questão, bem como seus respectivos protocolos e suas devidas chains.

	Chains	Input	Input	Output	Output	Forward	Forward
Serviços	Portas/Protocolos	TCP	UDP	TCP	UDP	TCP	UDP
FTP	20			*Sim*		*Sim*	
FTP	21			*Sim*		*Sim*	
SSH	59327	*Sim*					
DNS	53	*Sim*	*Sim*	*Sim*	*Sim*		
HTTP	80			*Sim*			
POP	110					*Sim*	
NTP	123		*Sim*		*Sim*		
HTTPS	443			*Sim*			
SSL	563			*Sim*			
SMTP	587					*Sim*	
MSN	1863			*Sim*		*Sim*	
SQUID	3128	*Sim*					

Tabela 4.1: Portas utilizadas - *Iptables*

A versão utilizada do *Iptables* é 1.4.2 compativel ao *kernel* `2.6.26-2-amd64` utilizado pelo sistema operacional *GNU/Linux* Debian. Para essa versão e no intuito de liberar acesso a arquivos `ftp`, carregou-se os modulos auxiliares, pois o mesmo responde ao controle de acessos ftp, uma vez que as médias escolares são enviadas diretamente para o site da unidade escolar, partindo-se diretamente do servidor, cujo endereço IP é 192.168.0.1. A nível de proteção do *kernel,* o mesmo fora protegido contra *syn-flood, ipspoofing, ping broadcast,* frames inválidos e *timestamps.* A Figura 4.8 mostra os modulos auxiliares bem como os comandos utilizados para proteção do *kernel.*

Tratando-se a *chain* `input`, direcionada ao roteador em questão, foram liberadas as portas para os serviços `ssh`[6], `DNS`, protocolo `NTP` e proxy `SQUID`. Em contrapartida, bloqueou-se o uso do comando *ping,* descartou-se pacotes inválidos e protegeu-se a *chain* contra *port scanners* ocultos. Para a *chain* `output` liberou-se os protocolos `DNS, FTP, HTTP, HTTPS, NTP` e `ssl`[7]. Além dos protocolos citados, fora necessário também a liberação da porta 1863, relacionada ao serviço *MS-Messenger.*

[6]SSH - Aplicativo utilizado para acesso remoto onde a conexão entre o cliente e o servidor é executada de modo criptografado.

[7]*SSL - Secure socket layer,* é um protocolo desenvolvido pela *Netscape* para transmissão de documentos privados através da Internet.

```
# Carregando modulos para ftp...
/sbin/modprobe ip_conntrack
/sbin/modprobe ip_conntrack_ftp
/sbin/modprobe ip_nat_ftp

# Proteção ao kernel
# Proteçao syn-flood...
sysctl -w net.ipv4.tcp_syncookies=1 > /dev/null 2>&1

# Proteção ipspoofing...
sysctl -w net.ipv4.conf.all.rp_filter=1 > /dev/null 2>&1

# Proteçao ping broadcast...
sysctl -w net.ipv4.icmp_echo_ignore_broadcasts=1 > /dev/null 2>1&

# Protecao frames invalidos...
sysctl -w net.ipv4.icmp_ignore_bogus_error_responses=1 > /dev/null 2>1&

# Protecao contra timestamps
sysctl -w net.ipv4.tcp_timestamps=1 > /dev/null 2>1&
```

Figura 4.8: Modulos carregados para ftp e protecao ao Kernel

Para a *chain* `forward`, o encaminhamento fora feito somente para protocolos `FTP`[8] `POP`[9], `SMTP`[10] e `MSN`.[11] Os protocolos, POP e SMTP foram respectivamente direcionados das estações de trabalho responsáveis por acesso as contas de e-mail, aos servidores de correio eletrônico do *Internet Service Provider,* respondendo respectivamente nas portas 110 e 587 e, respectivamente nos endereços utilizados para acesso as contas de e-mail do Colegio. O protocolo FTP também fora liberado somente ao servidor de arquivos e banco de dados da unidade escolar, uma vez que o sistema de gestão da escola, hospeda as médias escolares do corpo discente automaticamente.

Afim de impor melhor segurança para a rede local, esta estará impossibilitada de receber qualquer tipo de pacotes provenientes da rede *wireless* destinada ao uso do corpo docente da unidade escolar. Essa segurança é também imposta para a rede *wireless,* onde a mesma não receberá qualquer tipo de pacotes de dados proveni-

[8] *FTP - File Transfer Protocol* é um protocolo para transferencia de arquivos.
[9] *POP - Post office protocol,* relacionado a recebimento de mensagens via e-mail.
[10] *SMTP - Sender message transfer protocol*
[11] *MESSENGER* - Aplicativo para comunicação em tempo real, desenvolvido pela *Microsoft.*

entes da rede local, o qual é destinada ao uso do corpo administrativo e discente da unidade escolar. A Figura 4.9 mostra as regras aplicadas a tais controles.

```
# Liberando acesso conta pop e smtp para area Administrativa
# Porta pop 110
IPTABLES -A FORWARD -p tcp -s 192.168.0.2 -d
 pop3.colegioadv.com.br --dport 110 -j ACCEPT

# Porta pop 587
IPTABLES -A FORWARD -p tcp -s 192.168.0.2 -d
 smtp.colegioadv.com.br --dport 587 -j ACCEPT

# Bloqueando acesso da Rede Wireless para Rede Local
IPTABLES -A FORWARD -s  192.168.3.0/24 -i 192.168.3.254 -d
 192.168.0.0/24 -o 192.168.0.254 -j DROP

# Bloqueando acesso da Rede Local para Rede Wireless
IPTABLES -A FORWARD -s  192.168.0.0/24 -i 192.168.0.254 -d
 192.168.3.0/24 -o 192.168.3.254 -j DROP
```

Figura 4.9: Acesso a contas de e-mail e bloqueio da rede wireless

As regras para o NAT impostas para o servidor em questão, foram aplicadas ao final do *script* do *firewall,* onde habilitou-se o encaminhamento de pacotes, redirecionou-se todo o tráfego proveniente da porta 80 para porta 3128 do *SQUID,* bem como, fora feito o mascaramento para o acesso as contas de e-mail do colegio, respondendo nas portas 110 e 587 respectivamente e, também a liberação para o serviço *FTP* utilizado no servidor para envio de médias escolares para o site da unidade escolar automaticamente. A Figura 4.10 traz exemplificações das regras utilizadas no *script.*

4.7 *Proxy - SQUID*

O *proxy Squid* disponibilizado, irá cuidar do acesso HTTP e HTTPS, uma vez que os serviços de FTP serão executados diretamente pelo *firewall,* não tendo passagem pelo serviço de *proxy.* Com o intuito de disponibilizar *gateway* para Internet para a rede local e para a rede *wireless,* o autor utilizou-se de *proxy* autenticado, onde estações de trabalho e unidades móveis necessitam obrigatoriamente de nome de usuário e senha para acesso a Internet.

No intuito de manter-se em arquivos de log todos acessos, houve a necessidade de cadastramento de usuário e senha para cada membro das áreas administra-

```
# Habilitando encaminhamento de pacotes no Kernel
sysctl -w net.ipv4.ip_forward=1 > /dev/null 2>&1

# Direcionando porta 80 para o squid na porta 3128
IPTABLES -t nat -A PREROUTING -i 10.1.1.2 -p tcp --dport 80 -j
 REDIRECT --to-port 3128

# Mascaramento para acesso  pop - Porta 110
IPTABLES -t nat -A POSTROUTING -p tcp -s 192.168.0.2 -d
 pop3.colegioadv.com.br --dport 110 -j MASQUERADE

# Mascaramento para acesso smtp - Porta 587
IPTABLES -t nat -A POSTROUTING -p tcp -s 192.168.0.2 -d
 pop3.colegioadv.com.br --dport 587 -j MASQUERADE

## Habilitando o ftp para o servidor de arquivos
IPTABLES -t nat -A POSTROUTING -s 192.168.0.1 -d
 asp.colegioadv.com.br -j SNAT --to $IP_IF_EXT
```

Figura 4.10: Encaminhamento pacotes e mascaramento de portas

tiva, corpo docente e discente da unidade escolar. Para o cadastramento de usuário e senha utilizou-se do script `htpasswd`, o qual faz parte do pacote apache2-utils. Após sua instalação, houve o processo de cadastramento de usuários, utilizando-se como parâmetro a opção `-m` encriptando a senha do usuário cadastrado através do MD5. A Figura 4.11 mostra o uso do script `htpasswd` com encriptação MD5, bem como o arquivo de senhas do *SQUID*.

```
squalidus:/home/suporte# htpasswd -m /etc/squid/squid_passwd
 valeriasantos
New password:
Re-type new password:
Updating password for user valeriasantos


squalidus:/home/suporte# cat /etc/squid/squid_passwd
antoniopazebao:$apr1$.8VcmObd$dcTS5jz05sBGVVaMeySQ.1
patriciarivera:$apr1$kUvGK6Lo$BzA/Cw1i7cVhVuzCMprdZ.
meireserino:$apr1$1fYegJGu$mts3FgikoDKR3tiREySLx.
valeriasantos:$apr1$lJ958yM4$Pm9XMl6nJeXNPjzY.f34C1
```

Figura 4.11: Comando htpasswd e o arquivo de usuarios do squid

Para cada estação de trabalho da rede local houve a necessidade de informar-se o seu *gateway* ou seja, 192.168.0.254, onde é aplicado ao *browser*[12] O mesmo foi aplicado na rede *wireless* onde a mesma possui o *gateway* 192.168.3.254 e também é aplicado ao *browser* utilizado pela unidade móvel em questão. As Figuras 4.12 e 4.13 mostram respectivamente a tela de informação do *gateway* para rede local e para a rede *wireless.*

Figura 4.12: Configuração do *browser* - estação de trabalho rede local

As regras utilizadas no *SQUID* para o recinto escolar basearam-se em bloqueios por extensões, por IP, por palavras e por URLs organizadas e disponibilizadas em um diretório específico. Tais regras ora criadas, tem o propósito de coibir o acesso a conteúdos não condizentes com o recinto escolar, uma vez que o acesso a sites indesejados podem trazer riscos desnecessários para a rede local, bem como o comprometimento de informações e dados condizentes ao ramo de negócio da unidade escolar.

[12] *Browser ou web browser* - conhecido pelos termos ingleses é um programa de computador que habilita seus usuários a interagirem com documentos virtuais da Internet, também conhecidos como páginas da *web* utilizado na estação de trabalho.

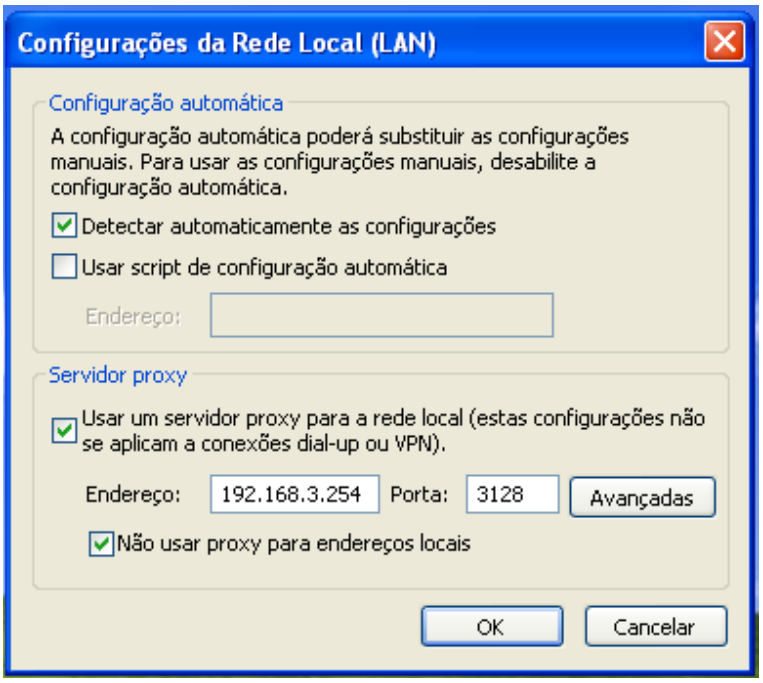


Figura 4.13: Configuração do *browser* - unidade móvel rede wireless

A liberação de atualizações de sistemas operacionais, anti-vírus e aplicativos para as estações de trabalho, foram permitidas somente para a rede local. Com o presente modo de estruturação, facilita-se alterações nas regras do *SQUID,* além da compreensão da estrutura do mesmo por uma outra pessoa que possa vir administrá-lo. Tal assertiva a respeito das configurações do *SQUID,* são descritas na Figura 4.14, onde tem-se um fragmento da configuração basica do mesmo.

4.8 SARG - Gerando relatorios

O *SARG (Squid Analisys Report Generator)* comentado no capitulo 3, foi a aplicação escolhida para monitorar o acesso a Internet. O mesmo, após sua instalação, teve seu uso automatizado alterado, editando-se o arquivo `/etc/crontab` para que o mesmo seja executado todos os dias às 17 horas e 55 minutos. Sua modificação deu-se por motivo das atividades inerentes ao uso da Internet estarem encerradas a partir desse horário, ficando a cargo da aplicação gerar os relatórios de acessos ocorridos. A Figura 4.15 mostra a saida do arquivo crontab e sua alteração. Na

```
# Porta disponivel para o squid
http_port 3128

# Nome host squid
visible_hostname squalidus

# Linguagem das Páginas de Erro - Padrão Português Brasil
error_directory /usr/share/squid/errors/Portuguese/

# Tamanho do cache em Memória - Utilizado ate 1/3 de memória ram
cache_mem 350 MB

# Tamanho para armazenamento de cache armazenado na memória ram
maximum_object_size_in_memory 100 KB

# Tamanho maximo para armazenamento download em cache
maximum_object_size  512 MB

# Tamanho minimo para armazenamento download em cache
minimum_object_size 0 KB
```

Figura 4.14: Parâmetros de configuração do SQUID

Figura 4.16 tem-se a saida do arquivo `html` com relatorio de acesso gerado pelo SARG.

4.9 SSH e módulo pam_abl

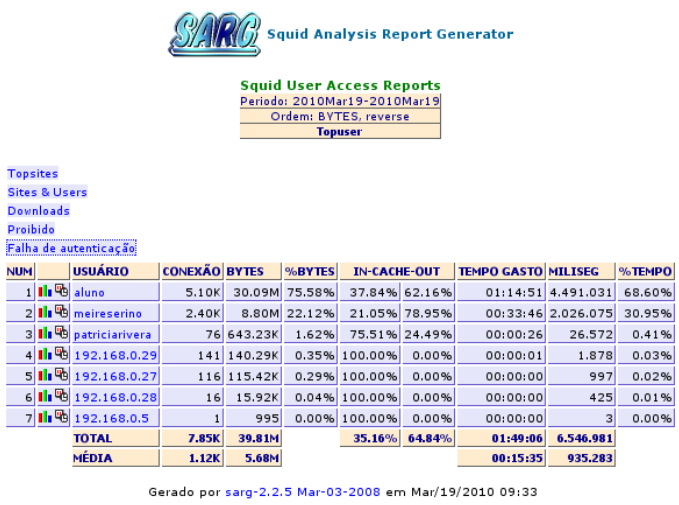
Para utilização do serviço `ssh`, partiu-se do pressuposto que o servidor em questão é acessado somente a partir da rede local. Para tanto, após a instalação do módulo servidor ou seja *openssh_server,* editou-se o arquivo de configuração do servidor `ssh` para ajustes de algumas regras, principalmente no tocante a porta padrão utilizada pelo mesmo serviço. Uma vez utilizado porta alta para acesso através do *ssh,* se reduz a exposição do servidor aumentando sua segurança de acesso.

Conforme (MORIMOTO, 2008) explica *"muitos dos ataques casuais (quando o atacante simplesmente varre faixas inteiras de endereços, sem um alvo definido), começam com um portscan genérico, onde é feita uma varredura em faixas inteiras de endereços IP, procurando por servidores com portas conhecidas abertas, como a 21, a 22 e a 80. Isso torna o teste mais rápido, permitindo localizar rapidamente*

```
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the 'crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 *  * * * root    cd / && run-parts --report /etc/cron.hourly
55 17 * * * root  test -x /usr/sbin/anacron || ( cd / && run-parts
  --report /etc/cron.daily )
47 6  * * 7 root  test -x /usr/sbin/anacron || ( cd / && run-parts
  --report /etc/cron.weekly )
52 6  1 * * root  test -x /usr/sbin/anacron || ( cd / && run-parts
  --report /etc/cron.monthly )
#
```

Figura 4.15: Configuração do horario no crontab



Squid Analysis Report Generator

Squid User Access Reports

Periodo: 2010Mar19-2010Mar19

Ordem: BYTES, reverse

Topuser

Topsites
Sites & Users
Downloads
Proibido
Falha de autenticação

NUM		USUÁRIO	CONEXÃO	BYTES	%BYTES	IN-CACHE-OUT		TEMPO GASTO	MILISEG	%TEMPO
1		aluno	5.10K	30.09M	75.58%	37.84%	62.16%	01:14:51	4.491.031	68.60%
2		meireserino	2.40K	8.80M	22.12%	21.05%	78.95%	00:33:46	2.026.075	30.95%
3		patriciarivera	76	643.23K	1.62%	75.51%	24.49%	00:00:26	26.572	0.41%
4		192.168.0.29	141	140.29K	0.35%	100.00%	0.00%	00:00:01	1.878	0.03%
5		192.168.0.27	116	115.42K	0.29%	100.00%	0.00%	00:00:00	997	0.02%
6		192.168.0.28	16	15.92K	0.04%	100.00%	0.00%	00:00:00	425	0.01%
7		192.168.0.5	1	995	0.00%	100.00%	0.00%	00:00:00	3	0.00%
		TOTAL	7.85K	39.81M		35.16%	64.84%	01:49:06	6.546.981	
		MÉDIA	1.12K	5.68M				00:15:35	935.283	

Gerado por sarg-2.2.5 Mar-03-2008 em Mar/19/2010 09:33

Figura 4.16: Relatorio de acesso gerado pelo SARG

um grande volume de hosts com portas abertas." Para o acesso somente a partir da rede local, o mesmo fora configurado com endereço IP proprio da rede local, evitando assim a disponibilização do serviço na Internet.

O protocolo permitido para acesso é somente o protocolo 2, uma vez que há problemas de segurança para o protocolo 1. Desativou-se também a opção onde se permite o *login* como *root,* pois motivos de segurança impedem esse acesso. Por padrão, o autor durante a instalação do sistema operacional, já havia desabilitado o *login* como *root,* onde o acesso ao sistema se da por *login* normal e após, através do uso do sudo obtem-se as permissões de *root* para o sistema. Outro ponto destacado na configuração do *daemon* `ssh` é com relação a permissão de acesso a usuários, onde apenas um único usuário terá o acesso disponibilizado.

Para essa questão, editou-se o comando *AllowUsers* aplicando-se a ele o usuário que irá ter acesso ao sistema. Como primeiro *login,* habilitou-se também o parâmetro *`Banner,`* para ofertar uma mensagem de aviso quando logado ao sistema, onde o mesmo tem um arquivo em separado localizado em `/etc/issue.net,` para editar-se mensagens de avisos. A Figura 4.17, mostra os parâmetros modificados no arquivo `/etc/ssh/sshd_config.`

```
# Package generated configuration file
# See the sshd(8) manpage for details

# What ports, IPs and protocols we listen for
Port 59327
# Use these options to restrict which interfaces/protocols
# sshd will bind to
ListenAddress 192.168.0.254
Protocol 2

# Authentication:
PermitRootLogin no
AllowUsers suporte

Banner /etc/issue.net
```

Figura 4.17: Alteração de parâmetros do sshd

A Figura 4.18, mostra a tela *login* do ssh, bem como a mensagem inicial de acesso. Após a conclusão da configuração, restartou-se o serviço ssh através do comando `/etc/init.d/ssh` *restart.* Além do ssh ora otimizado, uma outra aplicação será utilizada para acesso ao servidor em questão, o `scp`. Bastante primitiva, permite a especificação em uma única linha o login e o endereço do servidor bem como, o arquivo a ser transferido. A mesma, será de grande valia para o projeto uma vez que o servidor em questão não disponibiliza o uso do servidor Apache devido o seu uso ser bastante esporádico. O uso do comando scp para esse caso,

vem acompanhado do parâmetro -r onde copia-se o diretório todo para um host especificado para posterior análise.

```
stratos:/home/cesar# ssh -p 59327 suporte@192.168.0.254

Acesso restrito somente ao Administrador do Sistema...


Conteudo  PROTEGIDO   e   MONITORADO !!!


Obrigado!!!!


suporte@192.168.0.254's password:
```

Figura 4.18: Tela de login com mensagem

Para ofertar maior segurança ao acesso ssh, utilizou-se do módulo `pam_abl`. Para uso com a distribuição Debian, houve a necessidade de instalação das dependências `libpam-dev e libdb-dev`. Sua versão para compilação é ***pam_abl-0.2.3,*** onde após o *download* e sua respectiva compilação, copiou-se o arquivo `pam_abl.conf` para o diretório `/etc/security`.

Após editou-se o arquivo ***/etc/pam.d/common-auth*** aplicando os parâmetros `auth required, auth suficient` juntamente com seu diretorio de acesso, conforme Figura 4.19.

```
squalidus:/home/suporte# cat /etc/pam.d/common-auth
auth    required        pam_unix.so nullok_secure
auth    required        /lib/security/pam_env.so
auth    required        /lib/security/pam_abl.so
 config=/etc/security/pam_abl.conf
auth    sufficient      /lib/security/pam_unix.so likeauth nullok
auth    required        /lib/security/pam_deny.so
```

Figura 4.19: Arquivo de configuração pam.d/common-auth

Para finalizar sua configuração, editou-se o arquivo ***/etc/security/pam_abl.conf*** alterando-se o parâmetro `host_purge` para 3 dias para banimento de *hosts* e usuários e também o parâmetro `host_rule`, com banimento para 3 tentativas de acesso em um período de 1 hora e 10 tentativas de acesso no periodo de 1 dia. A regra em questão conforme a Figura 4.20, ilustra a configuração adotada para o *script* `pam_abl.conf`.

```
squalidus:/home/suporte# nano /etc/security/pam_abl.conf
host_db=/var/lib/abl/hosts.db
host_purge=3d
host_rule=*:3/1h,10/1d
user_db=/var/lib/abl/users.db
user_purge=3d
user_rule=!suporte:3/1,10/1d
```

Figura 4.20: Arquivo de configuração da biblioteca pam

4.10 Monitoramento de tráfego das interfaces

Para monitoramento do tráfego das interfaces de rede, o autor utilizou-se da aplicação *MRTG (Multi Router Traffic Grapher),* onde o mesmo fora otimizado sem o uso protocolo *SNMP.* A decisão pelo não uso do mesmo, deu-se por ser um protocolo simples de gerenciamento de redes onde utiliza o UDP *(unit datagram protocol)*[13] para transporte de pacotes de dados.

Para coleta de dados através do MRTG, editou-se o arquivo ***/etc/mrtg/mrtg.cfg***, informando-se o diretório `análise` como local para armazenamento dos dados o `WorkDir` e o `Htmldir`, uma vez que o próprio servidor não utiliza-se do Apache para visualização dos arquivos gerados. Para o tempo de carregamento da página, usou-se o parâmetro Refresh com valor de 300 segundos e o interval para atualização do tempo de chamada do MRTG para 5 minutos.

Estipulou-se também o parâmetro *Language* para *portuguese* e o script otimizado para ser executado como *daemon,* afim de otimizá-lo para coleta de dados a cada 5 minutos. No *script* em questão, informou-se as interfaces de rede a serem monitoradas, bem como o tamanho máximo para *maxbytes* para cada interface.

Com o intuito de prevenção de possíveis paradas do servidor em questão e por conseguinte a interrupção do serviço de coleta de dados das interfaces, fora necessário a criação de um *script* instalado na sau reinicialização do sistema operacional, a fim de que o mesmo continuasse sua coleta depois do mesmo ser reiniciado. A coleta dos dados referente ao monitoramento com aplicação MRTG teve seu início em 08 de Março de 2010 e seu término em 8 de Abril de 2010. Para tanto, na Figura 4.21 tem-se o *script* criado, bem como seu numero na fila de inicialização do arquivo *runlevel.conf.*

[13] *UDP - (Unit Datagram Protocol)* protocolo simples da camada de transporte descrito na RFC 768 o qual permite a uma aplicação escrever um datagrama encapsulado num pacote IPv4 ou IPv6 e, quando enviado ao destino não oferece garantia alguma de chegada do pacote.

```
squalidus:/etc# nano /etc/init.d/scrpt_mrtg.sh
#!/bin/bash

# Script para daemon mrtg

env LANG=C /usr/bin/mrtg /etc/mrtg/mrtg.cfg



squalidus:/etc# /etc/init.d/scrp_mrtg.sh
Daemonizing MRTG ...
squalidus:/etc#

# Entrada no runlevel.conf
30      -       2               /etc/init.d/scrpt_mrtg.cfg
```

Figura 4.21: Script de inicialização do MRTG

4.10.1 Monitoramento interface eth0 IP 10.1.1.2

Em conformidade com o servidor do projeto proposto, na analise do monitoramento para a interface 10.1.1.2, observa-se que o fluxo máximo de entrada de pacotes em *bits* por segundo, fora de 1222.1 kb/s, perfazendo um percentual de 12,2% (pontos percentuais). Em contrapartida obteve-se 327,4 kb/s perfazendo um total de 3,3% (pontos percentuais) do tráfego de saida notando-se em média aproximadamente 25% (pontos percentuais) de tráfego de saída em relação ao tráfego entrante.

Em porcentagem média relacionada ao tráfego entrante e sainte, obteve-se 0,1% (pontos percentuais) do tráfego entrante correspondente a 9592,0 b/s e, 2880.0 b/s de tráfego sainte. A Figura 4.22 mostra o relacionamento do tráfego entrante e sainte.

4.10.2 Monitoramento interface eth1 IP 192.168.0.254

Analisando-se o monitoramento da interface eth1, conclui-se que a mesma comporta-se inversamente proporcional a interface eth0 onde tem-se maior saida de pacotes de dados em relação a pacotes entrantes. Tem-se como fluxo máximo de entrada de pacotes apenas 3,2% (pontos percentuais) perfazendo um total de 321,5 kb/s.

Para o tráfego sainte tem-se 12,2% (pontos percentuais) perfazendo um total de 1222,0 kb/s. Oteve-se ainda com o monitoramento do fluxo de pacotes, a média

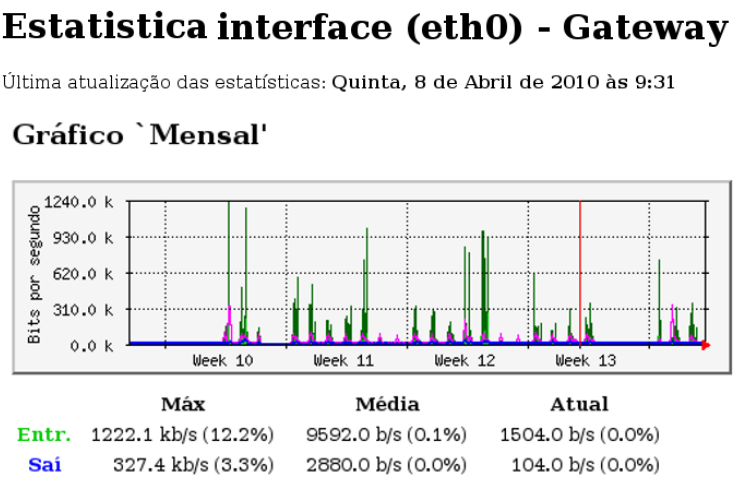


Figura 4.22: Monitoramento interface eth0 - IP 10.1.1.2

de saida de 13,4 kb/s totalizando 0,1% (ponto percentual). Em contrapartida, o mesmo apresentou como entrada média de pacotes o valor de 3432,0 b/s. A figura 4.23 disponibiliza o gráfico relativo ao monitoramento da interface eth1. Notoriamente houve um aumento considerável em relação ao acesso a interface eth1 com IP 192.168.0.254, pois a mesma está otimizada como *gateway* para a rede local.

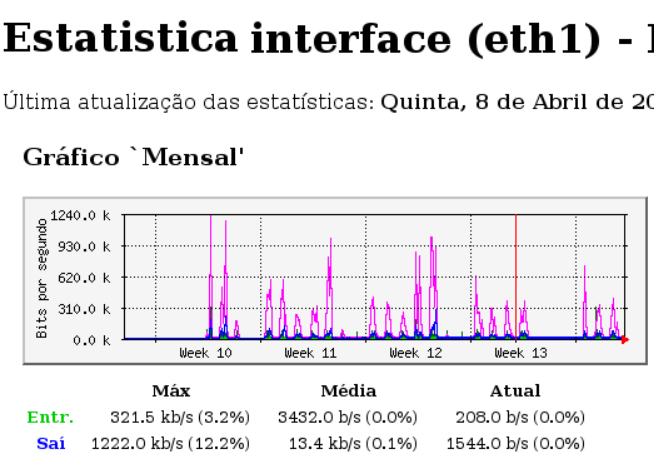


Figura 4.23: Monitoramento interface eth1 - IP 192.168.0.254

4.10.3 Monitoramento interface eth2 IP 192.168.3.254

Devido a dependência do corpo docente e administrativo para gerar o monitoramento da interface eth2, uma vez que fora solicitado os dispositivos móveis para cadastramento e não foram apresentados em sua totalidade bem como o uso não frequente, o monitoramento da interface em questão não gerou tráfego de rede considerável.

Teve-se como entrada máxima 12,1 kb/s perfazendo um total de 0,1% (ponto percentual). Em sua saída obteve-se apenas 165,3 kb/s onde totalizou apenas 1,7% (pontos percentuais). A nível médio de entrada de pacotes o resultado apresentado fora de 96,0 b/s e de saída fora de 1064,0 b/s. A Figura 4.24 mostra os resultados obtidos com o monitoramento dessa interface.

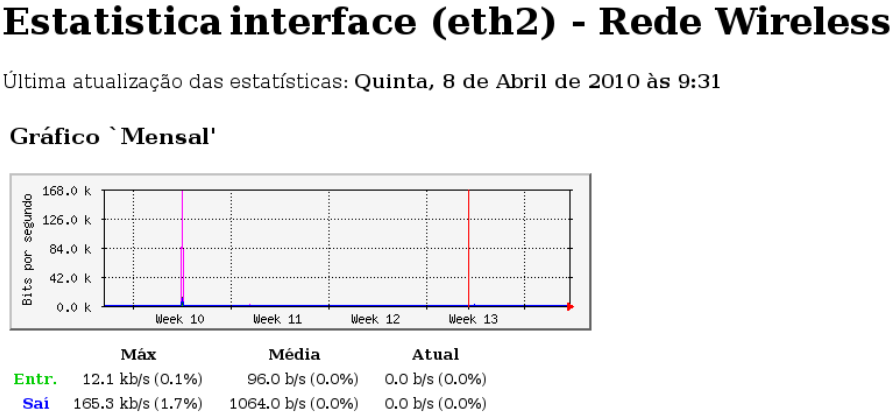


Figura 4.24: Monitoramento interface eth2 - IP 192.168.3.254

4.11 *Snort* - Otimizado para 3 interfaces

Para o sistema de detecção de intrusão, o *Snort* fora otimizado para monitorar as 3 interfaces de rede do servidor em questão, responsáveis respectivamente para conexão da rede local, rede *wireless* e *gateway* para Internet.

Após instalação do *Snort,* editou-se o arquivo de configuração ***/etc/snort/snort.conf***, alterando-se os parâmetros `var HOME_NET`, atribuindo-se a esse os endereços de *host* - no caso o servidor em questão, as redes locais e *wireless.* Para a mesma configuração, houve necessidade do parâmetro `var EXTERNAL_NET`, onde

está otimizada para monitorar todos os endereços IPs à excessão das redes local, *wireless* e o *gateway* para Internet. A Figura 4.25 ilustra os parâmetros otimizados no arquivo `/etc/snort/snort.conf`.

```
#--------------------------------------------------
#   http://www.snort.org     Snort 2.7.0 Ruleset
#     Contact: snort-sigs@lists.sourceforge.net
#--------------------------------------------------
# $Id$
# Redes de aplicação...
var HOME_NET [10.1.1.2/8,192.168.0.0/24,192.168.3.0/24]
var EXTERNAL_NET !$HOME_NET

# Lista de servidores
# List of DNS servers on your network
var DNS_SERVERS $HOME_NET

# List of SMTP servers on your network
var SMTP_SERVERS $HOME_NET

# List of web servers on your network
var HTTP_SERVERS $HOME_NET

# List of sql servers on your network
var SQL_SERVERS $HOME_NET

# List of telnet servers on your network
var TELNET_SERVERS $HOME_NET

# List of snmp servers on your network
var SNMP_SERVERS $HOME_NET
```

Figura 4.25: Parâmetros de configuração das redes local, *wireless* e *gateway* Internet

Após a configuração dos parâmetros onde o *Snort* irá monitorar, habilitou-se as regras utilizadas pelo mesmo. As mesmas fazem parte do arquivo de configuração do *Snort* e para tanto, estão habilitadas para que possam desempenhar suas funções dentro da otimização do *Snort.* São regras relacionadas a uso não corrente, relacionadas a politicas entre outras. A Figura 4.26 mostra o conjunto de regras habilitadas.

Após a otimização das regras do *Snort,* startou-se o serviço *Snort* a partir do comando `/etc/init.d/snort start`. No intuito de se fazer uma checagem em

```
# This ruleset is almost useless currently:
 include $RULE_PATH/virus.rules
# Note: this rule is extremely chatty, enable with care
 include $RULE_PATH/shellcode.rules

# Policy related rules:
 include $RULE_PATH/policy.rules
 include $RULE_PATH/community-policy.rules
 include $RULE_PATH/porn.rules
 include $RULE_PATH/community-inappropriate.rules
 include $RULE_PATH/chat.rules
 include $RULE_PATH/multimedia.rules
 include $RULE_PATH/p2p.rules
 include $RULE_PATH/community-game.rules
 include $RULE_PATH/community-misc.rules

# Extremely chatty rules:
 include $RULE_PATH/info.rules
 include $RULE_PATH/icmp-info.rules
 include $RULE_PATH/community-icmp.rules
```

Figura 4.26: Configuração da política de regras

sua otimização, aplicou-se o comando utilizado para verificar seu correto funcionamento, não gerando qualquer arquivo de alerta. A Figura 4.27 demonstra o resultado obtido com a saída do comando `snort -i -T -N -u snort -g snort` onde tem-se como parâmetros:

- `-i` relacionado a interface de rede.
- `-T` iniciado em modo auto-teste, checando linhas de comando e arquivos de regras.
- `-N` não gera packet logging. O programa continua gerando alertas.
- `-u` parâmetro relacionado ao usuário.
- `-g` parâmetro relacionado ao grupo.

```
squalidus:/home/suporte# snort -i eth0 -T -N -u snort -g snort
Running in packet logging mode
Log directory = /var/log/snort

        --== Initializing Snort ==--
Initializing Output Plugins!
Var 'eth1_ADDRESS' defined, value len = 25 chars,
   value = 192.168.0.0/255.255.255.0
Var 'eth2_ADDRESS' defined, value len = 25 chars,
   value = 192.168.3.0/255.255.255.0
Var 'any_ADDRESS' defined, value len = 15 chars,
   value = 0.0.0.0/0.0.0.0
Var 'lo_ADDRESS' defined, value len = 19 chars,
   value = 127.0.0.0/255.0.0.0
Verifying Preprocessor Configurations!
Decoding LoopBack on interface eth0
Preprocessor/Decoder Rule Count: 0

        --== Initialization Complete ==--

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.7.0 (Build 35)
   ''''    By Martin Roesch & The Snort Team:
      http://www.snort.org/team.html
           (C) Copyright 1998-2007 Sourcefire Inc., et al.


Snort sucessfully loaded all rules and checked all rule chains!
Snort exiting
```

Figura 4.27: Teste de inicialização do Snort

4.12 Uso de aplicação *HDIS - Tripwire*

Após otimizar o servidor com as aplicações inerentes à segurança do perímetro compreendido entre a Internet, rede local e rede *wireless,* no mesmo fora otimizado aplicação para proteção da integridade dos diretorios e arquivos de configuração.

A aplicação utilizada fora o **Tripwire**, onde em uma primeira instancia, durante sua instalação a mesma solicita passphrase o qual será utilizada quando se solicitar relatórios ao *Tripwire.* A Figura 4.28 mostra a instalação do mesmo.

Após sua otimização, aplicou-se o comando `tripwire -init` com intuito de gerar sua base de dados, aplicando-se para isso a senha de acesso a aplicação.

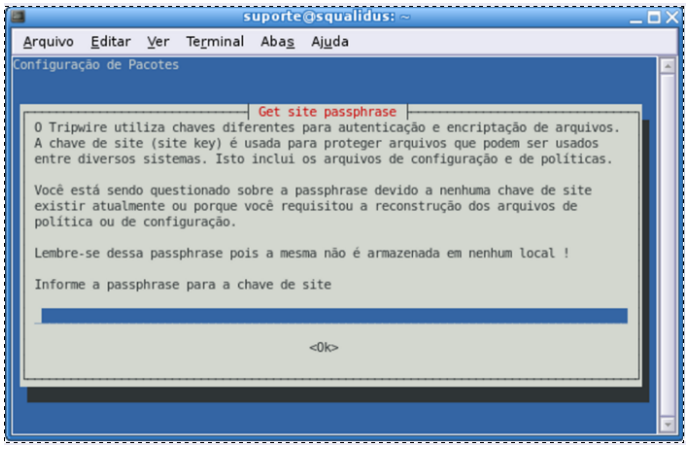


Figura 4.28: Tela para passphrase gerada no tripwire

Após sua otimização, aplicou-se ao servidor em questão, uma checagem afim de se verificar a integridade dos arquivos e diretórios. Para essa checagem, o comando utilizado fora `tripwire - -check`, o qual relatou ausência de possível intrusão real, onde houve apenas modificações decorrentes de ações por parte do administrador do sistema. A Figura 4.29 mostra parte do relatório gerado com a checagem do servidor.

```
--------------------------------------------------------------------------------
  Section: Unix File System
--------------------------------------------------------------------------------

  Rule Name                       Severity Level    Added    Removed  Modified
  ---------                       --------------    -----    -------  --------
  Invariant Directories           66                0        0        0
* Tripwire Data Files             100               1        0        0
  Other binaries                  66                0        0        0
  Tripwire Binaries               100               0        0        0
  Other libraries                 66                0        0        0
  Root file-system executables    100               0        0        0
  System boot changes             100               0        0        0
  Root file-system libraries      100               0        0        0
  (/lib)
  Critical system boot files      100               0        0        0
* Other configuration files       66                0        0        1
  (/etc)
  Boot Scripts                    100               0        0        0
  Security Control                66                0        0        0
  Root config files               100               0        0        0
* Devices & Kernel information    100               160      156      0

Total objects scanned:  20667
Total violations found:  318
```

Figura 4.29: Checagem da integridade do sistema - Tripwire

Após a checagem do servidor em questão, atualizou-se novamente a base de dados do *Tripwire* e procedeu-se um backup de todo o diretório Tripwire, a fim do

mesmo ser usado em possíveis analises de integridade dos arquivos e diretorios do servidor em questão.

4.13 Verificando estado das portas - *NMAP*

Com o intuito de checar o estado das portas do servidor, a aplicação utilizada foi o *NMAP*. Para esse escaneamento, levou-se em consideração o uso da aplicação *SNORT* estar otimizada no servidor em questão. Para tanto, utilizou-se do parâmetro `-sS` para escaneamento do mesmo, sem a necessidade de gerar um "falso positivo"para o *Snort*.

Conforme (MORIMOTO, 2008), *"operando neste modo, o NMAP apenas envia um pacote SYN para cada porta alvo e espera para ver se recebe um pacote ACK de confirmação sem, entretanto, responder com o segundo pacote ACK, que abriria a conexão. Isso permite burlar muitos programas de detecção de intrusão, que monitoram e logam apenas conexões efetivamente estabelecidas."*

Os devidos escaneamentos, foram executados a partir de uma estação de trabalho na rede local, diretamente a interface de rede interna, cujo ip é 192.168.0.254, onde apurou-se 2 serviços em execução durante o escaneamento. O mesmo escaneamento, fora realizado no *gateway* para Internet cujo ip é 10.1.1.2 e teve como resultado obtido nenhuma porta em atividade no momento do escaneamento. O escaneamento também fora realizado para interface da rede *wireless* cujo ip é 192.168.3.254 e teve como resultado obtido nenhuma porta aberta no momento do escaneamento. A Figura 4.30 mostra o escaneamento efetuado no servidor em questão.

```
Starting Nmap 4.62 ( http://nmap.org ) at 2010-04-01 12:31 BRT
Interesting ports on 192.168.0.254:
Not shown: 1713 filtered ports
PORT     STATE SERVICE    VERSION
53/tcp   open  domain
3128/tcp open  http-proxy Squid webproxy 2.7.STABLE3
MAC Address: 00:00:89:1D:59:42 (Cayman Systems)

Service detection performed. Please report any incorrect results
 at http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 33.023 seconds
```

Figura 4.30: Escaneamento portas interface local

4.14 Configuração do roteador *wireless*

Antecipando-se a configuração do mesmo, tratou-se da segurança física do roteador em questão, onde se teve como ponto estratégico, sua alocação mais proximo possível do centro da unidade escolar, evitando-se assim possível propagação do sinal para vizinhos e com isso obtendo-se nível do sinal excelente para todas as dependencias da unidade escolar. A Figura 4.31 mostra o *croqui* da alocação do *acess point* da unidade escolar.

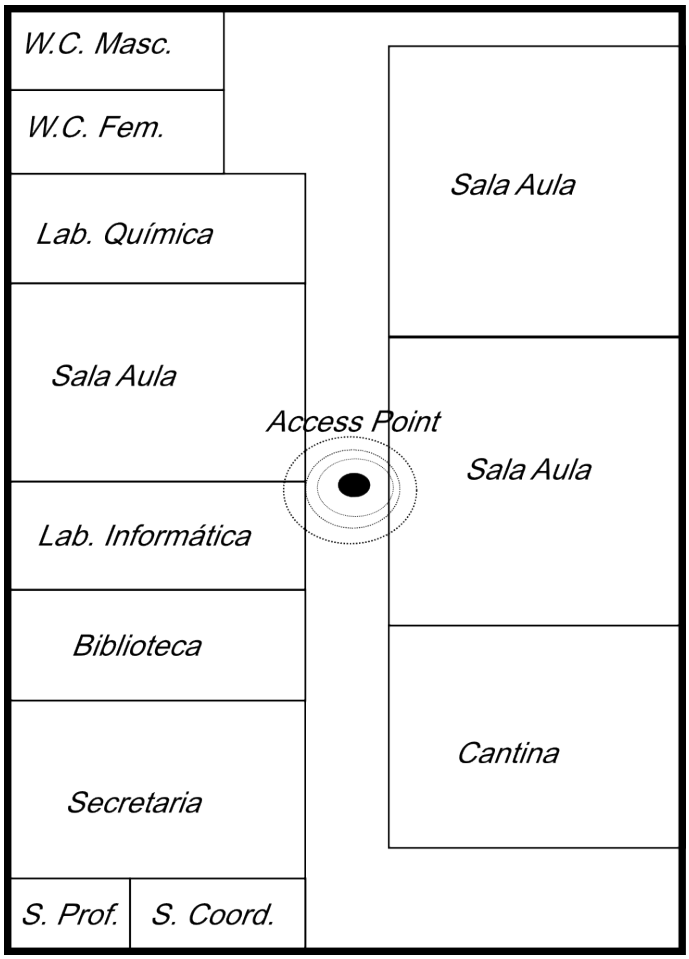


Figura 4.31: Alocação acess point - Wireless

Uma vez instalado em ponto estratégico, configurou-se o mesmo para disponibilizar sinal *wireless* para a unidade escolar. Em primeira instância, teve sua

reconfiguração iniciada a partir da modificação de seu endereço IP ora atribuido em conformidade com o projeto em questão, bem como a alteração de usuário e senha responsável pela administração do mesmo. Após as configurações iniciais, configurou-se os seguintes parâmetros:

- Region - No caso selecionou-se a região para Brazil
- Channel - Selecionou-se o canal 13, para evitar interferências provocadas por aparelhos como telefones sem-fio
- Mode - Modo alterado para 108 Mbps.
- Habilitou-se o `Enable Wireless Router Radio` e também o `Enable SSID Broadcast`.

Relacionando-se ainda parâmetros de segurança para o roteador em questão, selecionou-se como tipo de segurança o `WPA-PSK/WPA2-PSK`, onde teve-se como opção escolhida o `WPA2-PSK` usando como encriptação `AES`. Fora cadastrado também uma `PSK Passphrase` utilizando 15 caracteres como chave incluindo-se numeros, letras e caracteres especiais. A Figura 4.32 mostra o sistema de configuração utilizado no roteador em questão

Acordando com o projeto em questão, os dispositivos móveis utilizados no recinto escolar, serão obrigatoriamente cadastrados no roteador *wireless.* Para tanto, editou-se a opção `MAC address filtering`, onde fora solicitado a todo o corpo docente e administrativo que por ventura têm interesse em estarem utilizando o sinal *wireless* que apresentassem seu dispositivo móvel (no caso *notebook*) para a devida coleta do endereço fisico e, consequentemente o cadastramento no roteador em questão.

Além do quesito solicitado para usufruir-se do sinal *wireless* disponibilizado pela unidade escolar, todos os usuários administrativos e corpo docente receberam senha de acesso ao roteador *wireless* e também foram instruídos com relação ao procedimento para conexão ao *proxy squid* do perímetro disponibilizando o *gateway* da rede *wireless*.

As configurações fisicas, como acesso ao roteador e MAC *Address* foram implantadas com o intuito de reforçar a segurança do perímetro em questão, impedindo assim possíveis bisbilhoteiros de tentarem usufruir do *gateway* Internet da unidade escolar. A Figura 4.33 mostra a tela de cadastramento do endereço `MAC address filtering`.

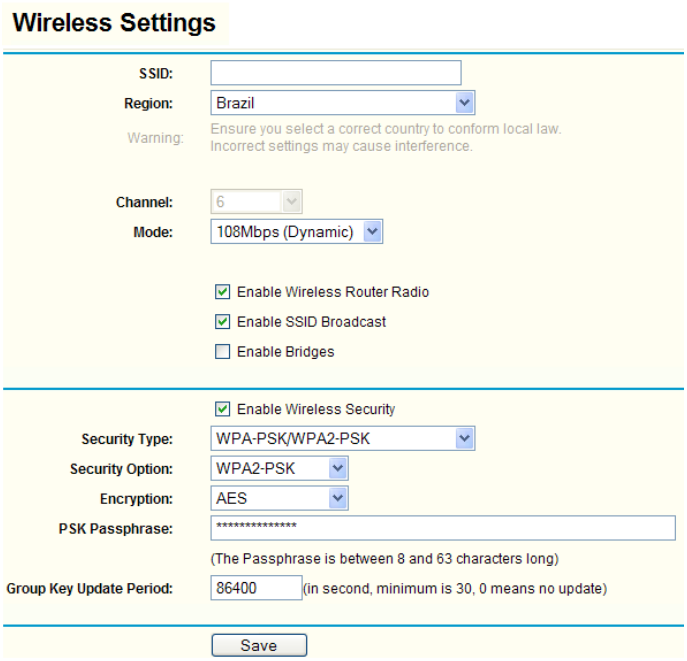


Figura 4.32: Configuração roteador wireless

Figura 4.33: Configuração MAC address filtering

Capítulo 5

Conclusão

Através desta monografia, pode-se concluir diversos fatores a serem levados em consideração para a devida implantação de um projeto de rede, seja ele de pequeno, médio ou grande porte. Como primeiro fator ponderante, focou-se o projeto do presente trabalho no perímetro compreendido entre a rede local, rede *wireless* e a Internet.

Para tanto, ferramentas de proteção para a rede local em plataforma proprietária foram substituídas em sua totalidade por ferramentas livres otimizadas em um servidor com sistema operacional **Gnu/Linux**. O mesmo fora implantado em ponto estratégico ou seja, como servidor de borda responsavel pelo controle do perímetro local e a Internet, onde após sua implantação possibilitou-se o acesso a Internet de forma segura e independente, necessitando somente de sua operacionalidade.

Tratando-se de segurança e por mais rígida que possa ser, pode haver falhas de administração ou vulnerabilidades em *software.* A principio, a segurança fora implementada no proprio servidor em questão onde tem-se o relatório completo do comportamento dos diretórios e arquivos de configuração com algumas técnicas de *hardening* aplicadas ao mesmo, com o intuito de coibir possíveis acessos por pessoas indesejadas.

Com o foco na proteção do perimetro local e do servidor, otimizou-se um *firewall iptables* bastante restritivo e de forma transparente. Não dispensou-se também um serviço de *proxy* autenticado para as redes em questão, pois o mesmo é de fundamental importância para garantir a integridade de qualquer usuário que

venha a utilizar a Internet na unidade escolar. Aplicou-se também ferramenta para escaneamento de portas do servidor com resultados bastante satisfatorios.

A respeito da velocidade de acesso a Internet, pode-se dizer que as redes, local e *wireless,* usufruem de um sistema de cache onde torna o acesso a conteúdos da Internet bastante rápidos, chegando em média ate 25% (pontos percentuais) de ganho de velocidade. O presente projeto teve embasamento em técnicas com conhecimento comprobatório onde as aplicações ora utilizadas foram estudadas para o uso, podendo em quaisquer outras situações serem utilizadas com novas fontes de recursos, bem como novas formas de otimização.

Em relação ao projeto, conclui-se que sua implantação oferta inúmeras chances para ampliações e trabalhos futuros utilizando-se *software* livre. Trabalhos que podem iniciar-se na substituição do sistema operacional proprietario hoje, utilizado no servidor de arquivos e do sistema de gestão da unidade escolar bem como, culminar com a utilização de aplicações livres com intuito de ofertar maior segurança, melhor proteção e também controle rigoroso no acesso aos dados do servidor em questão por usuários da rede da unidade escolar. Vale-se lembrar as diferenças inerentes a projetos de implantação de sistemas de segurança com *software* livre, de acordo com o modelo de negócio, bastando apenas elaborar-se um estudo de caso com o intuito de selecionar sempre a melhor escolha.

Referências Bibliográficas

ARMSTRONG, A. *The Auto Blacklist Module*. [S.l.]. Disponível em: <http://www.hexten.net/assets/pam_abl_doc/index.html>.

BADDINI, F. C. *Windows 2000 Server: Implementação e Administração*. São Paulo: Editora Érica, 2003.

BALL, B.; PITTS, D.; et al. *Dominando Red Hat Linux 7*. [S.l.]: Editora Ciencia Moderna Ltda, 2002. ISBN 85-7393-170-1.

CHESWICK, W. R.; BELLOVIN, S. M.; RUBIN, A. D. *Firewalls e Segurança na Internet - Repelindo o Hacker Ardiloso*. 2.a edição. ed. Porto Alegre - RS: Bookman, 2005.

GOMES, C. L.; ARRUDA, F. M. J.; WATTER, L. H.; SZTOLTZ, L.; TEIXEIRA, R. S. *Guia do Servidor Conectiva Linux*. Conectiva S.A., 2001. ISBN 85-87118-38-2. Disponível em: <http://www.dimap.ufrn.br/~aguiar/Manuais-/Servidor/part-opcoes-especiais.html>.

GUIA FOCA GNU/LINUX. *Guia Foca GNU/Linux Capítulo 10 - Firewall iptables*. [S.l.]. Disponível em: <http://focalinux.cipsga.org.br/guia/avancado/ch-fw-iptables.htm>.

JARDIM, F. de M. *Treinamento Avançado em Redes Wireless*. [S.l.]: Digerati Books, 2007. ISBN 978-85-60480-21-0.

MORIMOTO, C. E. *Servidores Linux - Guia Prático*. [S.l.]: GDH Press e Sul Editores, 2008. ISBN 978-85-99593-13-4.

NEMETH, E.; SNYDER, G.; HEIN, T. R. *Manual Completo do Linux - Guia do Administrador - Segunda Edição*. Segunda edição. [S.l.]: Pearson Education, 2007.

NETO, U. *Dominando Linux Firewall Iptables*. [S.l.]: Editora Ciencia Moderna Ltda, 2004. ISBN 85-7393-320-8.

NORTHCUTT, S.; ZELTSER, L.; WINTERS, S.; KENT, K.; RITCHEY, R. W. *Inside Network Perimeter Security*. [S.l.]: Sams Publishing, 2005.

RNP - REDE NACIONAL DE ENSINO E PESQUISA. *Implementando o serviço NTP na sua rede local*. [S.l.], Agosto 2000. Disponível em: <http://www.rnp.br/_arquivo/cais/manual_ntp_v1b.pdf>.

RUFINO, N. M. de O. *Segurança em Redes Sem Fio*. 2.a edição. ed. São Paulo: Novatec Editora Ltda, 2007. ISBN 978-85-7522-132-7.

SHINDER, D. T. W.; SHINDER, D. L.; GRASDAL, M. *ISA Server 2000 - Building Firewalls for Windows 2000*. 800 Hingham Street Rockland, MA 02370: Syngress Publishing, Inc., 2001. ISBN 1-928994-29-6.

TAMBORIM, A. L. *SSH com Modulo PAM*. [S.l.], jun. 2007. Disponível em: <http://www.vivaolinux.com.br/artigo/Seguranca-no-SSH-via-plugins-da-PAM-/?pagina=3>.

TURNBULL, J. *Hardening Linux*. [S.l.: s.n.], 2005. ISBN 1-59059-444-4.

WWW.TRIPWIRE.COM. *www.tripwire.com*. [S.l.]. Disponível em: <http:/-/www.tripwire.com/products/enterprise/ost/>.

Apêndice A

Firewall iptables

```
#!/bin/bash

#################################################################
##     Script firewall - Padrão                                 ##
###    Desenvolvimento padrao System V                          ###
####   Padronização para 3 interfaces                          ####
###    Runlevel: 2                                              ###
##     By - Antonio Cesar Pazebão -> pazebao@gmail.com          ##
#################################################################
############### Arquivo de ajuda ################################
#################################################################
Msg_uso="
Olá seja bem vindo!!!

Para utilizar o script, é necessario que ele esteja incluido
no arquivo /etc/init.d e tenha permissões de root para
execução.

Para visualizar os parâmetros de funcionamento do script, p
or favor digite:

/etc/init.d/scrpt_fwl.sh

Obrigado !!!!
"
```

```
#################################################################
############## Controle de Versões ##############################
#################################################################
controle_versao="
Versao 1: Padrao inicial para 2 interfaces de rede sendo eth0
Gateway Internet e eth1 Rede Local...

Versao 2: Padrao para 3 interfaces de rede sendo eth0 Gateway
Internet, eth1 Rede Local e eth2 Rede Wireless...

11/03/2010 - Liberado acesso ftp
        Hospedagem de médias e faltas no site do colegio

"


#################################################################
############## Definindo Variaveis...          ##############
#################################################################
# Caminho...
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:
 /usr/bin

# Variavel iptables...
IPT="/sbin/iptables"


#################################################################
############## Redes e Interfaces...           ##############
#################################################################
###################### Gateway Internet #########################
# Acesso internet...
## Nomeando interface eth0
IF_EXT="eth0"

## Atribuindo IP da interface eth0
IP_IF_EXT="10.1.1.2"
```

```
###################### Rede Local ##########################
# Acesso local...
## Nomeando interface eth1
IF_INT="eth1"

## Atribuindo IP interface eth1
IP_IF_INT="192.168.0.254"

## Atribuindo faixa de rede eth1
RANGE_IF_INT="192.168.0.0/24"


###################### Rede Wireless ########################
# Acesso Wireless
## Nomeando interface eth2
IF_WRL="eth2"

## Atribuindo IP interface eth2
IP_IF_WRL="192.168.3.254"

## Atribuindo faixa de rede eth2
RANGE_IF_WRL="192.168.3.0/24"


############################################################
############## Redes e Interfaces...          ##############
############################################################
# Ativando politicas...
atv_pol()
{

# Carregando modulos para ftp...
/sbin/modprobe ip_conntrack
/sbin/modprobe ip_conntrack_ftp
/sbin/modprobe ip_nat_ftp

# Limpando as regras
```

```
$IPT -F
$IPT -F -t nat
$IPT -F -t mangle
$IPT -X
$IPT -X -t nat
$IPT -X -t mangle
$IPT -Z
$IPT -Z -t nat

# Definindo politica padrao
$IPT -P INPUT DROP
$IPT -P OUTPUT DROP
$IPT -P FORWARD DROP


#################################################################
############## Protegendo Kernel...            ##############
#################################################################
# Proteçao syn-flood...
sysctl -w net.ipv4.tcp_syncookies=1 > /dev/null 2>&1

# Proteção ipspoofing...
sysctl -w net.ipv4.conf.all.rp_filter=1 > /dev/null 2>&1

# Proteçao ping broadcast...
sysctl -w net.ipv4.icmp_echo_ignore_broadcasts=1 >
 /dev/null 2>1&

# Protecao frames invalidos...
sysctl -w net.ipv4.icmp_ignore_bogus_error_responses=1 >
 /dev/null 2>1&

# Protecao contra timestamps
sysctl -w net.ipv4.tcp_timestamps=1 > /dev/null 2>1&


#################################################################
############## Checando estado da conexao...   ##############
#################################################################
```

```
# Estabelecendo estado conexao para INPUT
$IPT -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

# Estabelecendo estado conexao para OUTPUT
$IPT -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

# Estabelecendo estado conexão para FORWARD
$IPT -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT


############################################################
############## Regras de Input...              ##############
############################################################
# Permitindo Loopback...
$IPT -A INPUT -i lo -j ACCEPT

# Proibindo ping...
$IPT -A INPUT -p icmp --icmp-type echo-request -j DROP

# Protegendo conta port scanners ocultos
$IPT -A INPUT -p tcp --tcp-flags SYN,ACK,FIN,RST RST
 -m limit  --limit 1/s -j ACCEPT

# Descartando pacotes invalidos...
$IPT -A INPUT -m state --state INVALID -j DROP

# Liberando acesso ssh ( regra para configurar o router )
$IPT -A INPUT -p tcp -s 192.168.0.129 -d 192.168.0.254
 --dport 59327 -j ACCEPT

# Liberando acesso ntp - estações rede local -  udp apenas
$IPT -A INPUT -p udp -s $RANGE_IF_INT -d $IP_IF_INT
 --dport 123 -j ACCEPT

# Liberando acesso para dns - rede local
$IPT -A INPUT -p tcp -s $RANGE_IF_INT -d $IP_IF_INT
 --dport 53 -j ACCEPT
$IPT -A INPUT -p udp -s $RANGE_IF_INT -d $IP_IF_INT
 --dport 53 -j ACCEPT
```

```
# Liberando acesso para dns - rede wireless
$IPT -A INPUT -p tcp -s $RANGE_IF_WRL -d $IP_IF_WRL
 --dport 53 -j ACCEPT
$IPT -A INPUT -p udp -s $RANGE_IF_WRL -d $IP_IF_WRL
 --dport 53 -j ACCEPT

# Liberar input para squid - rede local
$IPT -A INPUT -p tcp -s $RANGE_IF_INT -i $IF_INT
 --dport 3128 -j ACCEPT

# Liberar input para squid - rede wireless
$IPT -A INPUT -p tcp -s $RANGE_IF_WRL -i $IF_WRL
 --dport 3128 -j ACCEPT


############################################################
############## Regras  de  Output...           ##############
############################################################
# Liberando loopback
$IPT -A OUTPUT -o lo -j ACCEPT

# Liberando ping
$IPT -A OUTPUT -p icmp --icmp-type echo-request -j ACCEPT

# Proteção contra port-scanners ocultos
$IPT -A OUTPUT -p tcp --tcp-flags SYN,ACK,FIN,RST RST
 -m limit --limit 1/s -j ACCEPT

# Liberando  consulta ao DNS
$IPT -A OUTPUT -p udp --dport 53 -j ACCEPT
$IPT -A OUTPUT -p tcp --dport 53 -j ACCEPT

# Liberando acesso para dns - rede local
$IPT -A OUTPUT -p tcp -d $RANGE_IF_INT -s $IP_IF_INT
 --dport 53 -j ACCEPT
$IPT -A OUTPUT -p udp -d $RANGE_IF_INT -s $IP_IF_INT
 --dport 53 -j ACCEPT
```

```
# Liberando acesso para dns - rede wireless
$IPT -A OUTPUT -p tcp -d $RANGE_IF_WRL -s $IP_IF_WRL
 --dport 53 -j ACCEPT
$IPT -A OUTPUT -p udp -d $RANGE_IF_WRL -s $IP_IF_WRL
 --dport 53 -j ACCEPT

# Liberando porta 80
$IPT -A OUTPUT -p tcp --dport 80 -j ACCEPT

# Liberando porta 443 - HTTPS
$IPT -A OUTPUT -p tcp --dport 443 -j ACCEPT

# Liberando porta 563 - SSL
$IPT -A OUTPUT -p tcp --dport 563 -j ACCEPT

# Liberando acesso ftp para o servidor
$IPT -A OUTPUT -p tcp -s 192.168.0.1 -d
 asp.colegioadv.com.br --dport 21 -j ACCEPT
$IPT -A OUTPUT -p tcp -s 192.168.0.1 -d
 asp.colegioadv.com.br --dport 20 -j ACCEPT

# Liberando porta 1863 - MSN
$IPT -A OUTPUT -p tcp --dport 1863 -j ACCEPT

# Liberando a.ntp.br - sincronizando relogio
$IPT -A OUTPUT -p udp -s $IP_IF_EXT -d 200.160.0.8
 --dport 123 -j ACCEPT

# Liberando b.ntp.br - sincronizando relogio
$IPT -A OUTPUT -p udp -s $IP_IF_EXT -d 200.189.40.8
 --dport 123 -j ACCEPT

# Liberando c.ntp.br - sincronizando relogio
$IPT -A OUTPUT -p udp -s $IP_IF_EXT -d 200.192.232.8
 --dport 123 -j ACCEPT


################################################################
###############     Regras de Forward...        ###############
```

```
#############################################################
# Bloqueando pacotes invalidos...
$IPT -A FORWARD -m state --state INVALID -j DROP

# Proteção contra port-scanners ocultos
$IPT -A FORWARD -p tcp --tcp-flags SYN,ACK,FIN,RST RST
-m limit --limit 1/s -j ACCEPT

# Bloqueando acesso da Rede Wireless para Rede Local
$IPT -A FORWARD -s $RANGE_IF_WRL -i $IF_WRL -d $RANGE_IF_INT
 -o $IF_INT -j DROP

# Bloqueando acesso da Rede local para Rede Wireless
$IPT -A FORWARD -s $RANGE_IF_INT -i $IF_INT -d $RANGE_IF_WRL
 -o $IF_WRL -j DROP

# Liberando acesso ftp - Destino Hospedaria.com.br (ISP)
$IPT -A FORWARD -p tcp -s 192.168.0.1 -d
 asp.colegioadv.com.br --dport 21 -j ACCEPT
$IPT -A FORWARD -p tcp -s 192.168.0.1 -d
 asp.colegioadv.com.br --dport 20 -j ACCEPT

# Libera acesso pop para area Administrativa Porta pop 110
$IPT -A FORWARD -p tcp -s 192.168.0.2 -d
 pop3.colegioadv.com.br --dport 110 -j ACCEPT
$IPT -A FORWARD -p tcp -s 192.168.0.4 -d
 pop3.colegioadv.com.br --dport 110 -j ACCEPT
$IPT -A FORWARD -p tcp -s 192.168.0.5 -d
 pop3.colegioadv.com.br --dport 110 -j ACCEPT
$IPT -A FORWARD -p tcp -s 192.168.0.6 -d
 pop3.colegioadv.com.br --dport 110 -j ACCEPT

# Libera acesso smtp para area Administrativa Porta smtp 587
$IPT -A FORWARD -p tcp -s 192.168.0.2 -d
 smtp.colegioadv.com.br --dport 587 -j ACCEPT
$IPT -A FORWARD -p tcp -s 192.168.0.4 -d
 smtp.colegioadv.com.br --dport 587 -j ACCEPT
$IPT -A FORWARD -p tcp -s 192.168.0.5 -d
 smtp.colegioadv.com.br --dport 587 -j ACCEPT
```

```
$IPT -A FORWARD -p tcp -s 192.168.0.6 -d
 smtp.colegioadv.com.br --dport 587 -j ACCEPT


############################################################
############## Regras para Nat...              ##############
############################################################
# Habilitando encaminhamento de pacotes no Kernel
sysctl -w net.ipv4.ip_forward=1 > /dev/null 2>&1

# Direcionando porta 80 para o squid na porta 3128
$IPT -t nat -A PREROUTING -i $IF_EXT -p tcp --dport 80 -j
 REDIRECT --to-port 3128

# Mascaramento para acesso  pop - Porta 110
$IPT -t nat -A POSTROUTING -p tcp -s 192.168.0.2 -d
 pop3.colegioadv.com.br --dport 110 -j MASQUERADE
$IPT -t nat -A POSTROUTING -p tcp -s 192.168.0.4 -d
 pop3.colegioadv.com.br --dport 110 -j MASQUERADE
$IPT -t nat -A POSTROUTING -p tcp -s 192.168.0.5 -d
 pop3.colegioadv.com.br --dport 110 -j MASQUERADE
$IPT -t nat -A POSTROUTING -p tcp -s 192.168.0.6 -d
 pop3.colegioadv.com.br --dport 110 -j MASQUERADE

# Mascaramento para acesso smtp - Porta 587
$IPT -t nat -A POSTROUTING -p tcp -s 192.168.0.2 -d
 smtp.colegioadv.com.br --dport 587 -j MASQUERADE
$IPT -t nat -A POSTROUTING -p tcp -s 192.168.0.4 -d
 smtp.colegioadv.com.br --dport 587 -j MASQUERADE
$IPT -t nat -A POSTROUTING -p tcp -s 192.168.0.5 -d
 smtp.colegioadv.com.br --dport 587 -j MASQUERADE
$IPT -t nat -A POSTROUTING -p tcp -s 192.168.0.6 -d
 smtp.colegioadv.com.br --dport 587 -j MASQUERADE

## Habilitando o roteamento de pacotes = PostRouting
$IPT -t nat -A POSTROUTING -s 192.168.0.1 -o $IF_EXT
 -j SNAT --to $IP_IF_EXT
$IPT -t nat -A POSTROUTING -s 192.168.0.1 -d
 asp.colegioadv.com.br -j SNAT --to $IP_IF_EXT
```

```
}

# Limpando politicas...
limp_polt()
{

# Definindo politica padrao - Modo Default
$IPT -P INPUT ACCEPT
$IPT -P OUTPUT ACCEPT
$IPT -P FORWARD ACCEPT

# Limpando as regras
$IPT -F
$IPT -F -t nat
$IPT -F -t mangle
$IPT -X
$IPT -X -t nat
$IPT -X -t mangle
$IPT -Z
$IPT -Z -t nat

}


################################################################
##############             MAIN...             ##############
################################################################
case "$1" in
start)
echo "Aplicando regras de firewall... "
atv_pol
exit 0
;;

stop)
echo "Limpando políticas..."
limp_polt
```

```
exit 0
;;

help)
if test "$1" = "help"
then
echo "$Msg_uso"
exit 0
fi
;;

versao)
# Extrai a versão diretamente do cabeçalho do programa
if test "$2" = "$versao"
then
echo "$controle_versao"
exit 0
fi
;;

*)
echo "Utilização: /etc/init.d/scrptf_flw.sh
 {start|stop|help|versao}"
exit 1
;;
esac
```

Apêndice B

Proxy Squid

```
# Configurações básicas para o Squid
# Colegio ADV - Unidade Jau SP.

# Porta disponivel para o squid
http_port 3128

# Nome host squid
visible_hostname squalidus

# Linguagem das Páginas de Erro - Padrão Português Brasil
error_directory /usr/share/squid/errors/Portuguese/

# Tamanho cache em Memória  utilizado ate 1/3 de memória ram
cache_mem 350 MB

# Tamanho de armazenamento  cache armazenado na memória ram
maximum_object_size_in_memory 100 KB

# Tamanho maximo para armazenamento download em cache
maximum_object_size  512 MB

# Tamanho minimo para armazenamento download em cache
minimum_object_size 0 KB

# Tamanho minimo para descarte de arquivos antigos
```

```
cache_swap_low 90

# Tamanho maximo para descarte de arquivos antigos
cache_swap_high 95

# Diretorio armazenamento cache
cache_dir ufs /var/spool/squid 10240 16 256

# Setando usuário
cache_effective_user proxy

# Setando grupo
cache_effective_group proxy

# Diretorio armazenamento log's de acesso
cache_access_log /var/log/squid/access.log

# Tempo máximo de verificação de atualizações
refresh_pattern ^ftp: 15 20% 2280
refresh_pattern ^gopher: 15 0% 2280
refresh_pattern . 15 20% 2280

acl all src 0.0.0.0/0.0.0.0
acl manager proto cache_object
acl localhost src 127.0.0.1/255.255.255.255
acl SSL_ports port 443 563
acl Safe_ports port 20 21 80 443 563 70 210 280 488 59 777
 901 1025-65535
acl purge method PURGE
acl CONNECT method CONNECT

http_access allow manager localhost
http_access deny manager
http_access allow purge localhost
http_access deny purge
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports

#############################################################
```

```
#########    REDES CONHECIDAS                          ########
#############################################################
# Rede Local
acl redelocal src 192.168.0.0/24

# Rede wireless
acl redewireless src 192.168.3.0/24

# Bloqueio acessos fora redelocal e redewireless
http_access deny !redelocal !redewireless

# Bloqueando acesso por url...
acl sites url_regex -i  "/etc/squid/denied/denied_url"
http_access deny sites

# Bloqueando acesso por palavras...
acl bloq_palavras dstdom_regex
 "/etc/squid/denied/denied_palavras
http_access deny bloq_palavras

# Bloqueando acesso por ip...
acl acesso_ip dst  "/etc/squid/denied/denied_ip"
http_access deny acesso_ip


# Liberando updates para antivirus...
acl libera_antivirus url_regex -i
 "/etc/squid/restrict/antivirus"
http_access allow libera_antivirus

# Liberando usuários restritos [ sem autenticação ]
acl libera_user src "/etc/squid/restrict/granted"
http_access allow libera_user

# Autenticando usuário...
auth_param basic realm Squid
auth_param basic program /usr/lib/squid/ncsa_auth
 /etc/squid/squid_passwd
acl aut_users proxy_auth REQUIRED
```

```
http_access allow aut_users

# Liberando downloads [ Usuarios restritos ]
acl libera_down src "/etc/squid/restrict/downloads"
http_access allow libera_down

# Bloqueando acesso por extensões
acl proibidos url_regex -i  "/etc/squid/denied/denied
 _extensoes"
http_access deny proibidos

# Liberando acesso redelocal, wireless  e local host
http_access allow localhost
http_access allow redelocal
http_access allow redewireless

# Proibindo demais
http_access deny all
```

Apêndice C

Detecção intrusão - SNORT

```
# $Id$
# Redes de aplicação...
 var HOME_NET [10.1.1.2/24,192.168.0.0/24,192.168.3.0/24]
 var EXTERNAL_NET !$HOME_NET

# Lista de servidores
var DNS_SERVERS $HOME_NET
var SMTP_SERVERS $HOME_NET
var HTTP_SERVERS $HOME_NET
var SQL_SERVERS $HOME_NET
var TELNET_SERVERS $HOME_NET
var SNMP_SERVERS $HOME_NET

# Portas
var HTTP_PORTS 80
var SHELLCODE_PORTS !80
var ORACLE_PORTS 1521

# Outras variaveis
var AIM_SERVERS [64.12.24.0/23,64.12.28.0/23,64.12.161.0/24,
 64.12.163.0/24,64.12.200.0/24,205.188.3.0/24,205.188.5.0/24,
 205.188.7.0/24,205.188.9.0/24,205.188.153.0/24,
 205.188.179.0/24,205.188.248.0/24]

var RULE_PATH /etc/snort/rules
```

```
# Configurar decoder snort
# Passo 2: Configurar dll's

dynamicpreprocessor directory
 /usr/lib/snort_dynamicpreprocessor/
dynamicengine
 /usr/lib/snort_dynamicengine/libsf_engine.so
##################################################################
# Passo 3: Configurar pre-processadores
preprocessor flow: stats_interval 0 hash 2
preprocessor frag3_global: max_frags 65536
preprocessor frag3_engine: policy first detect_anomalies

 reassembly for Snort
preprocessor stream5_global: max_tcp 8192, track_tcp yes, \
                             track_udp no
preprocessor stream5_tcp: policy first,
 use_static_footprint_sizes


# Estatistica de Performance
preprocessor http_inspect: global \
    iis_unicode_map unicode.map 1252

preprocessor http_inspect_server: server default \
  profile all ports { 80 8080 8180 } oversize_dir_length 500

preprocessor rpc_decode: 111 32771

preprocessor bo

preprocessor ftp_telnet: global \
   encrypted_traffic yes \
   inspection_type stateful

preprocessor ftp_telnet_protocol: telnet \
   normalize \
   ayt_attack_thresh 200
```

```
preprocessor ftp_telnet_protocol: ftp server default \
   def_max_param_len 100 \
   alt_max_param_len 200 { CWD } \
   cmd_validity MODE < char ASBCZ > \
   cmd_validity MDTM < [ date nnnnnnnnnnnnnn[.n[n[n]]] ]
     string > \
   chk_str_fmt { USER PASS RNFR RNTO SITE MKD } \
   telnet_cmds yes \
   data_chan

preprocessor ftp_telnet_protocol: ftp client default \
   max_resp_len 256 \
   bounce yes \
   telnet_cmds yes


preprocessor smtp: \
  ports { 25 } \
  inspection_type stateful \
  normalize cmds \
  normalize_cmds { EXPN VRFY RCPT } \
  alt_max_command_line_len 260 { MAIL } \
  alt_max_command_line_len 300 { RCPT } \
  alt_max_command_line_len 500 { HELP HELO ETRN } \
  alt_max_command_line_len 255 { EXPN VRFY }

preprocessor sfportscan: proto  { all } \
                         memcap { 10000000 } \
                         sense_level { low }


preprocessor dcerpc: \
    autodetect \
    max_frag_size 3000 \
    memcap 100000

preprocessor dns: \
    ports { 53 } \
```

```
    enable_rdata_overflow


# Passo 4: Configure saida de plugins
output log_tcpdump: tcpdump.log


include classification.config
include reference.config

#################################################################
# Passo 5: Configurar as regras

include $RULE_PATH/local.rules
include $RULE_PATH/bad-traffic.rules
include $RULE_PATH/exploit.rules
include $RULE_PATH/community-exploit.rules
include $RULE_PATH/scan.rules
include $RULE_PATH/finger.rules
include $RULE_PATH/ftp.rules
include $RULE_PATH/telnet.rules
include $RULE_PATH/rpc.rules
include $RULE_PATH/rservices.rules
include $RULE_PATH/dos.rules
include $RULE_PATH/community-dos.rules
include $RULE_PATH/ddos.rules
include $RULE_PATH/dns.rules
include $RULE_PATH/tftp.rules

# Especificas para servidores web:
include $RULE_PATH/web-cgi.rules
include $RULE_PATH/web-coldfusion.rules
include $RULE_PATH/web-iis.rules
include $RULE_PATH/web-frontpage.rules
include $RULE_PATH/web-misc.rules
include $RULE_PATH/web-client.rules
include $RULE_PATH/web-php.rules
include $RULE_PATH/community-sql-injection.rules
include $RULE_PATH/community-web-client.rules
```

```
include $RULE_PATH/community-web-dos.rules
include $RULE_PATH/community-web-iis.rules
include $RULE_PATH/community-web-misc.rules
include $RULE_PATH/community-web-php.rules

# Para outros serviços:
include $RULE_PATH/sql.rules
include $RULE_PATH/x11.rules
include $RULE_PATH/icmp.rules
include $RULE_PATH/netbios.rules
include $RULE_PATH/misc.rules
include $RULE_PATH/attack-responses.rules
include $RULE_PATH/oracle.rules
include $RULE_PATH/community-oracle.rules
include $RULE_PATH/mysql.rules
include $RULE_PATH/snmp.rules
include $RULE_PATH/community-ftp.rules
include $RULE_PATH/smtp.rules
include $RULE_PATH/community-smtp.rules
include $RULE_PATH/imap.rules
include $RULE_PATH/community-imap.rules
include $RULE_PATH/pop2.rules
include $RULE_PATH/pop3.rules

include $RULE_PATH/nntp.rules
include $RULE_PATH/community-nntp.rules
include $RULE_PATH/community-sip.rules
include $RULE_PATH/other-ids.rules

# Regra para ataques em andamento:
include $RULE_PATH/web-attacks.rules
include $RULE_PATH/backdoor.rules
include $RULE_PATH/community-bot.rules
include $RULE_PATH/community-virus.rules
# This ruleset is almost useless currently:
 include $RULE_PATH/virus.rules
# Note: this rule is extremely chatty, enable with care
 include $RULE_PATH/shellcode.rules
```

```
# Politicas relacionadas:
 include $RULE_PATH/policy.rules
 include $RULE_PATH/community-policy.rules
 include $RULE_PATH/porn.rules
 include $RULE_PATH/community-inappropriate.rules
 include $RULE_PATH/chat.rules
 include $RULE_PATH/multimedia.rules
 include $RULE_PATH/p2p.rules
 include $RULE_PATH/community-game.rules
 include $RULE_PATH/community-misc.rules

# Extremamente falantes:
 include $RULE_PATH/info.rules
 include $RULE_PATH/icmp-info.rules
 include $RULE_PATH/community-icmp.rules

# Regras experimentais:
include $RULE_PATH/experimental.rules

include threshold.conf
```

Apêndice D

Configuração MRTG

```
#Configurando MRTG
WorkDir: /etc/mrtg/analise
Htmldir: /etc/mrtg/analise
icondir: images/
Refresh: 300
Interval: 5
#Language: portuguese
Language: brazilian
RunAsDaemon:Yes
#---------------------
# Monitorar eth0
# Gateway Internet
#---------------------
Target[eth0]: `cat /proc/net/dev |grep eth0 |awk -F':'
 '{print $2}' |awk '{print $1}'; cat /proc/net/dev |grep eth0
  | awk -F':' '{print $2}' |awk '{print $9}'; echo -e; echo -e`
MaxBytes[eth0]: 1250000
Title[eth0]: Trafego - Gateway Internet
PageTop[eth0]: <H1>Estatistica da interface (eth0) -
 Gateway Internet</H1>
YLegend[eth0]: Bits por segundo
WithPeak[eth0]: Grafico Diario (5 minutos Media)
Options[eth0]: growright,bits

#---------------------
```

```
# Monitorar eth1
# REDE LOCAL
#---------------------
Target[eth1]: `cat /proc/net/dev |grep eth1 |awk -F':'
 '{print $2}' |awk '{print $1}'; cat /proc/net/dev |grep eth1
  | awk -F':' '{print $2}' |awk '{print $9}'; echo -e; echo -e`
MaxBytes[eth1]: 1250000
Title[eth1]: Trafego - Rede Local
PageTop[eth1]: <H1>Estatistica da interface (eth1) -
 Rede Local</H1>
YLegend[eth1]: Bits por segundo
WithPeak[eth1]: Grafico Diario (5 minutos Media)
Options[eth1]: growright,bits

#---------------------
# Monitorar eth2
# REDE WIRELESS
#---------------------
Target[eth2]: `cat /proc/net/dev |grep eth2 |awk -F':'
 '{print $2}' |awk '{print $1}'; cat /proc/net/dev |grep eth2
  | awk -F':' '{print $2}' |awk '{print $9}'; echo -e; echo -e`
MaxBytes[eth2]: 1250000
Title[eth2]: Trafego - Rede Wireless
PageTop[eth2]: <H1>Estatistica da interface (eth2) -
 Rede Wireless</H1>
YLegend[eth2]: Bits por segundo
WithPeak[eth2]: Grafico Diario (5 minutos Media)
Options[eth2]: growright,bits
```

Apêndice E

Configuração interfaces rede

```
# Interface loopback
auto lo
iface lo inet loopback

# Interface eth0
allow-hotplug eth0
iface eth0 inet static
        address 10.1.1.2
        netmask 255.255.255.0
        network 10.1.1.0
        broadcast 10.1.1.255
        gateway 10.1.1.1


# Interface eth1
allow-hotplug eth1
iface eth1 inet static
        address 192.168.0.254
        netmask 255.255.255.0
        network 192.168.0.0
        broadcast 192.168.0.255

# Interface eth2
allow-hotplug eth2
iface eth2 inet static
```

```
address 192.168.3.254
netmask 255.255.255.0
network 192.168.3.0
broadcast 192.168.3.255
```

Apêndice F

Configuração cache-dns - (named.conf.options)

```
options {
 directory "/var/cache/bind";

 forwarders {
 // Servidor DNS Root Master
    4.2.2.2;
 // Servidores DNS Fapesp
    208.67.222.222;
    200.160.0.10;
    };

 // Opções de segurança
 // Rede Local
 listen-on { 127.0.0.1; 192.168.0.254;  192.168.3.254;  };
 allow-query { 127.0.0.1; 192.168.0.0/24; 192.168.3.0/24; };
 allow-recursion { 127.0.0.1; 192.168.0.0/24; 192.168.3.0/24; };
 allow-transfer { none; };
 auth-nxdomain no;    # conform to RFC1035
 //      listen-on-v6 { any; };
};
```

Apêndice G

Configuração protocolo NTP

```
driftfile /etc/ntp.drift

statistics loopstats peerstats clockstats
filegen loopstats file loopstats type day enable
filegen peerstats file peerstats type day enable
filegen clockstats file clockstats type day enable

# Servidores publicos do projeto ntp.br
server a.ntp.br iburst
server b.ntp.br iburst
server c.ntp.br iburst


# Configuração das restriçoes de acesso...
restrict default kod notrap nomodify nopeer
```

Apêndice H

Arquivos de incialização - runlevel.conf

```
# Format:
# <sort> <off-> <on-levels>     <command>
01      -       S           /etc/init.d/glibc.sh
02      -       S           /etc/init.d/hostname.sh
02      -       S           /etc/init.d/mountkernfs.sh
03      -       S           /etc/init.d/udev
04      -       S           /etc/init.d/mountdevsubfs.sh
05      -       1           /etc/init.d/single
05      -       S           /etc/init.d/bootlogd
05      -       S           /etc/init.d/keymap.sh
08      -       S           /etc/init.d/hwclockfirst.sh
10      -       2           /etc/init.d/rsyslog
10      -       S           /etc/init.d/checkroot.sh
11      1       -           /etc/init.d/cron
11      -       S           /etc/init.d/hwclock.sh
12      -       2           /etc/init.d/acpid
12      -       S           /etc/init.d/mtab.sh
15      -       2           /etc/init.d/bind9
16      -       2           /etc/init.d/ssh
18      -       S           /etc/init.d/ifupdown-clean
20      -       2           /etc/init.d/scrpt_fwl.sh
20      0,1,6   2           /etc/init.d/openbsd-inetd
```

```
20      -       0,6         /etc/init.d/sendsigs
20      -       S           /etc/init.d/module-init-tools
20      0,1,6   2           /etc/init.d/snort
23      0,1,6   2           /etc/init.d/ntp
25      0,6     -           /etc/init.d/hwclock.sh
30      -       0,6         /etc/init.d/urandom
30      -       1           /etc/init.d/killprocs
30      -       S           /etc/init.d/checkfs.sh
30      -       S           /etc/init.d/procps
30      0,1,6   2           /etc/init.d/squid
30      -       2           /etc/init.d/scrpt_mrtg.cfg
30      -       2           /etc/init.d/load_mrtg.sh
31      -       0,6         /etc/init.d/umountnfs.sh
35      -       0,6         /etc/init.d/networking
35      -       S           /etc/init.d/mountall.sh
36      -       0,6         /etc/init.d/ifupdown
36      -       S           /etc/init.d/mountall-bootclean.sh
36      -       S           /etc/init.d/udev-mtab
37      -       S           /etc/init.d/mountoverflowtmp
39      -       S           /etc/init.d/ifupdown
40      -       0,6         /etc/init.d/umountfs
40      -       S           /etc/init.d/networking
40      1       2           /etc/init.d/dhcp3-server
45      -       S           /etc/init.d/mountnfs.sh
46      -       S           /etc/init.d/mountnfs-bootclean.sh
48      -       S           /etc/init.d/console-screen.sh
55      -       S           /etc/init.d/bootmisc.sh
55      -       S           /etc/init.d/urandom
60      -       0,6         /etc/init.d/umountroot
63      0,6     -           /etc/init.d/mountoverflowtmp
70      -       S           /etc/init.d/x11-common
75      -       S           /etc/init.d/sudo
84      1       -           /etc/init.d/ssh
85      0,1,6   -           /etc/init.d/bind9
88      1       -           /etc/init.d/acpid
89      -       2           /etc/init.d/atd
89      -       2           /etc/init.d/cron
90      0,1,6   -           /etc/init.d/rsyslog
90      -       0           /etc/init.d/halt
```

```
90      -       6           /etc/init.d/reboot
99      -       2           /etc/init.d/rc.local
99      -       2           /etc/init.d/rmnologin
99      -       2           /etc/init.d/stop-bootlogd
99      -       S           /etc/init.d/stop-bootlogd-single
```

Apêndice I

Configuração *sshd*

```
# Porta de escuta
Port 59327
Protocol 2
HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_dsa_key
UsePrivilegeSeparation yes
KeyRegenerationInterval 3600
ServerKeyBits 768
SyslogFacility AUTH
LogLevel INFO
LoginGraceTime 120
PermitRootLogin no
AllowUsers suporte
StrictModes yes

RSAAuthentication yes
PubkeyAuthentication yes

IgnoreRhosts yes
RhostsRSAAuthentication no
HostbasedAuthentication no
PermitEmptyPasswords no

ChallengeResponseAuthentication no
```

```
PrintMotd no
PrintLastLog yes
TCPKeepAlive yes

Banner /etc/issue.net

AcceptEnv LANG LC_*

UsePAM yes
```

Apêndice J

Configuração *dhcp*

```
#
# Configuração dhcp
# Versão: 1.0 -
# Responder comp. lab. informatica  MAC-ADDRESS
# incluídos no dhcp e disponibilizar
# endereços de 192.168.0.21 a 192.168.0.34
# Responder às maquinas da Rede Wireless
# a partir de 192.168.3.1 a 192.168.3.253

ddns-update-style none;
default-lease-time 600;
max-lease-time 7200;
authoritative;

# Rede Local
        subnet 192.168.0.0 netmask 255.255.255.0{
        range 192.168.0.21 192.168.0.34;
        option routers 192.168.0.254;
        option domain-name-servers 192.168.0.254;
        option broadcast-address 192.168.0.255;

# Micros concatenados por MAC Address
host lab_micro1 {
hardware ethernet 00:0F:EA:DE:35:C7;
fixed-address 192.168.0.21;
```

```
}

host lab_micro2 {
hardware ethernet 00:0F:EA:DE:38:9D;
fixed-address 192.168.0.22;
}

host lab_micro3 {
hardware ethernet 00:0F:EA:DE:53:5F;
fixed-address 192.168.0.23;
}

host lab_micro4 {
hardware ethernet 00:0F:EA:DE:50:34;
fixed-address 192.168.0.24;
}

host lab_micro5 {
hardware ethernet 00:1A:4D:A4:F2:8A;
fixed-address 192.168.0.25;
}

host lab_micro6 {
hardware ethernet 00:11:5B:F9:B7:47;
fixed-address 192.168.0.26;
}

host lab_micro7 {
hardware ethernet 00:01:6C:39:CE:85;
fixed-address 192.168.0.27;
}

host lab_micro8 {
hardware ethernet 00:0F:EA:DE:54:E7;
fixed-address 192.168.0.28;
}

host lab_micro9 {
hardware ethernet 00:0F:EA:DE:4E:03;
```

```
fixed-address 192.168.0.29;
}

host lab_micro10 {
hardware ethernet 00:0F:EA:DE:52:84;
fixed-address 192.168.0.30;
}

host lab_micro11 {
hardware ethernet 00:E0:4C:64:06:EB;
fixed-address 192.168.0.31;
}

host lab_micro12 {
hardware ethernet 00:0F:B0:55:EA:13;
fixed-address 192.168.0.32;
}
}

# Rede Wireless
        subnet 192.168.3.0 netmask 255.255.255.0{
        range 192.168.3.1 192.168.3.253;
        option routers 192.168.3.254;
        option domain-name-servers 192.168.3.254;
        option broadcast-address 192.168.3.255;
}
```

Apêndice K

Sistema de detecção *HDIS* - Tripwire

```
# Configuração padrao Tripwire

# Definição de variaveis globais
@@section GLOBAL
TWBIN = /usr/sbin;
TWETC = /etc/tripwire;
TWVAR = /var/lib/tripwire;

# Definição de aquivos de sistema
@@section FS
SEC_CRIT      = $(IgnoreNone)-SHa ;
SEC_BIN       = $(ReadOnly) ;
SEC_CONFIG    = $(Dynamic) ;
SEC_LOG       = $(Growing) ;
SEC_INVARIANT = +tpug ;
SIG_LOW       = 33 ;
SIG_MED       = 66 ;
SIG_HI        = 100 ;

# Tripwire Binario
```

```
(
  rulename = "Tripwire Binaries",
  severity = $(SIG_HI)
)
{
$(TWBIN)/siggen -> $(SEC_BIN) ;
$(TWBIN)/tripwire -> $(SEC_BIN) ;
$(TWBIN)/twadmin -> $(SEC_BIN) ;
$(TWBIN)/twprint -> $(SEC_BIN) ;
}

(
  rulename = "Tripwire Data Files",
  severity = $(SIG_HI)
)
{
$(TWVAR)/$(HOSTNAME).twd -> $(SEC_CONFIG) -i ;
$(TWETC)/tw.pol -> $(SEC_BIN) -i ;
$(TWETC)/tw.cfg -> $(SEC_BIN) -i ;
$(TWETC)/$(HOSTNAME)-local.key -> $(SEC_BIN) ;
$(TWETC)/site.key -> $(SEC_BIN) ;

$(TWVAR)/report -> $(SEC_CONFIG) (recurse=0) ;
}

# Arquivos de boot criticos
(
  rulename = "Critical system boot files",
  severity = $(SIG_HI)
)
{
/boot -> $(SEC_CRIT) ;
/lib/modules -> $(SEC_CRIT) ;
}

(
  rulename = "Boot Scripts",
  severity = $(SIG_HI)
)
```

```
{
/etc/init.d -> $(SEC_BIN) ;
/etc/rc.boot -> $(SEC_BIN) ;
/etc/rcS.d -> $(SEC_BIN) ;
/etc/rc0.d -> $(SEC_BIN) ;
/etc/rc1.d -> $(SEC_BIN) ;
/etc/rc2.d -> $(SEC_BIN) ;
/etc/rc3.d -> $(SEC_BIN) ;
/etc/rc4.d -> $(SEC_BIN) ;
/etc/rc5.d -> $(SEC_BIN) ;
/etc/rc6.d -> $(SEC_BIN) ;
}


# Executaveis criticos
(
  rulename = "Root file-system executables",
  severity = $(SIG_HI)
)
{
/bin -> $(SEC_BIN) ;
/sbin -> $(SEC_BIN) ;
}

# Bibliotecas Critica
(
  rulename = "Root file-system libraries",
  severity = $(SIG_HI)
)
{
/lib -> $(SEC_BIN) ;
}

# Programa de login e privilegios
(
  rulename = "Security Control",
  severity = $(SIG_MED)
)
{
```

```
/etc/passwd -> $(SEC_CONFIG) ;
/etc/shadow -> $(SEC_CONFIG) ;
}

# Arquivos de mudança de boot do sistema
(
  rulename = "System boot changes",
  severity = $(SIG_HI)
)
{
/var/lock -> $(SEC_CONFIG) ;
/var/run -> $(SEC_CONFIG) ;
/var/log -> $(SEC_CONFIG) ;
}

# Arquivos de mudança de root
(
  rulename = "Root config files",
  severity = 100
)
{
/root -> $(SEC_CRIT) ;
/root/mail -> $(SEC_CONFIG) ;
/root/Mail -> $(SEC_CONFIG) ;
/root/.xsession-errors -> $(SEC_CONFIG) ;
/root/.xauth -> $(SEC_CONFIG) ;
/root/.tcshrc -> $(SEC_CONFIG) ;
/root/.sawfish -> $(SEC_CONFIG) ;
/root/.pinerc -> $(SEC_CONFIG) ;
/root/.mc -> $(SEC_CONFIG) ;
/root/.gnome_private -> $(SEC_CONFIG) ;
/root/.gnome-desktop -> $(SEC_CONFIG) ;
/root/.gnome -> $(SEC_CONFIG) ;
/root/.esd_auth -> $(SEC_CONFIG) ;
/root/.elm -> $(SEC_CONFIG) ;
/root/.cshrc         -> $(SEC_CONFIG) ;
/root/.bashrc -> $(SEC_CONFIG) ;
/root/.bash_profile -> $(SEC_CONFIG) ;
/root/.bash_logout -> $(SEC_CONFIG) ;
```

```
/root/.bash_history -> $(SEC_CONFIG) ;
/root/.amandahosts -> $(SEC_CONFIG) ;
/root/.addressbook.lu -> $(SEC_CONFIG) ;
/root/.addressbook -> $(SEC_CONFIG) ;
/root/.Xresources -> $(SEC_CONFIG) ;
/root/.Xauthority -> $(SEC_CONFIG) -i ;
/root/.ICEauthority     -> $(SEC_CONFIG) ;
}

# Dispositivos criticos
(
  rulename = "Devices & Kernel information",
  severity = $(SIG_HI),
)
{
/dev -> $(Device) ;
/proc -> $(Device) ;
}

# Outros arquivos de configuração
(
  rulename = "Other configuration files",
  severity = $(SIG_MED)
)
{
/etc -> $(SEC_BIN) ;
}

# Binarios
(
  rulename = "Other binaries",
  severity = $(SIG_MED)
)
{
/usr/local/sbin -> $(SEC_BIN) ;
/usr/local/bin -> $(SEC_BIN) ;
/usr/sbin -> $(SEC_BIN) ;
/usr/bin -> $(SEC_BIN) ;
}
```

```
# Bibliotecas
(
  rulename = "Other libraries",
  severity = $(SIG_MED)
)
{
/usr/local/lib -> $(SEC_BIN) ;
/usr/lib -> $(SEC_BIN) ;
}

(
  rulename = "Invariant Directories",
  severity = $(SIG_MED)
)
{
/ -> $(SEC_INVARIANT) (recurse = 0) ;
/home -> $(SEC_INVARIANT) (recurse = 0) ;
/tmp -> $(SEC_INVARIANT) (recurse = 0) ;
/usr -> $(SEC_INVARIANT) (recurse = 0) ;
/var -> $(SEC_INVARIANT) (recurse = 0) ;
/var/tmp -> $(SEC_INVARIANT) (recurse = 0) ;
}
```