Protegendo-se contra ataques

Mantendo-se seguro

Nestes tempos em que pragas virtuais assolam os sistemas Windows, os usuários de Linux tendem a se sentir a salvo. Ledo engano: falhas de segurança sempre foram um problema para o Linux e para os programas que nele rodam – foi assim e assim sempre será. Como podemos então minimizar o risco de invasão em nossos sistemas? POR RALF SPENNEBERG

As últimas notícias não nos deixam dúvidas: o Microsoft Windows parece estar fadado a ser a vítima número um de vírus e worms. Os usuários de Linux costumam acomodar-se em suas cadeiras e relaxar quanto a isso, pois julgam que o único problema que irão enfrentar é um certo aumento no tráfego de Internet.

Entretanto, pense um pouco no assunto antes de desdenhar dos infelizes: o que aconteceria se houvessem falhas de segurança similares no Linux e seus aplicativos, apenas esperando para ser exploradas ou mesmo criadas em versões futuras? O ataque aos servidores centrais do projeto Debian [1] é prova cabal de que mesmo administradores experientes são incapazes de proteger a si mesmos contra todos os tipos de vulnerabilidades. Só isso já é razão suficiente para que os usuários de Linux busquem adquirir um conhecimento básico sobre segurança de sistemas, como reconhecer ataques e invasões e as técnicas de resposta a incidentes que se seguem a uma invasão. Neste artigo, veremos algumas formas de minimizar esses riscos.

Atualizações constantes

Muitos hackers publicam as falhas de segurança que encontraram em listas de discussão [2,3]. Os programadores e mantenedores de aplicativos e mesmo as distribuidoras de Linux normalmente têm acesso a essas informações e isso permite que prontamente desenvolvam patches e correções.

A maneira mais fácil e, de longe, a melhor, de diminuir o risco de invasão ou outras mazelas é manter as máquinas atualizadas. Muitas distribuições possuem sistemas de atualização para facilitar a tarefa, e o cron [4] permite automatizá-los. Os usuários do Fedora, por exemplo, podem usar o comando *yum update* para atualizar seus sistemas; quem usa uma distribuição baseada no Debian deve usar os seguintes comandos:

```
/usr/bin/apt-get update -q -y
/usr/bin/apt-get upgrade -q -y
```

O comando *apt-get update* atualiza a lista de pacotes disponíveis. Os usuários de Debian devem sempre passar por essa etapa antes de verdadeiramente atualizar o sistema. O comando *apt-get upgrade* serve para isso: baixa e instala as últimas versões dos pacotes já instalados no sistema. A opção *-q* (“quiet”) suprime as informações na tela, enquanto *-y* responde automaticamente “yes” (sim) para qualquer pergunta que possa ser feita pelo programa.

A sintaxe desses comandos é talhada com perfeição para um job diário, ou mesmo horário, do cron. Normalmente, é necessário apenas criar um arquivo apropriado (por exemplo, */etc/cron.daily/atualizar*) e adicionar as duas linhas de comando nele. Por fim, digite

Listagem 1: Comando lsof -i

```
[root@kermit root]# lsof -i
COMMAND  PID  USER  FD  TYPE  DEVICE  SIZE  NODE NAME
ntpd     882   ntp  4u  IPv4          2788  UDP  *:ntp
ntpd     882   ntp  5u  IPv4          2789  UDP  localhost:ntp
sshd     942  root  3u  IPv4          2858  TCP  *:ssh (LISTEN)
master  1019  root 11u  IPv4          2972  TCP  *:smtp (LISTEN)
cupsd   1569  root  0u  IPv4          3720  TCP  localhost:ipp (LISTEN)
cupsd   1569  root  2u  IPv4          3721  UDP  *:631
```

```
chmod 755 /etc/cron.daily/➥
atualizar
```

para tornar o arquivo executável. O cron irá automaticamente executar os comandos do arquivo diariamente.

Um sistema adequadamente atualizado oferece mais segurança do que qualquer combinação de firewall e antivírus. Os administradores precisam, ainda, estar atentos aos novos "furos" que aparecem diariamente. O segundo passo para uma máquina mais difícil de invadir é desabilitar quaisquer serviços que não estejam sendo usados. Felizmente, as distribuições mais novas não habilitam, por padrão, todos os serviços instalados no computador.

Minimalismo

Os administradores que desejarem descobrir quais serviços estão ativos no sistema têm duas opções. A primeira é emitir o comando *lsof -i* na própria máquina; a segunda é disparar remotamente um scanner de redes contra a máquina sob testes, sendo o nmap a escolha mais indicada. O comando lsof mostra os arquivos abertos num sistema Linux. Se levarmos em conta que qualquer coisa num sistema Linux é um arquivo, as conexões de rede também o são. A opção *-i* diz ao lsof para mostrar todas as conexões de rede. Um exemplo disso pode ser visto na Listagem 1.

A primeira coluna indica o comando que abriu a conexão de rede. A coluna à direita mostra o PID. A terceira coluna traz o usuário dono do processo e os privilégios com os quais o serviço está rodando. As últimas duas colunas são mais interessantes, entretanto. A coluna NODE indica o protocolo que este serviço usa para falar com o mundo lá fora (TCP ou UDP). A coluna NAME mostra o endereço IP e porta usada pelo serviço. O programa ainda traduz os IPs e portas para nomes de domínio (usando DNS e o arquivo */etc/hosts*) e nomes conhecidos do serviço (usando */etc/services*). É possível impedir essa tradução usando *-n* para os nomes de domínio e *-P* para os nomes de serviços.

Na listagem 1, vemos que os serviços ntpd, sshd, master e cupsd estão rodando. Os nomes indicam um servidor de sincronização de horário, um servidor de shell seguro SSH, o servidor principal do Postfix e o servidor de impressão CUPS. Se o endereço na última coluna é localhost (ou 127.0.0.1), o serviço está disponível apenas localmente e não a partir da Internet. Em outras palavras, a primeira e a última colunas da listagem são as mais interessantes para nós. O comando lsof não tem como saber se esses serviços estão protegidos por um firewall.

Serviços aceitando conexões

Entra em cena o nmap, capaz de detectar se existem serviços disponíveis e aceitando conexões em uma máquina remota. Uma porta aceitando conexões indica um serviço ativo. Ao contrário do lsof, o nmap não verifica se o serviço está escutando naquela porta, simplesmente verifica se a porta está aberta e indica o candidato mais provável. *(Nota do Editor: a versão 3.75 do nmap já faz, por fingerprinting, essa verificação. Verifique http://www.insecure.org/nmap)*. O comando a seguir testa todos os serviços que estão "escutando" a rede:

```
nmap -sS IP.da.máquina.alvo
```

A listagem 2 mostra que apenas a porta 22 (SSH) e o servidor Postfix (SMTP) na porta 25 estão abertos para acesso remoto. O servidor CUPS, mostrado na Listagem 1, não está acessível para conexões externas via TCP.

Mais serviços

Para verificar quais serviços UDP estão rodando na máquina, use a opção *-sU* em vez de *-sS*. A letra depois de *-s* indica o tipo de scan a ser disparado: *U* especifica uma varredura com pacotes UDP, enquanto *S* especifica uma varredura com pacotes TCP SYN.

Infelizmente o nmap não é capaz de determinar com exatidão o estado das portas UDP. Como o programa identifica silêncio (nenhuma resposta vinda das portas) como "escutando" (i.e. aceitando conexões), a situação pode ficar um tanto confusa: se a máquina sob teste estiver atrás de um firewall que bloqueie os pacotes UDP do nmap, o programa vai indicar que todas as portas UDP estão abertas!

Certifique-se de que todos os serviços não usados estão desligados. Os serviços são controlados por daemons normalmente iniciados durante o boot. Há duas maneiras completamente diferentes de se fazer isso: o serviço pode ter um script de inicialização no diretório */etc/init.d* ou entregar a tarefa ao superservidor inetd ou xinetd. O superservidor abre a porta durante o boot mas roda o serviço associado apenas se houver uma conexão a ela.

A técnica para desabilitar um serviço difere drasticamente de distribuição para distribuição. Além disso, há muitas soluções possíveis para o problema, mesmo dentro de uma mesma distribuição (por exemplo, o Red Hat usa as ferramentas *chkconfig*, *ntsysv* e *redhat-config-services* para essa tarefa). Consulte os artigos indicados em [5] e [6] para mais detalhes.

Se você não está familiarizado com o que um determinado serviço faz, pode sempre digitar

```
man nome_do_serviço
```

Listagem 2: Varredura com o nmap

```
# nmap -sS kermit

Starting nmap V. 3.00
(www.insecure.org/nmap/)
Interesting ports on kermit
(192.168.0.202):
(The 1594 ports scanned but not
shown below are in state: closed)
Port     State    Service
22/tcp   open     ssh
25/tcp   open     smtp

Nmap run completed - 1 IP address
(1 host up) scanned in 3 seconds
```

Listagem 3: Firewall simples

```
#!/bin/bash
iptables -F
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT DROP
iptables -A OUTPUT -m state ➥
--state NEW,ESTABLISHED,RELATED -j ➥
ACCEPT
iptables -A INPUT -m state --state ➥
ESTABLISHED,RELATED -j ACCEPT
iptables -A INPUT -j LOG ➥
--log-prefix "Firewall:"
```

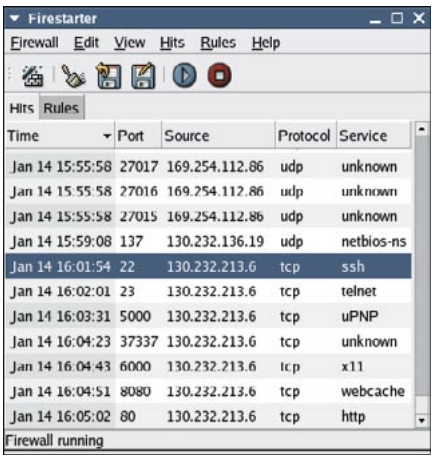

Figura 1: Firestarter mostrando pacotes "maliciosos", não permitidos pelo conjunto de regras do firewall. Antes disso, usuários devem rodar o assistente para criar as novas regras.

Listagem 4: Email gerado pelo Logwatch

```
Subject: LogWatch for kermit.spenneberg.de
Date: Tue, 27 Jan 2004 10:33:29 +0100 (CET)
From: root@kermit.spenneberg.de (root)
  ##################### LogWatch 4.3.1 (01/13/03) ####################
        Processing Initiated: Tue Jan 27 10:33:28 2004
        Date Range Processed: yesterday
        Detail Level of Output: 0
        Logfiles for Host: kermit.spenneberg.de
  ###################################################################
  ----------- pam_unix Begin -----------
su:
  Authentication Failures:
     spenneb(500) -> root: 1 Time(s)
xscreensaver:
  Unknown Entries:
     authentication failure; logname= uid=500 euid=500 tty=:0.0 ➥
ruser= rhost= user=spenneb 1 Time(s)
```

para descobrir se precisa realmente dele. Substitua *nome_do_serviço* pelo nome do comando mostrado pelo lsof. Infelizmente, o texto de algumas man pages não é lá muito claro.

Usuários que possuam manuais da distribuição podem procurar por mais informações sobre os serviços. Como complemento, há o Google para nos ajudar. Em último caso, sempre é possível desabilitar o serviço e ver o que acontece quando se reinicia a máquina. Há pouquíssimas situações em que essa ação torna o sistema instável.

Firewalls

Você realmente precisa ir até o fim do trabalho de preparação que vimos até aqui antes de sequer pensar em adicionar um firewall para proteger os serviços que devem ficar ativos. A listagem 3 mostra um script bem simples de firewall que permite apenas conexões de saída e as respostas a elas em um computador com Linux.

O comando iptbles é usado para manutenção das regras de filtragem de pacotes do kernel. O primeiro passo é apagar quaisquer regras que porventura existam; para isso, usamos a opção *-F*.

O próximo passo define as regras-padrão (policy) das três cadeias principais, INPUT, FORWARD e OUTPUT, com o alvo DROP. Isso configura o firewall para bloquear ("derrubar") qualquer pacote que não tenha sido explicitamente permitido. A cadeia INPUT é responsável por pacotes destinados ao próprio computador; a cadeia OUTPUT filtra pacotes que sejam originados no computador; finalmente, a cadeia FORWARD contém as regras de filtragem dos pacotes que são roteados dentro da máquina. A Listagem 3 não configura o Linux para agir como roteador, apenas como estação de trabalho normal.

Isso nos deixa com as cadeias INPUT e OUTPUT. Como o computador deve ser capaz de iniciar conexões com outros computadores, a cadeia OUTPUT deve permitir pacotes que originem conexões (NEW) e pacotes pertencentes a conexões já estabelecidas (ESTABLISHED). A palavra RELATED (relacionado) é necessária para algumas operações específicas, como por exemplo tráfego FTP e mensagens de erro.

Agora precisamos configurar a cadeia INPUT do firewall para aceitar a entrada de qualquer pacote pertencente a uma conexão já estabelecida (ESTABLISHED) ou pacotes com o estado RELATED. A última linha registra nos logs do sistema qualquer pacote bloqueado pelo firewall. Essas mensagens são marcadas, nos logs do sistema, pelo prefixo *Firewall:*.

A melhor maneira de ativar o firewall é rodando o script na inicialização da máquina. Cada distribuição tem sua maneira particular de fazer isso. Em muitos casos, simplesmente rode o script ou digite os comandos manualmente – você deve ter privilégios de superusuário (root) para tal. Para terminar, digite o comando

```
iptables-save
```

para salvar a configuração atual, que será lida no próximo reboot. Se não se sentir confortável com toda esses comandos, você provavelmente irá gostar do firestarter [8,9] (figura 1). O front-end gráfico é decepcionante, entretanto. Você deve, necessariamente, entender o funcionamento de firewalls em geral para usar a ferramenta de maneira adequada. O manual do firestarter e o site oficial do Iptables [7] são bons lugares para quem está começando.

Análise de relatórios

Um firewall é tão eficiente quanto os relatórios que gera. E um administrador só é considerado eficiente se ler esses relatórios (e também os logs criados pelos serviços rodando na máquina). Infelizmente (algumas pessoas diriam "Graças a Deus!") uma máquina Linux pode gerar diariamente vários megabytes de logs. O registro de eventos mais importante no Linux encontra-se em */var/log/messages*. Uma mensagem do firewall nesse arquivo se parece com esta:

```
Mar 19 10:12:15 kermit kernel:
Firewall: IN=eth0 OUT=MAC=00:20
:E0:6C:72:1E:00:50:56:C0:00:01:
08:00 SRC=130.232.213.6 DST=192
.168.0.1 LEN=84 TOS=0x00 PREC=
0x00 TTL=64 ID=0 DF PROTO=ICMP
TYPE=8 CODE=0 ID=29280 SEQ=1
```

A mensagem foi gerada pelo kernel em 19 de março às 10 horas, 12 minutos e 15 segundos na máquina cujo hostname é kermit. O prefixo *Firewall:* chama a nossa atenção para um pacote ICMP do tipo 8, código 0 - um "ping", ou *ICMP Echo Request* - que chegou à interface eth0. O endereço IP da máquina que enviou o pacote é 130.232.213.6.

É praticamente impossível ler e interpretar toda essa informação manualmente – pela dificuldade em ler as informações e pelo próprio tamanho dos arquivos. O ideal é usar ferramentas que automatizem a tarefa. Em geral, os administradores têm algumas opções de programas a escolher. Além das ferramentas específicas para um serviço em especial, há programas de uso geral para análise de logs, como o logwatch [10] e o logsurfer [11].

O logwatch é normalmente empacotado nas principais distribuições e vem completamente pré-configurado. É possível iniciar a ferramenta manualmente, simplesmente digitando o comando logwatch, que analisará seus logs e enviará um email ao administrador caso encontre algo suspeito. Muitas distribuições já têm o logwatch programado para rodar periodicamente pelo cron – procure pelo script correspondente (ou um link para ele) no diretório */etc/cron.daily*.

Mensagens que indiquem erros e tentativas de invasão permitem que o administrador faça análises mais profundas e específicas dos logs. Por exemplo, na Listagem 4 o logwatch alertou para tentativas fracassadas de login.

A listagem mostra que um usuário chamado *spenneb* não obteve sucesso em ganhar privilégios de usuário root usando o comando su, além de não informar a senha correta para o Xscreensaver.

Enquanto o logwatch monitora os logs em intervalos regulares, o logsurfer faz um monitoramento contínuo, alertando o administrador quanto a potenciais tentativas de ataque e invasão. Entretanto, o logsurfer não é simples de configurar e as complexidades inerentes a ele estão além do escopo deste artigo. Se você quiser saber mais sobre o logsurfer, visite seu site oficial em [11].

Analisando as mensagens

O FWLogwatch [12] analisa as mensagens do firewall. O pacote RPM com o programa pode ser encontrado no site oficial. Algumas distribuições incluem o FWLogwatch em seus CDs de instalação.

O FWLogwatch possui três modos (log summary mode, interactive report mode e realtime response mode) selecionáveis pelo usuário no arquivo de configuração. No modo log summary, a ferramenta produz um apanhado geral de centenas de mensagens de log em questão de segundos. Com ela, os administradores podem ter uma idéia de como o firewall se comportou nas últimas horas e aponta os endereços de possíveis atacantes.

No modo *interactive report*, o FWLogwatch envia automaticamente um email ao administrador da rede de origem do ataque. Cada um dos ataques detectados gera um email diferente. Tipicamente, um administrador não é responsável pelos ataques originados de sua própria rede; pelo contrário, é vítima de um invasor externo, que usa redes alheias como “mulas”. Portanto, use com muito cuidado esse modo de operação, pois você pode estar acusando a pessoa errada.

O modo realtime response é o mais interessante dos três. Nesse modo, o FWLogwatch monitora os arquivos de log em tempo real e alerta por email o administrador da máquina caso um ataque tenha sido detectado. Para que o FWLogwatch funcione nesse modo, o administrador tem de definir valores-limite e configurar a ferramenta para disparar quando o limite for ultrapassado. A Listagem 5 mostra um exemplo de configuração para */etc/fwlogwatch.config*.

A linha *realtime_response = yes* ativa o modo realtime response. Como o FWLogwatch pode analisar mensagens de inúmeros tipos de firewall, configure a linha *parser = n* para selecionar o modo Netfilter/IPtables mode.

A linha *run_as = fwloguser* configura o FWLogwatch para ser executado com uma conta não privilegiada exclusiva. É óbvio que o usuário fwloguser deve existir no sistema: certifique-se de que a conta foi criada depois da instalação

Glossário

Debian: *distribuição GNU/Linux livre, completa, independente e não-comercial, compilada por voluntários. (http://www.debian.org/). O Debian é popular entre os gurus do mundo Linux e também entre os que querem entrar nesse time.*

Patch: *Como o Linux e os programas que nele rodam são livres e possuem o código fonte aberto, as modificações e correções de falhas podem ser distribuídas na forma de patches (palavra inglesa que significa "remendo"). Um patch contém as linhas modificadas do código. O comando* diff *é usado para criar patches (e por isso são freqüentemente chamados de arquivos diff). Para tanto, o comando* diff *compara o código original e o modificado, gerando um arquivo apenas com as diferenças. Os usuários podem usar o comando* patch *para aplicar as correções no código fonte e, para completar a atualização, recompilar o código.*

Cron: *Um serviço que executa tarefas (ou seja, programas e scripts) agendadas. O cron verifica seus arquivos de configuração a cada minuto. Tanto os usuários comuns quanto o root podem criar suas próprias tabelas cron para agendar a execução de seus programas [4].*

Porta: *tipicamente, múltiplos serviços TCP e UDP rodam simultaneamente em máquinas Linux. Por isso, deve haver uma maneira de identificá-los, e para isso existem as portas. Cada programa (seja ele cliente ou servidor) comunica-se por meio de uma porta exclusiva. Cada número de porta corresponde a um serviço e esses números são normatizados: consulte o arquivo /etc/services. Uma rápida olhada nesse arquivo nos diz que, por exemplo, o serviço SSH usa a porta 22.*

Serviços TCP: *serviços de rede que usam o protocolo TCP [13] para comunicação. Isso inclui a maioria dos serviços de Internet pois o TCP oferece um meio de transporte confiável, assegurando a entrega dos dados na ordem correta. Tomar essa tarefa para si torna o protocolo TCP um pouco mais pesado, com um cabeçalho bem maior: uma de suas desvantagens. Exemplos de serviços que usam TCP são servidores Web, servidores de email, FTP, telnet, entre muitos.*

Serviços UDP: *serviços de rede que usam o protocolo UDP para comunicação. O UDP não garante a entrega dos pacotes na ordem correta, ou sequer se eles serão entregues. O UDP é usado naquelas aplicações em que os dados são todos entregues num único pacote, tornando a ordem em que chegam irrelevante. Se um programa não receber resposta depois de um determinado período, ele solicita a retransmissão dos dados. Como o UDP não precisa cuidar desses pormenores, o tamanho de seu cabeçalho é relativamente pequeno. Serviços que usam UDP incluem DNS, servidores de sincronização de horário e difusão (streaming) de áudio e vídeo.*

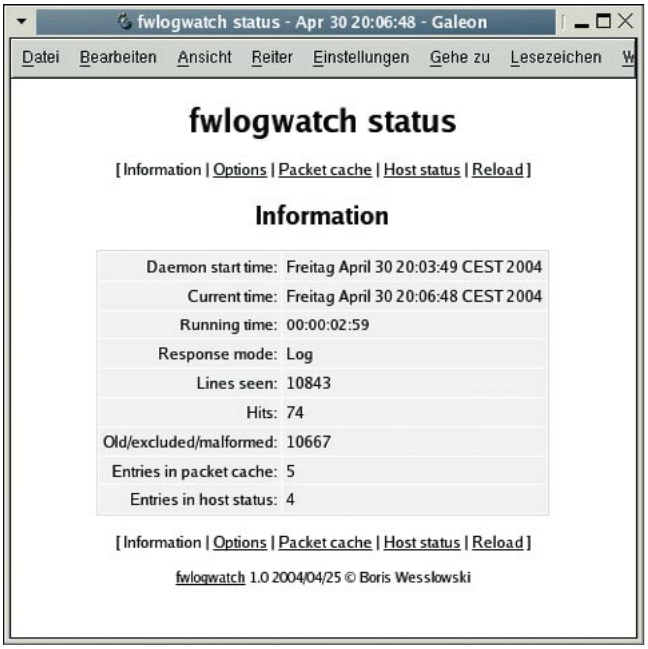

Figura 2:O FWLogWatch tem seu próprio servidor web, que permite ao administrador verificar o status do firewall.

da ferramenta. Esse usuário precisa ter permissões de leitura do arquivo */var/log/messages*.

A linha *alert_threshold* especifica o número de eventos que disparam um alerta, *notify = yes* e *notification_script = ?* configuram a ferramenta para usar o script especificado (*/usr/sbin/fwlw_notify* em nosso exemplo) para enviar o alerta ao administrador.

Em modo realtime response, o FWLogwatch prepara um servidor web (*server_status = yes*), permitindo que o administrador verifique o estado atual do firewall pelo endereço 192.168.0.1 (*bind_to*) e a porta 8888 (*listen_port*) (veja a figura 2). Para chegar a esse relatório, os usuários têm de, em seus browsers, navegar até *http://192.168.0.1:8888*, logar-se com o usuário ralf (*status_user*) e informar a senha apropriada (*status_password*). O comando *htpasswd -n ralf* cria a senha.

O melhor a fazer é iniciar o FWLogwatch durante o boot do sistema. Usuários do Red Hat Linux podem digitar o comando a seguir:

```
chkconfig fwlogwatch on
```

Também é possível iniciar o FWLogwatch manualmente com o comando:

```
/etc/init.d/fwlogwatch start
```

Todas essas medidas preventivas são úteis contra ataques externos. Muitos ataques podem ser causados por cavalos de Tróia ou vírus – casos em que os atacantes contam com a ajuda dos usuários desavisados para se infiltrar na rede. Os invasores podem ludibriar os usuários e fazê-los, por exemplo, visitar um site interessante, onde oferecem um programa de características aparentemente maravilhosas. Enquanto o software instala, silenciosamente um cavalo de Tróia é configurado e abre uma conexão para a máquina do invasor.

Felizmente em máquinas Linux é preciso possuir poderes de superusuário (root) para instalar software. Para minimizar o perigo, os administradores devem instalar programas apenas de fontes seguras e não usar a conta root para navegar na Internet.

No fim das contas, nenhum sistema, por mais atualizado, pode protegê-lo de todos os ataques. Mesmo os administradores mais proativos podem ser vítimas de invasões. Continuaremos esta série de artigos discutindo as melhores maneiras de rastrear e sair à caça de possíveis invasores, bem como novos métodos de resposta a incidentes. ■

Listagem 5: Configuração do FWLogwatch

```
realtime_response = yes
parser = n
run_as = fwloguser
alert_threshold = 5
notify = yes
notification_script = /usr/sbin/
fwlw_notify
server_status = yes
bind_to = 192.168.0.1
listen_port = 8888
status_user = ralf
status_password = gie0lzYkkk9sQ
refresh = 10
```


SOBRE O AUTOR

Ralf Spenneberg é instrutor e autor independente de Unix e Linux. Publicou três livros, "Intrusion Detection Systems for Linux Servers", "VPN for Linux" e "Intrusion Detection und Prevention mit Snort & Co.". Ralf também desenvolveu material didático variado e oferece treinamento em inglês e alemão.

INFORMAÇÕES

[1] Invasão nos servidores Debian: *http://www.debian.org/News/2003/20031121*

[2] Lista de discussão Bugtraq: *http://www.securityfocus.com/archive/1*

[3] Lista de discussão Full-Disclosure: *http://lists.netsys.com/mailman/listinfo/full-disclosure*

[4] Introdução ao Cron: Jürgen Jentsch, "Count Down - Program scheduling with Cron", Linux Magazine, Edição 22, julho e agosto de 2002. *http://www.linuxmagazine.com/issue/22/cron.pdf*

[5] Inicialização de serviços: Marc André Selig, "Ready, Steady, Go - Launching services at boot time:", Linux Magazine, Edição 33, agosto de 2003. *http://www.linuxmagazine.com/issue/27/Initialization.pdf*

[6] Segurança de sistemas: Anthony Stone, "Hardening your system - Operating System Hardening", Linux Magazine, Edição 33, agosto de 2003. *http://www.linuxmagazine.com/issue/33/Operating_System_Hardening.pdf*

[7] Netfilter/IPTables: *http://www.netfilter.org/*

[8] Front-ends para IPtables: Nico Lumma, "Building Firewalls - Front-ends for IPtables", Linux Magazine, Edição 34, September 2003. *http://www.linuxmagazine.com/issue/34/IPtables_Firewalling.pdf*

[9] Firestarter: *http://firestarter.sf.net/*

[10] Logwatch: *http://www.logwatch.org/*

[11] Logsurfer: *http://www.cert.dfn.de/eng/logsurf/*

[12] FWLogwatch: *http://cert.uni-stuttgart.de/projects/fwlogwatch/*

[13] Nico Lumma, "Tecendo a Rede", Linux Magazine Brasil, Edição 1, Agosto de 2004, p20