Fernando Sérgio Santos Fonseca

SISTEMA DE TRATAMENTO DE ARQUIVOS DE LOGS

Trabalho final apresentado ao Departamento de pós-graduação da Universidade Federal de Lavras, como parte das exigências do curso de pós-graduação Latu Sensu em Administração de Redes Linux para a obtenção do do título de especialista em Administração de Redes Linux.

Orientador

Prof. JOAQUIM QUINTEIRO UCHOA

LAVRAS

MINAS GERAIS - BRASIL

2005

Fernando Sérgio Santos Fonseca

SISTEMA DE TRATAMENTO DE ARQUIVOS DE LOGS

Trabalho final apresentado ao Departamento de pós-graduação da Universidade Federal de Lavras, como parte das exigências do curso de pós-graduação Latu Sensu em Administração de Redes Linux para a obtenção do do título de especialista em Administração de Redes Linux.

Aprovada em 17 de Abril de 2005

Prof. Herlon Ayres Camargo

Prof. Gustavo Guimarães Parma

Prof. Joaquim Quinteiro Uchôa
UFLA
(Orientador)

LAVRAS
MINAS GERAIS - BRASIL

Dedico este trabalho a minha mãe Margarida, à memória de meu pai Ubirajara e de minha avó Maria que me empurraram dia após dia para as aulas em minha infância e me ensinaram que ética e conhecimento são as maiores heranças que se pode deixar para os filhos.

Agradecimentos

A Deus meu amigo, que a cada manhã me concede a oportunidade de usar as horas de meu dia para mais um grande aprendizado e coloca sabiamente em meu caminho pessoas que sempre contribuem para minha evolução, seja em forma de pedra, mestre ou amigo.

A todas as pessoas que dedicam seu tempo a disseminar seu conhecimento pela Internet, e desenvolver programas para toda a sociedade, sem elas este e muitos outros trabalhos não estariam completos.

Agradeço especialmente à minha doce companheira Daniela que me ajudou de várias formas na conclusão deste trabalho.

Resumo

O Objetivo deste trabalho é estudar as formas de utilização do Syslog, Syslog-ng e servidores de *log,* assim como sua importância do tratamento de arquivos de *log* no aspecto da segurança computacional visando atender às especificações contidas nos ítens 9.7.1. e 9.7.2 da norma de segurança da informação NBR ISO/IEC 17799 (ABNT 2001).

Neste trabalho será abordado o redirecionamento de mensagens para outros servidores, a criação de servidores de alta segurança para receberem estas mensagens e manterem sua integridade e as diversas formas de se analisá-las e gerar alertas de acordo com a sua criticidade para o ambiente de TI. Considera-se que estas três abordagens juntas proporcionarão o melhor ambiente de monitoramento e segurança se trabalhadas com atenção e empenho.

Sumário

Lista de Figuras

Lista de Tabelas

1.Introdução

O Syslog é o serviço que registra as ações e eventos dos programas em execução em *hosts* Unix, Linux, roteadores, *Switches*, *Print Servers* e qualquer outro dispositivo que adote o padrão de mensagens definido pelo IETF (Internet Engineering Task Force) na RFC 3164 (IETF 2001). Em geral, estes registros chamados *logs* são salvos em arquivos geralmente localizados no diretório "/var/log" e possuem informações sobre data, hora, *host* e a mensagem emitida. Os *logs* podem ser configurados para registrar desde somente os eventos críticos até praticamente todos os eventos do sistema. É nos *logs* que encontram-se informações sobre o funcionamento dos programas servindo como ferramenta para a correção de erros e verificações de rotina.

> *Logs são muito importantes para a administração segura de sistemas, pois registram informações sobre o seu funcionamento e sobre eventos por eles detectados. Muitas vezes, os logs são o único recurso que um administrador possui para descobrir as causas de um problema ou comportamento anômalo. (NIC BR, 2003).*

O Syslog informa também sobre tentativas de acesso ao sistema ou a recursos do sistema, sendo uma ferramenta indispensável para uma análise de incidentes ou investigações. Em resumo o Syslog é uma grande fonte de informações para ajudar na prevenção e solução de problemas de funcionamento e segurança, subsidiando a manutenção da disponibilidade, integridade e confidencialidade dos dados. Na atualidade os ataques são

complexos e divididos em fases, sem um registro adequado não é possível entender como foi conduzido um ataque e consequentemente não é possível prevenir os próximos, pois não se conhece suas etapas.

Neste estudo será utilizado o sistema operacional Linux como base para instalação do Syslog, assim como o Windows XP, utilizando de algumas ferramentas Freeware e OpenSource que tornam possível que sistemas baseados em tecnologia Windows NT recebam e enviem mensagens no padrão Syslog.

O objetivo final do estudo é estabelecer uma plataforma única de tratamento de eventos do sistema, seja este um Unix, Windows, Cisco IOS ou qualquer outro que possa gerar mensagens no formato Syslog, mas sempre com foco no Linux para tratamento e armazenamento destas mensagens.

2. Gerenciadores de *Log*

Nos Linux os *logs* são gerados a partir de 2 *daemons* que devem estar permanentemente em execução, o Syslogd e o Klogd que registram os eventos do sistema e do kernel respectivamente e os registros seguem a formatação apresentada na figura 1.

Data | Hora | Máquina | *Daemon* | Mensagem

Figura 1: Formato dos registros de eventos

Os serviço de Syslog e Klog são inicializados automaticamente pelo "/etc/init.d/syslog" onde pode-se configurar as opções de inicialização e podem ser acessados pelos comandos descritos na figura 2.

```
Service syslog start
        Inicia o Syslogd e o Klogd

Service syslog stop
        Para o Syslogd e o Klogd

Service syslog rhstatus
        Mostra o status e o PID dos deamons Syslogd e Klogd

Service syslog restart ou reload
        Para e reinicia o Syslogd e o Klogd

Service syslog condrestart
        Para e reinicia o Syslogd e o Klogd, mas apenas se o Syslog estiver executando
```

Figura 2: Opções do Syslog

2.1 O Daemon Klogd

O *daemon* Klogd monitora as mensagens de saída do *kernel* e as envia para o *daemon* de monitoramento, que normalmente é o Syslogd. As opções para inicialização do Klogd podem ser vistas na figura 3

klogd [opções]

-d Ativa o modo de depuração do daemon

-f [arquivo] Envia as mensagens do kernel para o arquivo especificado ao invés de enviar ao daemon do Syslog

-i Envia um sinal para o daemon recarregar os símbolos de módulos do kernel.

-I Envia um sinal para o daemon recarregar os símbolos estáticos e de módulos do kernel.

-n Evita que o processo caia automaticamente em segundo plano. Útil se iniciado pelo init

-k [arquivo] Especifica o arquivo que contém os símbolos do kernel. Exemplos deste arquivo estão localizados em /boot/System.map-xx.xx.xx. necessária se desejar que sejam mostradas a tabela de símbolos ao invés de endereços numéricos do kernel

-o Faz com que o daemon leia e registre todas as mensagens encontradas nos buffers do kernel, após isto o daemon é encerrado.

-p Ativa o modo paranóia. Isto fará o Klogd somente carregar detalhes sobre os módulos quando os caracteres Oops forem detectados nas mensagens do kernel. É recomendável ter sempre a última versão do Klogd e evitar a utilização desta opção em ambientes críticos.

-s Força a utilização da interface de chamadas do sistema para comunicação com o kernel.

-x Esconde tradução EIP, assim ele não lê o arquivo /boot/System.map-xx-xx-xx.

Figura 3: Opções do Klogd

2.2 O Daemon Syslogd

O Syslog controla todos os registros do sistema e, de acordo com a configuração carregada do arquivo "/etc/syslog.conf", ele salva os eventos em arquivos especificados ou redireciona os mesmos para uma console, impressora ou outros *hosts.* As opções de inicialização para o Syslog podem ser vistas na figura 4.

syslogd [*opções*]

opções
-f Especifica um arquivo de configuração alternativo ao /etc/syslog.conf.

-h Permite redirecionar mensagens recebidas a outros servidores de logs especificados.

-l [computadores] Especifica um ou mais computadores (separados por ":") que deverão ser registrados somente com o nome de máquina ao invés do FQDN (nome completo, incluindo domínio).

-m [minutos] Intervalo em *minutos* que o Syslog mostrará a mensagem --MARK--. O valor padrão padrão é 20 minutos, 0 desativa.

-n Evita que o processo caia automaticamente em segundo plano. Necessário principalmente se o syslogd for controlado pelo init.

-p [soquete] Especifica um soquete UNIX alternativo ao invés de usar o padrão /dev/log.

-r Permite o recebimento de mensagens através da rede através da porta UDP 514. Esta opção é útil para criar um servidor de logs centralizado na rede. Por padrão, o servidor Syslog rejeitará conexões externas.

-s [domínios] Especifica a lista de domínios (separados por ":") que deverão ser retirados antes de enviados ao *log*.

-a [soquetes] Especifica soquetes adicionais que serão monitorados. Esta opção será necessária se estiver usando um ambiente chroot. É possível usar até 19 soquetes adicionais

-d Ativa o modo de depuração do Syslog. O Syslog permanecerá operando em primeiro plano e mostrará as mensagens no terminal atual.

Figura 4: Opções do Syslogd

2.3 O Arquivo syslog.conf

As eventos que chegam ao Syslog podem se originar do Klogd, dos programas em execução ou da rede e uma vez que cheguem ao Syslog devem ser tratados de acordo com as configurações estabelecidas no arquivo syslog.conf.

O Syslog recebe as mensagens assinaladas com o formato *facilidade.nivel*, ou seja, o sistema que gerou a mensagem e o nível de importância da mesma. O desenvolvedor de um programa escolhe a facilidade que deseja associar à sua aplicação e algumas vezes esta opção está disponível para ser configurada pelo usuário. Como o sistema pode gerar diversas mensagens por minuto cabe aos administradores configurarem um destino para estas mensagens de acordo com o sistema que a gerou, o nível de importância da mesma ou a combinação de sistema + importância.

Estas configurações são feitas no arquivo syslog.conf e seguem o formato apresentado na figura 5, especificando-se os sistemas como descritos na tabela 1 e os níveis de importância descritos na tabela 2. Caracteres especiais também podem ser utilizados para especificar um conjunto de níveis ou facilidades conforme a tabela 3.

facilidade.nivel	destino

Figura 5: Formato do arquivo syslog.conf

Facilidade	É usada para especificar que tipo de programa está enviando a mensagem. Os seguintes níveis são permitidos (em ordem alfabética):
auth	Mensagens de segurança/autorização
authpriv	Mensagens de segurança/autorização (privativas).
cron	Daemons de agendamento (cron e at).
daemon	Outros daemons do sistema que não possuem facilidades específicas.
ftp	Daemon de ftp do sistema.
kern	Mensagens do kernel.
lpr	Subsistema de impressão.
local0 a local7	Reservados para uso local.
mail	Subsistema de e- mail.
news	Subsistema de notícias da USENET.
security	Sinônimo para a facilidade auth (evite utilizá-la).
syslog	Mensagens internas geradas pelo syslogd.
user	Mensagens genéricas de nível do usuário.
uucp	Subsistema de UUCP.

Tabela 1 – Facilidades (sistemas) que geram eventos

Nível	Especifica a importância da mensagem. Os seguintes níveis são permitidos (em ordem de importância invertida; da mais para a menos importante):
emerg ou panic	O sistema está inutilizável.
alert	Uma ação deve ser tomada imediatamente para resolver o problema.
crit	Condições críticas.
err ou error	Condições de erro.
warning ou warn	Condições de alerta.
notice	Condição normal, mas significante.
info	Mensagens informativas.
debug	Mensagens de depuração.
none	Nenhuma prioridade

Tabela 2 – Níveis de importância das mensagens

"*"	Todas as mensagens da facilidade especificada serão redirecionadas.
"="	Somente o nível especificado será registrado.
"!"	Todos os níveis especificados e maiores NÃO serão registrados.
"-"	Pode ser usado para desativar o sync imediato do arquivo após sua gravação.

Tabela 3 – Caracteres de funções especiais

Uma vez que se conhece a origem das mensagens e o nível de criticidade que é atribuído a cada situação, pode ser criado um tratamento para estas mensagens. O *host* pode executar ações de forma proativa, avisando sobre o problema ou tomando ações corretivas que reduzam o impacto do problema ocorrido. Para isso pode-se configurar o Syslog para redirecionar as mensagens para diversos arquivos, *hosts* ou dispositivos de acordo com seu conteúdo configurando um conjunto de especificações no arquivo syslog.conf conforme listados na tabela 4.

Destino	Descrição
/caminho/para/arquivo	As mensagens serão acrescentadas ao arquivo especificado (forma mais comum).
@host_registro	As mensagens serão enviadas ao servidor do Syslog na máquina "host_registro".
|/caminho/para/pipe	As mensagens serão escritas no pipe especificado (bom para filtrar com um programa externo).
*	As mensagens serão escritas a todos os usuários conectados.
/dev/console	As mensagens serão escritas nos ttys nomeados.

Tabela 4 – Destino para mensagens

2.4 Redirecionando eventos

Um recurso muito útil do Syslog é o redirecionamento de suas mensagens para outro *host*, este recurso se torna importante tanto no aspecto de administração e gerenciamento de redes quanto no aspecto segurança, uma vez que se um *host* for invadido o invasor não poderá apagar os "rastros" de sua invasão. Num sistema de *log* centralizado podemos redirecionar todas as mensagens ou somente as que se consideram mais críticas para o *host* destino, centralizando as informações de todos os dispositivos compatíveis com o Syslog em um único servidor, o que facilita a filtragem de eventos, visualização de mensagens e respostas automáticas a estas mensagens.

Para que os *hosts* possam enviar eventos do Syslog via rede é necessário configurar a inicialização do *daemon* do Syslog para que ele possa enviar mensagens via rede (UDP/514) e configurar no arquivo syslog.conf o nome do *host* para o qual os eventos devem ser enviados. Na figura 6 é apresentado um exemplo bem "enxuto" de configuração do syslog.conf. Nele pode-se ver o redirecionamento de mensagens para arquivos específicos e de mensagens de maior importância para o *host servlog*. Para este exemplo o nome *servlog* precisa estar cadastrado no arquivo *hosts* da máquina local ou no servidor DNS desta máquina. Caso a resolução de nomes não seja confiável pode-se utilizar o formato @IP. Ex: @192.168.10.1.

```
# registra todas as mensagens em /var/log/messages
*.debug   /var/log/messages

# escreve em terminais quando a situação é realmente grave
kern,daemon.crit    /dev/console
kern,daemon.crit    @servlog
*.warning                   @servlog

# separa outros arquivos.log para ser mais fácil a leitura
kern.debug                  /var/log/kern.log
mail.debug                  /var/log/mail.log
daemon.debug        /var/log/daemon.log
auth.debug                  /var/log/auth.log
syslog.debug                /var/log/syslog.log
authpriv.debug      /var/log/authpriv.log
ftp.debug                   /var/log/ftp.log

# aviso para os demais
user.notice                 /var/log/user.log
lpr.notice                  /var/log/lpr.log
news.notice                 /var/log/news.log
uucp.notice                 /var/log/uucp.log
cron.notice                 /var/log/cron.log
```

Figura 6 – Exemplo de syslog.conf

Para configurar a inicialização do *daemon* Syslog deve-se editar os arquivos /etc/sysconfig/syslog (se existir) e o /etc/init.d/syslog acrescentar a opção –h para a linha com as opções do syslogd como no exemplo:

SYSLOGD_OPTIONS = ”-m 0 *–h*”

Para configurar quais eventos devem ser redirecionados para quais servidores basta selecionar as mensagens e configurar utilizando o símbolo @ (arroba/at) seguido por seu endereço IP ou o nome do *host* se o mesmo for conhecido via arquivo *hosts* ou dns. Agora basta reiniciar o *daemon* Syslog para que ele carregue as novas configurações e seja capaz de encaminhar mensagens de Syslog pela rede.

É importante lembrar que uma vez que o evento foi direcionado para um servidor de *log* ele deverá ser novamente tratado no servidor, uma vez que ele chega no *daemon* de destino como uma mensagem normal.

2.5 O Formato da mensagem Syslog

Uma mensagem Syslog consiste simplesmente de um pacote UDP para a porta 514 do *host* de destino contendo uma *string* com um número decimal representando a facilidade e severidade, data e hora do evento e o corpo da mensagem em si.

O Cálculo do numero decimal correspondente à facilidade e severidade é obtido através da soma da facilidade multiplicado por 8 mais a severidade. Assim uma mensagem <01> representaria kernel.alert, <02> = kernel.critical, <10> = user.critical, etc. Uma relação dos valores atribuídos às facilidades e severidade pode ser vista na figura 7.

Facilidade:	
(0) kernel	(12) ntp
(1) user	(13) log audit
(2) mail	(14) log alert
(3) system	(15) clock 2
(4) security/auth 1	(16) local 0
(5) syslog	(17) local 1
(6) line printer	(18) local 2
(7) news	(19) local 3
(8) uucp	(20) local 4
(9) clock 1	(21) local 5
(10) security/auth 2	(22) local 6
(11) ftp	(23) local 7

Severidade:	
(0) emergency	(4) warning
(1) alert	(5) notice
(2) critical	(6) information
(3) error	(7) debug

Figura 7 – Relação de valores para Facilidade e Severidade

Na figura 8 pode-se observar o Ethereal[1] capturando pacotes enviados pelo Utilitário Logger para o Syslog em um *host* Linux e redirecionados para uma estação Windows executando o 3Com Windows Syslog Daemon. O próprio Ethereal é capaz de identificar a facilidade e severidade conforme a o cálculo demonstrado anteriormente. O logger e o 3Com Syslog Daemon serão abordados posteriormente neste estudo.

[1] Software do tipo *sniffer* que captura pacotes TCP/IP para análise de rede, pode ser encontrado em http://sourceforge.net/projetcs/ethereal.

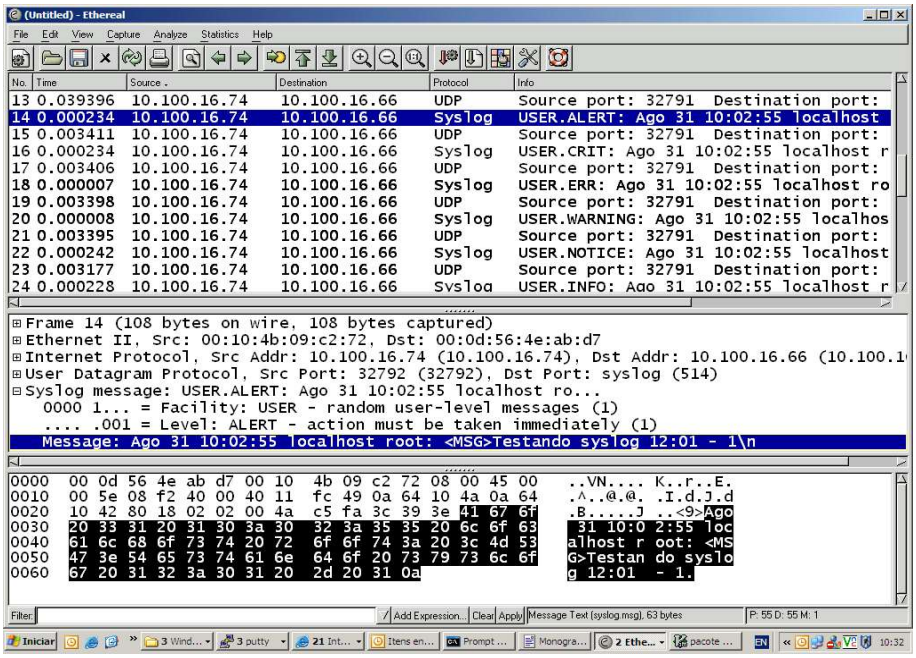

Figura 8 – Tela do Ethereal capturando um pacote Syslog

Conforme demonstrado Na figura 8 o protocolo do Syslog possui um cabeçalho simples contendo basicamente a facilidade * nível e o texto na mensagem em si. Esta simplicidade torna o padrão Syslog facilmente implementável mas também gera problemas de segurança uma vez que transporta as informações em texto puro, podendo ser facilmente interceptado ou alterado. Posteriormente neste estudo serão abordadas soluções de segurança para contornar esta vulnerabilidade.

3. O Servidor de *Logs*

Para ter-se uma visão centralizada dos eventos relevantes que ocorrem em nossa rede é necessário configurar um ou mais *hosts* para receber e tratar as informações provenientes dos outros *hosts* da rede. Este autor recomenda o uso de um servidor com uma versão ou um sistema operacional diferente dos servidores que enviarão o *log* pelo motivo que se o invasor conseguir atacar um *host* explorando uma vulnerabilidade do mesmo existe uma grande probabilidade de um *host* semelhante possuir a mesma vulnerabilidade.

É recomendável também o uso de um computador dedicado para tal tarefa, utilizando IPTABLES ou qualquer outro filtro de pacotes e aceitando conexões somente nas portas estritamente necessárias como a UDP/514 para receber eventos e TCP/22 para SSH por exemplo.

No caso do *host* que envia os eventos se encontrar em uma rede fisicamente separada da rede onde o servidor de *logs* se encontra pode-se utilizar o Stunnel (GERHARDS, 2004) para criar uma conexão segura SSL entre os *hosts* ligados por uma rede insegura, para isso basta configurar uma porta segura no Stunnel tanto no *host* de origem quanto de destino.

Por se tratarem de conceitos um pouco mais complexos e fora do escopo, o trabalho será limitado a sugerir estas abordagens de pacotes e Stunnel, servido estes como referências para estudos futuros.

3.1 O Sistema Operacional do Servidor de *Logs*

Seguindo a filosofia de um sevidor de *logs* "limpo" com poucos serviços sendo executado e poucas portas abertas disponíveis sugere-se também a utilização de sistemas operacionais com um nível de segurança mais alto como o Linux+LIDS, Linux+SELinux, OpenBSD, SecureBSD, etc. Estes sistemas são um pouco menos amigáveis para se trabalhar em função das sua configuração segura mas eles garantem a integridade dos *logs*, uma vez que na maioria deles nem o superusuário "root" é capaz de alterar o conteúdo dos arquivos de *log* nestes sistemas, o que garante que mesmo que o invasor se apodere da máquina ele pouco poderá fazer para adulterar os *logs.* A seguir haverá uma breve descrição de alguns destes sistemas:

3.1.1 LIDS

Projeto mantido pela comunidade de software livre no sourceforge.net e distribuido sob licença GNU General Public Licence, o *Linux Intrusion Detection System (LIDS)* é um *patch* para o kernel do Linux que acrescenta várias funcionalidades de segurança tais como Mandatory Access Controls (MACs), Detecção de Port Scanners, Proteção de acesso a arquivos e pastas (até mesmo do *root*) e proteção de processos, módulos e *interfaces.* Após aplicado o LIDS no kernel deve-se criar uma senha de administração que é gravada com criptografia de 185 *bits* e controla os programas lidsadm e lidsconf, responsáveis por prover e restringir acesso aos arquivos e outros recursos do sistema (TAMBORIM, 2004).

No LIDS pode-se restringir o acesso a arquivos com opções de somente leitura no caso de arquivos como os do sistema operacional que se deseja preservar intactos ou a opção de somente incrementar para o caso de arquivos como os de *log* que se precisa que sejam alterados mas somente para a inserção de linhas, nunca para deleção. Os comandos de atribuição de acesso do LIDS são semelhantes aos do IPTABLES como segue no exemplo:

```
# Protege a partição de boot
#
/sbin/lidsconf -A -o /boot -j READONLY

# Protege os binaries do sistema
#
/sbin/lidsconf -A -o /sbin -j READONLY
/sbin/lidsconf -A -o /bin -j READONLY

# Protege os logs do sistema
#
/sbin/lidsconf -A -o /var/log -j APPEND
/sbin/lidsconf -A -s /bin/login -o /var/log/wtmp -j WRITE
/sbin/lidsconf -A -s /bin/login -o /var/log/lastlog -j WRITE
/sbin/lidsconf -A -s /sbin/init -o /var/log/wtmp -j WRITE
/sbin/lidsconf -A -s /sbin/init -o /var/log/lastlog -j WRITE
/sbin/lidsconf -A -s /sbin/halt -o /var/log/wtmp -j WRITE
/sbin/lidsconf -A -s /sbin/halt -o /var/log/lastlog -j WRITE
/sbin/lidsconf -A -s /etc/rc.d/rc.sysinit -o /var/log/wtmp -i 1 -j WRITE
/sbin/lidsconf -A -s /etc/rc.d/rc.sysinit -o /var/log/lastlog -i 1 -j WRITE
```

Figura 9 – configuração do LIDS

Esta robustez e a facilidade de poder ser aplicado ao Kernel de qualquer distribuição faz do LIDS uma excelente opção para se proteger o servidor de *logs* em um projeto de centralização de *logs.*

3.1.2 SELinux

O SELinux é um projeto do NSA (National Security Agency) do governo norte americano que implementa através de um *patch* do *kernel* do linux e de aplicativos adaptados um sistema de controle de acesso chamado MAC (Mandatory Access Control). O MAC trabalha com a filosofia de se definir níveis de criticidade para recursos e acesso para usuários deixando por conta do sistema a definição do acesso ou não de forma mandatória. O SELinux modifica a forma de acesso aos recursos do sistema linux, fornecendo um grande aumento de segurança no acesso a arquivos, processos e *sockets* do sistema (NSA, 2004).

O SELinux proporciona uma mudança tão grande na utilização do linux que é necessário substituir alguns dos programas mais clássicos como o ls, mkdir, ps, id, find, login, sshd e crond para que operem de acordo com o novo controle inserido.

O SELinux está disponível para a comunidade Linux, tendo sido até incorporado a algumas distribuições como a Gentoo Linux e ao Fedora Core 2 em diante.

O Processo de configuração do SELinux envolve o aprendizado de novos modelos de segurança que tranquilamente seriam capazes de compreender vários estudos. Por achar inviável uma abordagem simples que forneça um modelo funcional do SELinux este autor se limita às citações feitas até agora.

3.1.3 Outros Sistemas

Existem outros projetos de *patches* que implementam alta segurança para Linux e BSD, por adotar o LIDS como o mais adequado para o proposito deste estudo será feita apenas uma citação aos demais sistemas.

O projeto TrustedBSD também cria inovações de segurança para a linha BSD, sendo desenvolvido primariamente para o FreeBSD alguns de seus componentes são eventualmente portados para OpenBSD e Darwin.

3.2 Stealth Logging

A palavra inglesa *Stealth* significa furtividade, este termo é utilizado para representar a tecnologia que visa tornar objetos invisíveis não sendo detectáveis quanto possível. Apesar de estar originalmente relacionada com aviões de guerra que se escondem do radar, o termo já foi bastante popularizado no mundo da informática.

Uma sugestão muito interessante dada por Lance Spitzner do projeto Honeynet (http://www.honeynet.org), foi a de se criar um servidor de *logs* que dificilmente poderiam ser alcançados por um invasor. Os servidores da DMZ enviariam suas mensagens de *log* para um IP que não foi atribuído a nenhum *host*. Como as mensagens utilizam o protocolo UDP os *hosts* não esperariam por um retorno para enviar as mensagens seguintes.

O segredo do processo estaria em se colocar um servidor de *log* sem um endereço IP conectado num mesmo *hub* ou *switch* (com uma porta espelhada) que os servidores da DMZ e trabalhando em modo promíscuo, isso faria com que este pudesse capturar os pacotes de Syslog sem que

pudesse ser endereçado por um invasor para tentar uma invasão. Para que este tipo de implementação funcione, alguns cuidados devem ser tomados. Os *hosts* da DMZ que enviarão as mensagens deverão ter uma entrada estática na tabela ARP para o endereço do servidor de *logs*, uma vez que o mesmo não existe e os pacotes não poderiam sair sem um endereço MAC de destino. Outro problema é que a *interface* sem IP não estará acessível a nenhum serviço, tornando necessário que se trabalhe na console do servidor ou que se instale uma segunda *interface* de rede configurada na rede interna.

3.3 A Configuração do Syslog no Servidor de *Logs*

Para que um *host* executando o Syslog possa receber eventos ele deve ser configurado para escutar a porta UDP/514, para isto basta editar os arquivos /etc/sysconfig/syslog (se existir) e o /etc/init.d/syslog acrescentar a opção –r para a linha com as opções do Syslogd como no exemplo:

SYSLOGD_OPTIONS = ”-m 0 –*r*”

Uma vez feito isso pode-se reiniciar o serviço de Syslog que o mesmo se iniciará com opção -r, ou seja, recebendo eventos da rede pela porta 514.

4. Utilitários para arquivos de *log*

4.1 Logger

O Logger é um utilitário que permite que se envie mensagens direto do *prompt* do sistema via syslogd ou *socket* do sistema, é possível especificar a prioridade, nível, identificador do processo e a mensagem em si. O Logger pode ser muito útil na fase de teste das configurações do Syslog.conf e para envio de eventos ao *log* scripts de shell[1]. Uma descrição dos parâmetros do comando é apresentada na figura 10.

logger [*opções*] [*mensagem*]

Onde:

mensagem - Mensagem que será enviada ao daemon *Syslog*

opções
-i - Registra o PID do processo
-s - Envia a mensagem ambos para a saída padrão (STDOUT) e Syslog.
-f [arquivo] - Envia o conteúdo do arquivo especificado como *mensagem* ao Syslog.
-t [nome] - Especifica o nome do processo responsável pelo log que será exibido antes do PID na mensagem do Syslog.

-p [prioridade] - Especifica a prioridade da mensagem do Syslog, especificada como facilidade.nível. O valor padrão *prioridade.nível* é *user.notice*
-u [soquete] - Envia a mensagem para o [soquete] especificado ao invés do Syslog

Figura 10 – Parâmetros do comando logger

Segue abaixo alguns exemplos de utilização do comando Logger:
Exemplo: logger -i -t Monografia Teste teste teste,

logger -i -t Monografia -p security.emerg

[1] Arquivos com comandos do shell executados em lote geralmente utilizados para realizar tarefas administrativas

4.2 Logrotate

O Logrotate é utilizado para fazer *backup* dos arquivos de *log* criando novos arquivos de *log* que serão utilizados pelo sistema. Os arquivos de *log* podem ser compactados ou enviados por e-mail para um determinado usuário. As configurações do Logrotate permitem que se especifique o tamanho máximo, número de arquivos e diversas outras opções que renderiam um estudo em separado de seu uso, este estudo limita-se a descrever as principais características da ferramenta e afirmar a importância desta para uma boa administração de *logs,* principalmente se associada e outras ferramentas para salva em meio magnético.

O Logrotate possui um arquivo de configuração para cada serviço que ele controla, estes arquivos podem ser encontrados no diretório "/etc/logrotate.d" e possuem os nomes correspondentes aos serviços que controlam como syslog, apache, squid, etc. Na figura 11 é possível verificar algumas das configurações do arquivo "/etc/logrotate.d/syslog" que é o objetivo deste estudo.

```
# Configuração do logrotate para os arquivos do Syslog
/var/log/messages /var/log/secure /var/log/maillog /var/log/boot.log /var/log/cron {
# Especifica que haverá um rotate de arquivos todos os dias
daily
# Configuração do logrotate para o syslog
# Especifica o número de arquivos de log que serão mantidos após cada rotate
rotate 5
# Especifica que um novo arquivo de log vazio será gerado e define sua permissões
create 0664 root utmp
# Compacta os arquivos antigos do logrotate com gzip
compress
# Especifica comando para ser executado após conclusão do logrotate
/sbin/killall -HUP syslogd
#indica final da configiração
endscript
 }
```

Figura 11 – Configuração do Logrotate

5. Syslog no Windows

Ao contrário da maioria dos sistemas operacionais que trabalham com TCP /IP o Windows (Tecnologia NT) não utiliza o Syslog, ele salva seus registros de eventos através do serviço chamado "*Event Log*". O Serviço de *log* do Windows é mais avançado que o Syslog quanto à proteção dos registros mas não oferece nenhuma funcionalidade nativa para centralização e tratamento. Os arquivos tratados pelo "Event Viewer" apresentam uma proteção bem maior aos dados devido à sua forma de armazenamento que salva os dados em formato binário, o que impede a simples edição de um arquivo de *log*. Existe também o fato de não ser possível nem ao administrador parar o serviço de *log*, para realizar isso é necessário desabilitá-lo e reiniciar a máquina.

Aliada a esta impossibilidade de se enviar os eventos pela rede gerando um *log* centralizado vê-se também as limitações de sua ferramenta de pesquisa, o "Event Viewer" que apresenta apenas funcionalidades básicas de pesquisa, e apesar de poder acessar *logs* de outros computadores na rede somente o faz um por um, sem nenhuma visão centralizada.

Em função destas deficiências e do desejo de se trabalhar de forma padronizada e centralizada com *logs* foram criadas várias alternativas comerciais ou gratuitas para se redirecionar eventos do Windows para servidores Syslog e servidores Syslog executando sob o Windows que serão também objeto deste estudo.

5.1 Redirecionadores

5.1.1 NTSyslog

O Projeto do NTSyslog desenvolvido por Jason Rhoads e mantido pela comunidade livre no site SourceForge[1] é um redirecionador dos eventos do Windows para um *deamon* Syslog. A instalação do produto é simples, basta descompactar o arquivo *zip* encontrado no site do projeto[2] e copiar o arquivo ntsyslog.exe para o diretório %SYSTEMROOT% \System32 e finalizar a instalação através do comando *Ntsyslog -install.*

Uma vez que o serviço esteja instalado o produto pode ser administrado pelo executável que acompanha o pacote ntsyslog.zip chamado NTsyslogCtrl.exe que apresenta a *interface* mostrada na figura 12

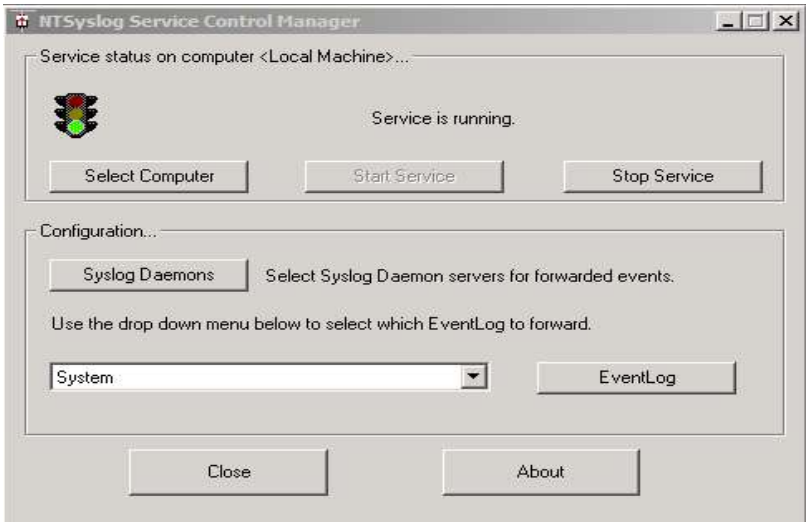

Figura 12 – Interface do NTSyslogCtrl do comando logger

[1] www.sourceforge.net,

[2] http://sourceforge.net/projects/ntsyslog/

Nesta *interface* o serviço pode ser ativado ou desativado, pode-se definir quais os *Daemons* receberão os eventos e selecionar com qual combinação facilidade.severidade os eventos serão enviados para cada tipo de alerta gerado pelo Windows, conforme mostra a Figura 13.

Figura 13 – Configuração de facilidade e severidade do NTsyslog

A Mensagem gerada pelo NTSyslog condensa as informações do "Event Viewer" em apenas uma linha, gerando eventos de Syslog conforme os mostrados em um arquivo texto na figura 14 ou capturados utilizando o Ethereal na figura 15 .

Conclui-se que esta ferramenta pode se tornar uma excelente aliada no controle de *logs* da rede Windows, uma vez que fornece a possibilidade de realizar um envio seletivo de mensagens enviadas que podem posteriormente serem tratadas pelo Syslog-ng ou Swatch, conforme será demonstrado mais à frente neste estudo.

Aug 31 15:27:17 10.100.16.66 security[failure] 680 AUTORIDADE NT\SYSTEM Logon tentado por: MICROSOFT_AUTHENTICATION_PACKAGE_V1_0 Conta de logo: Spock Estação de trabalho de origem: ENTERPRISE Código de erro: 0xC000006A

Aug 31 15:43:36 10.100.16.66 security[success] 560 FUP\KIRK Objeto aberto: Servidor de objetos:Security Account Manager Tipo de objeto:SAM_DOMAIN Nome do objeto:ENTERPRISE Identificação do identificador:834696 Identificação da operação: {0,143713097} Identificação do processo:844 Nome do arquivo de imagem: C:\WINDOWS\system32\lsass.exe Nome de usuário primário:ENTERPRISE $ Domínio primário:FUP Identificação do logon primário: (0x0,0x3E7) Nome de usuário cliente: KIRK Domínio do cliente:FUP Identificação do logon do cliente: (0x0,0x1B795) Acessos: %% 5392 %%5396 %%5401 Privilégios:- Contagem Sid restrita: 0 Aug 31 15:43:36 10.100.16.66 security[success] 624 FUP\KIRK Conta de usuário criada: Nome da conta nova:Spock Domínio novo: ENTERPRISE Identificador da conta nova: %{S-1-5-21-1715567821-884357618-839522115-1011} Nome de usuário chamador: KIRK Domínio do chamador: FUP Identificador do logon do chamador: (0x0,0x1B795) Privilégios-

Aug 31 15:43:36 10.100.16.66 security[success] 560 ENTERPRISE\KIRK Objeto aberto: Servidor de objetos:Security Account Manager Tipo de objeto:SAM_ALIAS Nome do objeto: DOMAINS\Builtin\Aliases\00000221 Identificação do identificador:836920 Identificação da operação:{0,143713195} Identificação do processo:844 Nome do arquivo de imagem: C:\WINDOWS\system32\lsass.exe Nome de usuário primário:K0721$ Domínio primário:ENTERPRISE Identificação do logon primário: (0x0,0x3E7) Nome de usuário cliente:KIRK Domínio do cliente:ENTERPRISE Identificação do logon do cliente: (0x0,0x1B795) Acessos:%%5424 Privilégios:- Contagem Sid restrita:0

Aug 31 15:43:36 10.100.16.66 security[success] 636 ENTERPRISE\KIRK Membro adicionado ao grupo local de segurança ativada: Nome do membro:- Identificação do membro: %{S-1-5-21-1715567821-884357618-839522115-1011} Nome da conta de destino:Usuários Domínio de destino:Builtin Identificador da conta de destino: %{S-1-5-32-545} Nome de usuário chamador:KIRK Domínio do chamador:ENTERPRISE Identificador do logon do chamador: (0x0,0x1B795) Privilégios:-

Figura 14 – Configuração de facilidade e severidade do Ntsyslog

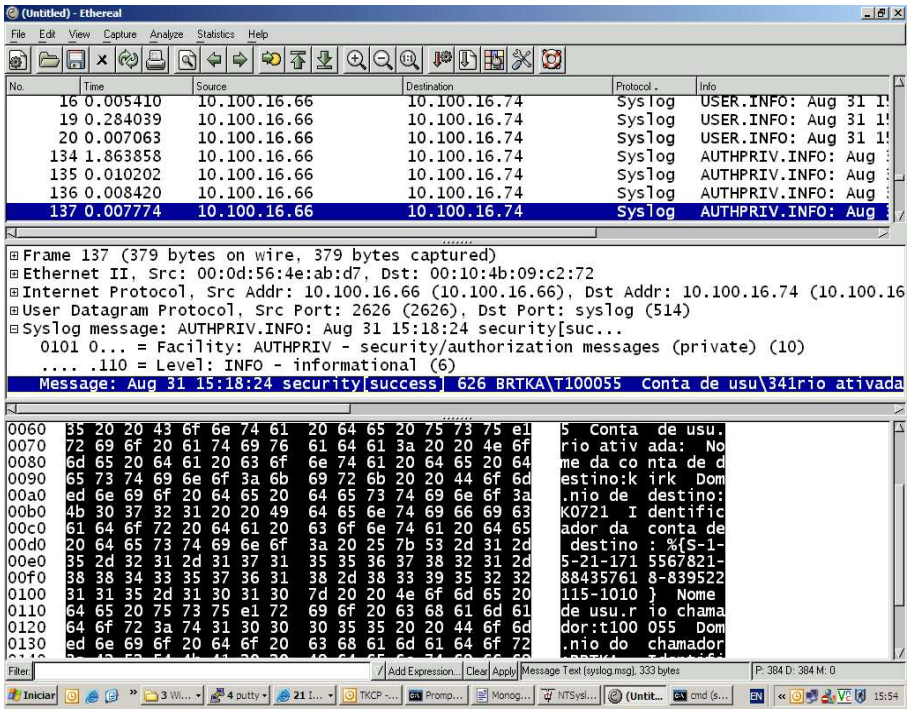

Figura 15 – Configuração de facilidade e severidade do Ntsyslog

5.1.2 O SNARE - System iNtrusion and Auditing Reporting Enviroment

O Snare é uma solução OpenSource para Windows hospedado pelo SourceForge e pode ser encontrado para *download* no *site* do projeto[1]. O objetivo do SNARE e detectar invasões através de *logs* com agentes que além de aumentarem o nível de auditoria no momento de sua instalação são capazes de selecionar determinados eventos e redirecioná-los para um servidor Syslog local ou remoto. A figura 16 mostra a tela de eventos do SNARE for Windows.

Figura 16 – Visualização de eventos do SNARE

A figura 17 mostra a primeira tela de configuração do SNARE, nela pode-se verificar a configuração do *host* de destino das mensagens, o cabeçalho Syslog a ser gerado (facilidade.severidade), as configurações de ajustar automaticamente a auditoria no Windows e um quadro com as associações dos eventos do Windows a serem enviados para o servidor Syslog.

[1] http://sourceforge.net/projects/snare/

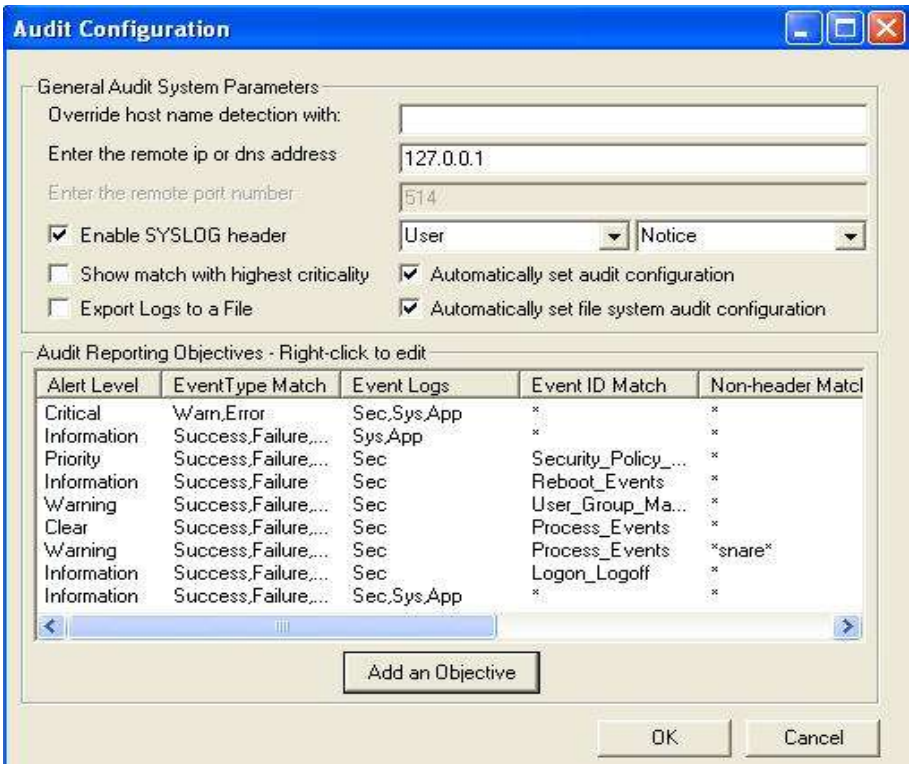

Figura 17 – Configuração do SNARE

A Figura 18 mostra a *interface* apresentada quando seleciona-se a opção de adicionar um objetivo na tela de configuração. O sistema dispõe de diversas opções de filtro de eventos do Windows para selecionar quais eventos serão enviados. O ponto fraco fica na impossibilidade de se poder associar uma prioridade (facilidade.severidade) a cada tipo de evento enviado.

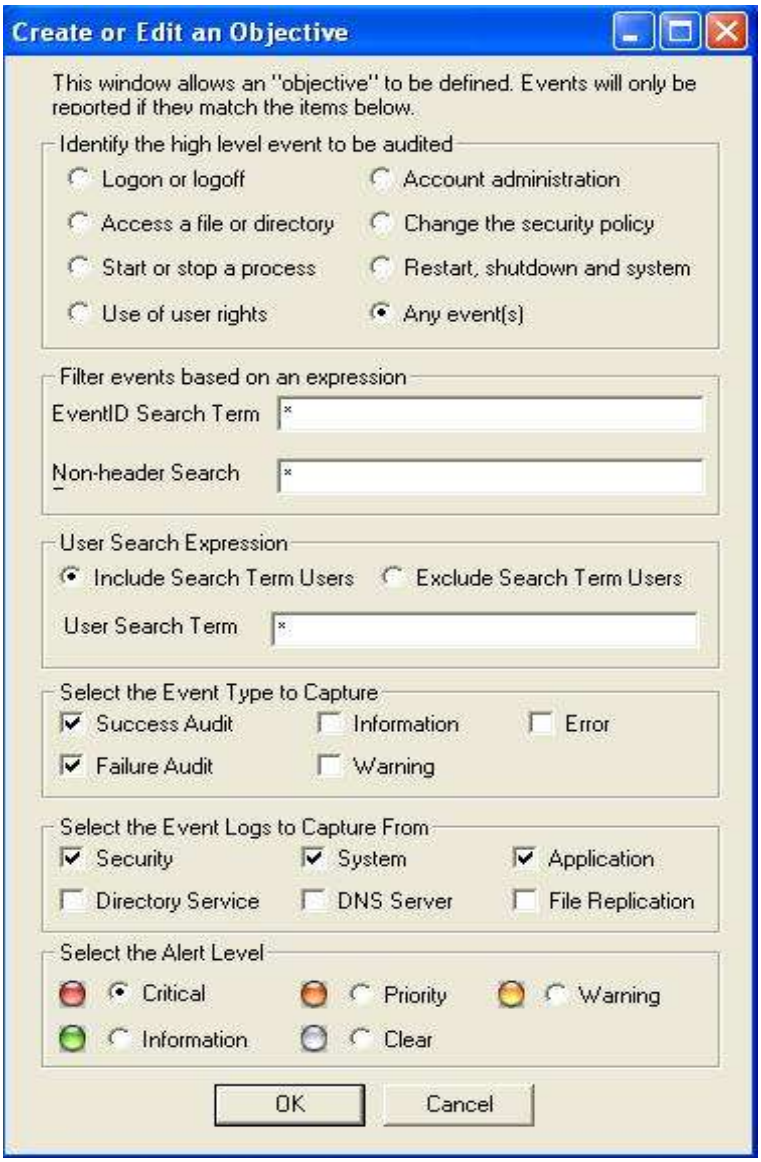

Figura 18 – Configuração de eventos a se capturar no SNARE

5.2 Servidores de Syslog

5.2.1 3ComWindows Syslog Daemon

A 3Com disponibiliza vários utilitários de Syslog sob licença *freeware* em sua página de suporte[1] um deles é o 3Com Windows Syslog Daemon que possui um servidor de Syslog e uma *interface* gráfica bem pobre e básica para se visualizar os eventos.

Para utiliza-lo, configura-se o arquivo "WsyslogD.ini" a partir do arquivo de exemplo "WsyslogD.ini.sample" que vem no pacote compactado do produto de acordo com a figura 19.

O próprio arquivo de configuração possui todas as explicações necessárias para se realizar a configuração do WsyslogD e o resultado é um ou mais arquivos de texto puro criados no diretório especificado por "Logdir" conforme mostrado na figura 20

[1] http://support.3com.com/software/utilities_for_windows_32_bit.htm.

```
# Este é um comentário. Comentários começam com "#"

# Entrada de  cabeçalho [Syslog]. Ela precisa existir para que o programa encontre os parâmetros
[Syslog]

# A linha  "LogDir" indica onde os arquivos de log serão criados.
# O Diretório deve existir previamente e ter permissões de escrita
# Certifique-se de incluir a barra final  "\" quado configurar o diretório
LogDir=C:\TEMP\

# "RotateLogs" permite a “rotação” de arquivos de log. Se contiver o valor  "1",
#  um novo arquivo de log será gerado a cad dia no formato
# "Mês-Dia-Ano.<nomedolog>.log". O valor "0" desabilita esta função
RotateLogs=1


# LogType determina em quais arquivos as mensagens de logserão salvas.

# LogType=1 significa que todas as mensagens irão para um único arquivo: syslog.log

# LogType=2 significa que será aberto um arquivo para cada endereço IP especificado na seção

# [Permit] . Se for especificado o valor "2", mas não houver endereços IP na seção [permit]

# serão aceitas mensagens de qualquer host e o valor de Logtype voltará a 1.
LogType=1

# A Seção [Permit]  especifica quais IP´s poderão enviar mensagens de log para este servidor.

# Coloque um IP por linha e não preencha com zeros, ou seja,

# use 10.1.1.1, NOT 010.001.001.001. Ise está seção  for omitida o parâmetro “Logtype” volta a 1

# e qualquer host que enviar mensagens para a pora 514 terá suas mensagens criadas salvas.

[Permit]
#10.1.1.1
#2.2.2.2
#3.3.3.3
#139.87.130.201
```

Figura 19 – Versão traduzida e explicada do arquivo WsyslogD.ini

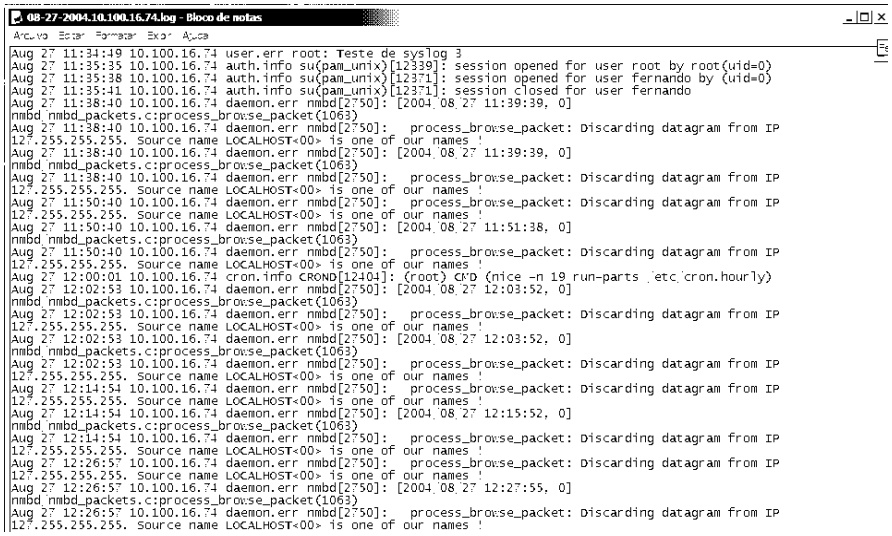

Figura 20 – Visualização do arquivo de log utilizando o Notepad

Após configurar o arquivo de parâmetros grava-se com o nome de ***"WsyslogD.ini"*** juntamente com o arquivo ***"WSyslogD.exe"*** para o diretório %Systemroot% ("C:\Windows" neste exemplo) e executa-se o comando ***WSyslogD.exe -i.*** Caso tudo ocorra corretamente haverá a seguinte mensagem: *3Com Syslog Server Installed.*

O Serviço pode ser iniciado com o comando "***WSyslogD.exe -s"*** e finalizado com o comando "***WSyslogD.exe -e".*** O serviço pode ser iniciado e parado também pelo ícone de serviços do painel de controle ou através dos comandos NET START "3Com Syslog Server" e NET STOP "3Com Syslog Server".

5.2.2 Kiwi Syslog Daemon

O pacote Kiwi Syslog Daemon é um *freeware* contendo um *daemon* Syslog que recebe mensagens pela porta 514 e um visualizador de eventos com muitas opções de visualização e estatísticas. A instalação é simples e feita através de um instalador padrão Windows, bastando executar o arquivo kiwi_syslogd.exe encontrado na página da Kiwi software[1].

Uma vez que o *daemon* esteja instalado será preciso executá-lo para que ele capture os eventos, somente a versão comercial do produto pode ser instalada como serviço. Na Figura 21 pode-se visualizar alguns eventos gerados pelo SNARE e pelo Kiwi logger.

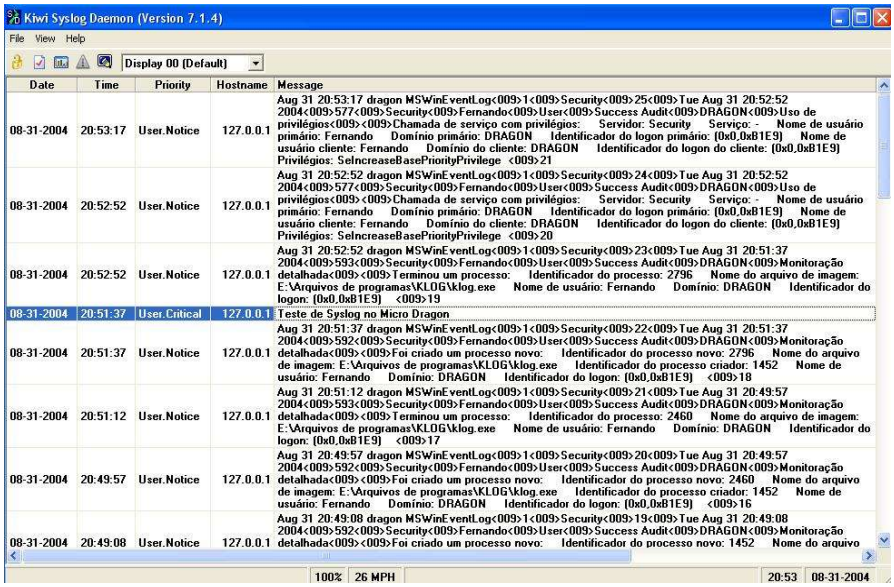

Figura 21 – Visualização de eventos do SNARE

[1] http://www.kiwisyslog.com

Uma característica interessante do produto são as estatísticas geradas pelo mesmo. O produto mostra gráficos com históricos de mensagens nas últimas 1 e 24 horas, distribuição percentual por severidade, média de mensagens recebidas por período e os 20 *hosts* que mais enviam mensagens conforme demonstrado na figura 22.

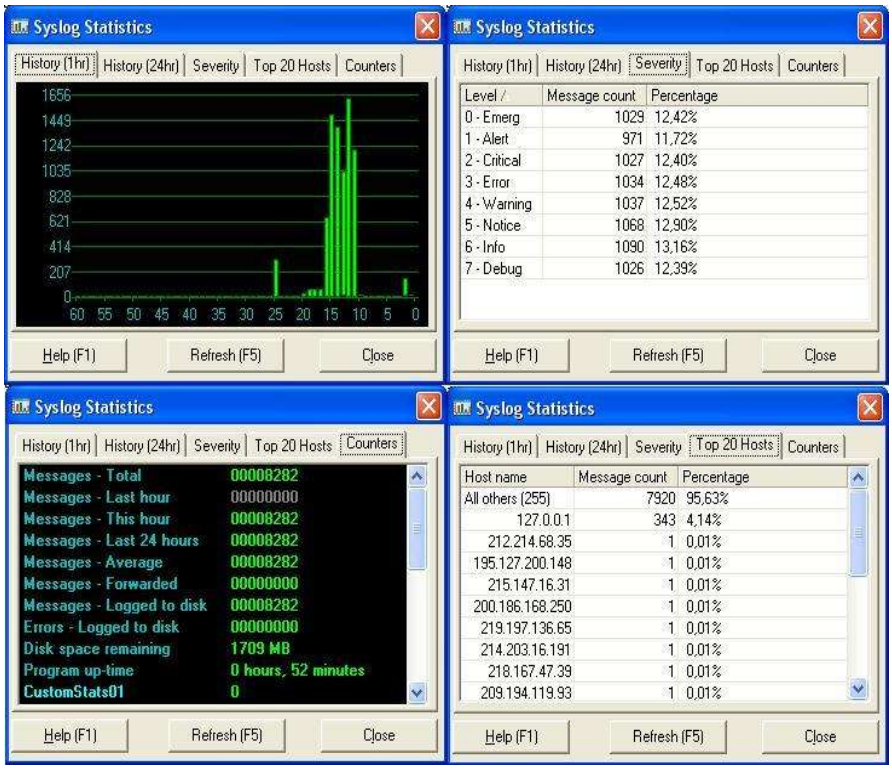

Figura 22 – Visualização de eventos do SNARE

5.3 Outras Ferramentas para Windows

5.3.1 Klog

O Klog e Klogwin são utilitários de linha de comando utilizados para enviar mensagens a um *daemon* Syslog, de forma semelhante ao comando Logger do Unix/Linux. Suas opções para envio de mensagens são listadas na tabela 5.

-u <port>	Porta de destino no host destino
-h <host>	Endereço do hot de destino do Syslog Daemon
-p <priority>	Prioridade da mensagem (0 to 191)
-f <facility no>	Facilidade da mensagem (0 to 23)
-l <level no>	Severidade da mensagem (0 to 7)
-F <facility name>	Nome da facilidade (kernel a local7)
-L <level name>	Nome da severidade (emergency a debug)
-r <process name>	Nome do processo que envia a mensagem
-R	Usar novo formato padrão RFC
-t	Usar sockets TCP ao invés de UDP
-w	Repetir a mensagem a cada segundo até que se pressione uma tecla
-s	*Log* silencioso
-m <message>	Texto da mensagem

Tabela 5 – Opções dos comandos Klog e Klogwin

Todos parâmetros nos comandos de linha klog e klogwin são opcionais, com excessão do -m (a mensagem em si). O *host* padrão para envio de mensagens é o *localhost.*

6. O Syslog-ng

O Syslog-ng (*new generation*) é uma evolução do Syslog e apresenta como principal evolução a possibilidade de se se organizar as mensagens por seu conteúdo e não somente por facilidade.nivel como no Syslog. Para que isso se torne possível o Syslog-ng utiliza expressões regulares para fazer a filtragem das mensagens e definir para onde enviá-las. Não está no escopo deste estudo discutir como utilizar expressões regulares dada a vasta divulgação e documentação sobre as mesmas.

O Syslog-ng pode ser encontrado em formato de código fonte ou RPM para o Suse Linux, para se utilizar o Syslog-ng deve-se instalar também a biblioteca “Libol”, ou indicar o seu caminho no arquivo configure do Syslog para utilizá-la através da opção –with-libol=/path/to/libol.

Uma vez que o Syslog-ng esteja instalado (via compilação ou RPM) deve-se criar ou configurar o arquivo /etc/syslog-ng.conf. Alguns exemplos do Syslog-ng.conf podem ser encontrados no diretório “doc” do fonte descompactado. Analisando o Syslog-ng.conf na figura 25 pode-se perceber que a sintaxe do Syslog-ng é muito mais avançada e completamente diferente da utilizada pelo Syslog, ela se baseia em 4 elementos que pode ser visto na figura **23.**

source ==>	Define de onde o Syslog-ng recebe os logs, que pode ser do sistema local como também de um servidor remoto.
destination ==>	Define para onde o Syslog-ng enviará os eventos recebidos que pode ser desde para um simples arquivo local/script ou até mesmo para um servidor de logs.
Filter ==>	Definição de filtros combinados entre si, podendo inclusive utilizar expressões regulares para separar os logs.
Log ==>	Diretiva final que indica que o que vier de "source" e atender a determinado "filter" Deve ser enviado para "destination"

Figura 23 – Parâmetros de configuração do Syslog-ng.conf

No Syslog-ng os elementos origem, filtro e destino são criados pelo administrador separadamente uns dos outros para posteriormente serem combinados em uma ação representada pelo diretiva "*log*". É possivel especificar mais de uma origem para um determinado destino ou mais de um destino para mensagens de uma determinada origem numa mesma instrução de *log*.

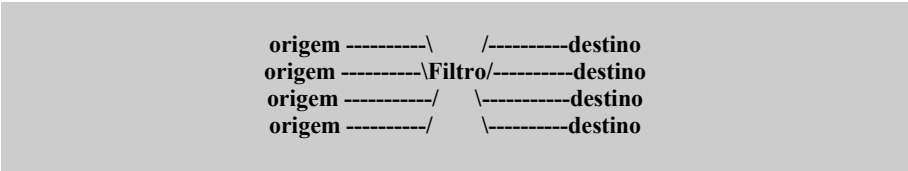

Figura 24 – Fluxo do Syslog-ng.conf

Adaptando-se um trecho apresentado na figura 6 e convertendo para Syslog-ng ter-se-a o resultado apresentado na figura 25

```
# escreve em terminais quando é situação é realmente grave
# No Syslog seria: kern,daemon.crit     /dev/console
#                                       kern,daemon.crit            @servlog
#                                       *.warning                   @servlog

destination console { file("/dev/tty12"); };
destination servlog { udp(servlog port(514)); };

filter f_daemon { facility(daemon); };
filter f_kern { facility(kern); };
filter f_warn { level(warn); };
filter f_crit { level(crit); };

log { source(src); filter(f_kern); filter(f_daemon); filter(f_crit); destination(console); };
log { source(src); filter(f_kern); filter(f_daemon); filter(f_crit); destination(servlog); };
log { source(src); filter(f_crit); destination(servlog); };
```

Figura 25 – Exemplo de configuração do syslog.conf

Após configurado o arquivo Syslog-ng.conf ainda é necessário que se altere algumas configurações do *host* para que este reconheça o Syslog-ng como seu *daemon* padrão. Para tanto deve-se alterar o arquivo "/etc/init.d/syslog". Também é necessário que se substitua o arquivo de configuração do logrotate "/etc/logrotate.d/syslog" pelo arquivo atualizado "/etc/logrotate.d/syslog-ng" (GONDIM, 2004).

7. SWATCH – Analise de *log* em tempo real.

O Syslog-ng pode executar comandos a partir de padrões detectados em mensagens, que correspondam a uma expressão regular, utilizando o destino *"program()"*. Porem não há no Syslog-ng uma forma de evitar que este disparo repetido de ações, cause um congestionamento e até mesmo uma indisponibilidade no servidor. Isso acontece porque o Syslog pode registrar uma mensagem várias vezes por segundo e a execução de um programa em resposta a este evento pode demorar muitas vezes mais que este tempo.

Existe um utilitário desenvolvido por Stephen Hansen e Todd Atkins da universidade de Stanford com o objetivo específico de monitorar arquivos de *log* em tempo real, seu nome é Swatch (the "Simple Watcher"). O Swatch é capaz de analisar um arquivo de *log* e tomar ações atrvés de regras pré-determinadas utilizando expressões regulares, de forma muito simples. Seu maior diferencial em relação ao Syslog-ng é a possibilidade de se estabelecer um intervalo entre uma execução e outra de uma ação disparada por um mesmo evento. É importante salientar que o Swatch não é um *deamon* de *log*, ele é um aplicativo desenvolvido em Perl e somente analisa os eventos já gerados e tratados ou não pelo Syslog ou Syslog-ng. Devido esta construção é necessário que se tenha os módulos Perl instalados no servidor, além de gerar um uso de CPU varias vezes maior do que o do Syslog-ng, o que também não chega a constituir um problema.

7.1 O Arquivo de Configuração .Swatchrc

O Swatch possui um único arquivo de configuração chamado . swatchrc que por padrão se encontra na raiz do diretório de instalação do próprio programa. O arquivo contém padrões de texto formatado utilizando expressões regulares seguido da ação a ser tomada quando uma linha de *log* corresponder a este padrão. Um exemplo de configuração de uma ação pode ser visto na figura 26.

```
watchfor /File name too long/
        mail addresses=webmaster\@sectools.com.br,subject=Possivel\ tentativa\ de\
BufferOverflow
        bell 2
```

Figura 26 – Exemplo de configuração simples do swatch

No exemplo anterior foi configurada a pesquisa pela frase "*File name too long*" nos arquivos de *log* e como resposta a existencia desta frase duas ações deverão acontecer, o envio de um *e-mail* para o webmaster@sectools.com.br e um alerta sonoro com dois "beeps". No exemplo acima foi utilizada uma barra invertida antes da arroba do *e-mail* de destino e dos espaços no assunto do *e-mail*. Isso é necessário para que o Perl não considere a arroba como um caracter especial e a frase como uma só.

7.2 Tipos de Ação do Swatch

Para cada tipo de evento podemos determinar uma série de ações a serem realizadas, no exemplo da figura 26 duas ações foram tomadas mas muitas outras poderiam ser realizadas. Para se ter uma melhor compreensão do que o Swatch pode fazer, o primeiro passo é conhecer os tipo de ação que podem ser tomadas quando se identifica um padrão em uma mensagem. A tabela 7 possui uma indicação das ações possíveis do Swatch.

Ação	**Descrição**
Echo=*normal, underscore, blue, inverse, etc.*	Imprime na console a linha que corresponde ao padrão com a formatação selecionada de cor, negrito, sublinhada, etc. O Padrão para impressão é Normal.
Bell *N*	Imprime na console a linha que corresponde ao padrão e soa um "beep" *N* vezes. O Padrão é 1
Exec *comando*	Executa o comando ou script descrito em *comando*, podemos atribuir qualquer comando a ser executado, desde a execução de um arquivo MP3 com alerta à configuração de uma regra de IPTABLES.
Pipe *comando*	Encaminha a linha que corresponde ao padrão para o comando representado por *comando*
Thorttle *HH:MM:SS*	Aguarda por um período representado por *HH:MM:SS* após um padrão ser localizado para executar novamente uma ação relacionada ao mesmo padrão.

Tabela 6 – Comandos do Swatch

7.3 Recomendações de uso

Para se tirar total proveito do Swatch, assim como do Syslog-ng é necessário que se aprenda a utilizar corretamente as expressões regulares (*rege*x). O Assunto é tão amplo e as possibilidades tantas que existem até livros inteiros ensinando o uso de *regex*. Nos exemplos a seguir serão utilizadas algumas expressões mais simples para demonstrar um pouco mais do poder do Swatch. Vale a pena lembrar que o Swatch monitora apenas um arquivo por vez, caso seja crucial monitorar mais de um arquivo de *log* deve-se abrir várias instâncias do Swatch, cada uma lendo um arquivo de configuração diferente e consequentemente monitorando um *log* diferente.

Uma configuração frequente para o Swatch é o uso da expressão lógica *"ou"*, representada no Swatch pelo sinal de pipe "|" um exemplo interessante seria a expressão *"/reject|failed/"* que corresponderia para qualquer frase que contivesse as palavras *reject* ou *failure*.

Outra configuração interessante utiliza a opção de ignorar caixa *"case insensitive"* esta é uma modificação específica do Perl para as expressões regulares que possibilita a procura por um padrão em qualquer caixa. Vejamos por exemplo o termo *reject,* se for configurada a palavra de pesquisa como *"/reject/i"* será encontrado qualquer evento que corresponda à palavra reject, não importa como seja escrita. Ex: "Reject", "REJECT", "ReJeCt", Etc.

7.4 Comentários Finais

O Swatch aparece com uma ferramenta extremamente eficaz para se monitorar *logs* mas sua característica de monitorar apenas um arquivo impede que possamos organizar nossos *logs* em arquivos diferenciados por *hosts*, serviços, etc sem que seja necessário iniciar várias instâncias do Swatch. Dependendo da arquitetura de rede pode-se optar por tratar os arquivos em um servidor de *logs* com Syslog-ng e direcionar os eventos que necessitem de ação imediata para uma console de operador executando o Swatch, o que contornaria outra limitação do produto, que só soa "beeps" e exibe mensagens na console e na sessão que iniciou o Swatch.

8.Exemplos de aplicação

8.1 - Analise de tráfego no roteador

Uma empresa em particular possuía um roteador cisco com pouca memória e sistema operacional desatualizado, o que lhe impossibilitava fazer uma análise mais precisa do que trafegava na sua rede Frame Relay que unia várias unidades do grupo. A solução encontrada foi criar ACL's[1] no *Switch* para se controlar o tráfego entre as unidades e protegê-las de contaminação por vírus ou acesso indevido.

Iniciou-se então um levantamento parcial dos protocolos comumente usados dentro da empresa e criou-se regras que permitiam o trafego de todos os protocolos realmente necessários. Como ultima regra, ao invés de bloquear o acesso a portas não cadastradas, criou-se uma regra que permitia todo o tráfego, mas geravam um *log* dos pacotes que passavam por estas portas. Estes *logs* eram enviados para o servidor de logs, visando uma análise posterior.

Como resultado obteve-se uma relação de portas utilizadas e através desta criou-se novas regras e até mesmo detectou-se trafego irregular nestes *links*. Atualmente a ultima regra bloqueia o tráfego e faz um *log* deste bloqueio para que se possa apurar a origem do mesmo, o que se mostrou muito útil para detecção de vírus e *software* não atualizados.

---

[1] Access Control Lists – Lista de protocolos que podem passar por cada conexão de um *Switch* ou Roteador

8.2 SUDO

Pode se adotar o "Sudo"[1] para que os administradores de sistema, administradores de SAP e banco de dados pudessem executar somente determinados comandos com permissão de *root*. A cada vez que um comando é executado através do Sudo uma mensagem contendo data, hora, terminal, nome do usuário e o comando é enviada para o Syslog.

Para que estes comandos sejam documentados pode-se enviar estas mensagens para um servidor seguro utilizando o Syslog local e no servidor de logs todas as mensagens do sudo seriam enviadas para uma base MySQL para que regularmente se pudesse gerar relatórios com o histórico de comandos executados por usuário, data, etc.

[1] Sudo é um Freeware que habilita usuários comuns a executar comandos como root. Documentação e programas podem ser encontrados na pagina da courtesan www.courtesan.com/sudo/sudo.html.

Conclusão

O Syslog é uma ferramenta extremamente útil e simples de ser configurada mas possui poucas opções de análise dos eventos ocorridos no sistema se utilizada isoladamente. É certo que se bem explorada e associada a outros programas pode fornecer ao administrador recursos para detecção pro-ativa de invasão, problemas de funcionamento em componentes de *hardware* ou problemas com recursos do sistema, podendo se tornar a principal ferramenta de algumas empresas.

Devido aos avançados recursos do Syslog-ng e sua simplicidade de uso, é provável que este venha a substituir em definitivo o *daemon* Syslog. O principal motivo para a substituição é o fato de cada vez mais encontramos indícios importantes de invasão em eventos que são considerados apenas de notificação pelo Syslog e podem ser tratados mais a fundo pelo Syslog-ng. O Syslog-ng pode ser dispensável somente no caso de um estudo dos *logs* feito com ferramentas adicionais como o Swatch.

Uma vez que foram definidas as ferramentas para o redirecionamento, recepção e tratamento de eventos, assim como diferentes métodos para se construir servidores mais seguros resta desenvolver uma maneira de visualizar os eventos de uma forma organizada e trata-los em "tempo-real". Pode-se também configurar alguns "gatilhos" a serem acionados quando um determinado evento ocorrer. Exemplos de ação "disparada" podem ser a execução de um arquivo Wav ou MP3, o envio de um *e-mail* ou mensagem SMS para um ou mais celulares. Pode-se ainda executar a configuração dinâmica de uma regra de IPTABLES bloqueando acesso a determinada porta de um *host* ou qualquer outra coisa que seja possível de ser executado pelo *host* onde se encontra o servidor de *log*.

Estas técnicas são utilizadas por consagrados sistemas comerciais de gerenciamento de redes como o CA Unicenter TNG e o HP Openview. Estes sistemas costumam possuir um agente de *log* que examina os *logs* em tempo real e quando identificam um padrão reagem a este evento. Sistemas de detecção de intrusos (IDS) e sistemas de prevenção de intrusos (IPS) utilizam-se também dos *logs* para identificar uma ameaça à rede,

Este estudo apresenta todos os componentes para se realizar um gerenciamento proativo de *logs*, com especial atenção para o Syslog-ng e o Swatch. Cabe ao administrador estudar seu ambiente e verificar como chamar a atenção para um evento localmente com som e mensagens ou enviando mensagens e até mesmo discando para um número. As possibilidades são diversas.

Os arquivos de *log* são fundamentais para realizar atividades de detecção pró-ativa de problemas, para pesquisar a causa raiz de problemas ocorridos, conforme pregado pelo ITIL[1] e para realizar investigações forenses sobre acontecimentos em nossa empresa. Para que essas detecções sejam eficazes deve-se desenvolver um sistema que mantenha a integridade destes registros, preferencialmente em um local seguro e centralizado.

Mais importante do que as ferramentas é a criação de processos bem definidos de controle de *logs* para se garantir a rastreabilidade de qualquer problema. Para isso é necessário que se crie rotinas online para detecção de eventos suspeitos ou que comprometam a disponibilidade, integridade e confidencialidade do sistema, além de rotinas para *backup* de *logs* utilizado o logrotate e *backup* para meio magnético dos arquivos compactados de *logs* gerados pelo mesmo.

---

[1] IT Infrastructure Library, série de documentos com melhores práticas sobre gerenciamento de serviços

Uma atenção especial deve ser dada à fase de identificar os eventos a serem registrados. Uma auditoria de Windows ou um registro de pacotes do IPTables podem gerar muitos registros e neste ponto pergunta-se que atividades este servidor executa para não vir a sobrecarregar, CPU e disco com o registro de informações que não precisamos. A necessidade de controle varia de acordo com a sensibilidade de cada negócio, e o Syslog nos dá a flexibilidade de ser configurado com o grau de rigidez exato para as devidas necessidades, desde somente alertar níveis críticos até a redirecionar eventos que sugiram uma invasão para uma impressora matricial conectada ao *host*.

A centralização dos *logs* num servidor dedicado a receber os *logs,* especialmente preparado e com auto nível de segurança utilizando distribuições seguras se mostra uma excelente prática pois agrega uma segunda camada ao atacante, se o invasor dominar um *host* da rede ele terá que invadir o servidor de *logs* para apagar os rastros de sua invasão, o que envolve utilizar outro tipo de *exploit* se considerarmos que o servidor foi instalado com outro sistema, de preferência mais seguro.

Para que os registros sejam fieis e possa estabelecer uma relação entre os eventos ocorridos em mais de um servidor é fundamental que todos os servidores estejam acusando exatamente o mesmo horário. Outro ponto importante diz respeito aos pacotes de mensagens, uma vez que utiliza-se o UDP, que é um protocolo não orientado a conexão pode-se ter pacotes chegando fora de ordem ao servidor, o que tende a confundir o administrador. Como solução para este problema pode-se utilizar o protocolo NTP (Network Time Protocol) que é suportado em praticamente todas as plataformas (inclusive Windows) para sincronizar os *hosts*. No caso específico do Windows pode-se sincronizar apenas os servidores via NTP e via *login script* sincronizar as estações da rede.

Também é importante lembrar que no Brasil não há um horário de verão regular, a cada ano o governo estabelece uma data de início e termino para o mesmo provocando reconfiguração dos roteadores e servidores (Unix ou Linux) para que automaticamente alterem seu horário para o horário de verão, se esta operação for mal sucedida ou se simplesmente se alterar o horário manualmente podem ocorrer eventos gerados com um fuso-horário inadequado, uma vez que quando entra-se no horário de verão o que muda é o fuso, e não o horário em si. Deve-se ter muito cuidado nesta operação para não comprometer a integridade dos *logs*.

9.Referências Bibliográficas

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR-17799**: Tecnologia da informação – Código de prática para a gestão da segurança da informação. Rio de Janeiro, 2001.

GERHARDS, RAINER , Forwarding Windows Events via stunnel to a UNIX/Linux Syslog
On-line: URL
http://www.winsyslog.com/Common/en/Articles/eventlog-stunnel-syslog.asp.Visitado em 24/08/2004

GRUPO DE USUÁRIOS SLACKWARE, Como instalar o Syslog-ng no Slackware:
On-line: http://gus-br.linuxmag.com.br/pt/documentacao/syslog-ng-howto.html Visitado em 17/06/2004

IETF INTERNET ENGINEER TASK FORCE, RFC 3164 The BSD Syslog Protocol
On-Line: http://www.faqs.org/rfcs/rfc3164.html Visitado em 16/06/2004

NATIONAL SECURITY AGENCY, SELinux – Protecting System File Integrity
On-line: URL
http://www.nsa.gov/selinux/papers/ottawa01/node8.html. Visitado em 25/08/2004

NIC BR, Práticas de Segurança para Administradores de Redes Internet
On-line: URL http://www.nbso.nic.br/docs/seg-adm-redes/seg-adm-redes.pdf. Visitado em 27/03/2005

TAMBORIM, ANDERSON LUIZ , Segurança extrema com LIDS
On-line: URL http://www.y2h4ck.hpg.ig.com.br/lids.htm.Visitado em 25/08/2004

10.Bibliográfia

3Com Software Library, Utilities for Windows 32 Bit On-line: URL http://support.3com.com/software/utilities_for_windows_32_bit.htm Visitado em 31/03/05

BAUER, MICHAEL D., Building secure servers with Linux, 1a Edição, California, Estados Unidos da America.
O'Reilly, 464 páginas. Capitulo 10 "System Log Management and Monitoring" p.216-241.

GARBRECHT, FREDERICK C, Practical Implementation of Syslog in Mixed Windows Environments for Secure Centralized Audit Logging On-line: URL http://www.sans.org/rr/papers/index.php?id=713.Visitado em 25/08/2004

GRUPO DE USUÁRIOS SLACKWARE, Como instalar o Syslog-ng no Slackware:
On-line: http://gus-br.linuxmag.com.br/pt/documentacao/syslog-ng-howto.html Visitado em 17/06/2004

(HACKDARK), RODRIGO, Syslog em Cisco e Linux
On-line:
http://www.ciscotrainingbr.com/modules.php?name=News&file=article&sid=17
Visitado em 17/06/2004

KIWITOOLS, Kiwi Syslog Tools:
On-line:http://www.kiwisyslog.com/products.htm Visitado em 17/06/2004

NIST, Configuring Windows 2000 and Windows XP to use NIST Time Servers
On-line: URL
http://www.boulder.nist.gov/timefreq/service/pdf/win2000xp.pdf.
Visitado em 27/08/2004
MAPLES, WAYNE, SysLog Servers for NT/2000/XP - On-line: URL http://www.windowsnetworking.com/kbase/WindowsTips/WindowsXP/AdminTips/Security/SysLogServersforNT2000XP.html. Visitado em 25/08/2004

MATES, JEREMY , Logging with Syslog-ng
On-line: URL http://sial.org/howto/logging/syslog-ng/?style=printable.Visitado em 24/08/2004

PINTO, BRENO SILVA, Auditoria de log com o logcheck
On-line: http://www.secforum.com.br/article.php?sid=1738 Visitado em 16/06/2004

RNP, CAIS, Implementando o serviço NTP na sua rede local
On-line: URL
http://www.rnp.br/_arquivo/cais/manual_ntp_v1b.pdf.Visitado em 27/08/2004

RHOADS, JASON, NTSyslog
On-line: URL http://sourceforge.net/projects/ntsyslog/. Visitado em 31/08/04

SCHEIDLER, BALÁZS , syslog-ng reference manual
On-line: URL
http://www.balabit.com/products/syslog_ng/reference/book1.htm.Visitado em 24/08/2004

SILVA, GLEYDSON MAZIOLI DA, Guia Foca GNU/Linux
On-line: http://www.htmlstaff.org/guiafoca/avancado/ch-log.htm
Visitado em 15/06/2004

SILVA, LUIZ ANTONIO DA, Analisando arquivos de registro (log)
On-line: URL
http://www.vivaolinux.com.br/artigos/impressora.php?codigo=555.Visitado em 15/06/2004