Documentação ABC
Índice geral

Informações gerais

Foi instalada a distribuição Debian 3.0 com suporte a Raid, LVM e IPSEC, com kernel 2.6.8.1.

No momento este servidor tem as funções de firewall, servidor de VPN, geração de gráficos de tráfego e controle de banda

Parâmetro	
Hostname	abc-fw
IP Inet - eth0	200.2xy.162.228
IPS DMZ - eth1	200.2xy.163.1 200.2xy.163.6 200.2xy.163.9 200.2xy.163.11 200.2xy.163.13
IP Lan - eth2	10.1.1xy.250
IP Lan 3 – eth3	10.3.xy.1
Gateway	200.2xy.162.225
DNS	127.x.y.1

Estas configurações ficam nos arquivos:

```
/etc/network/interfaces
/etc/resolv.conf
/etc/hostname
```

Para alterá-las edite os arquivos, e reinicialize a rede com o comando:

```
# /etc/init.d/networking restart
```

Configuração de Firewall

A configuração das regras de firewall é feita através do software Firewall Builder.

As regras ficam armazenadas em um arquivo .xml este arquivo após passar por um processo de compilação gera um script .fw que é executado para se carregar as regras. Estes arquivos estão armazenados no diretório

```
/etc/network
```

Para que estes arquivos sejam guardados neste local o script usado pelo Firewall Builder para instalação das regras foi alterado, este script é o

```
/usr/bin/fwb_install
```

A alteração feita foi:

```
60c60
< REMOTEDIR="/etc/network"
---
> REMOTEDIR="/etc/firewall"
```

Procedimento de conversão de regras

Antes da instalação das regras no Firewall Builder foi necessário convertê-las do formato usado anteriormente, além de adapta-las para a nova topologia.

Este processo seguiu as seguintes diretrizes:

1. Transformar a rede entre os Firewalls 1 e 2, conhecida como ENTRENET, para DMZ.
2. Padronizar o nome das variáveis, usando _ como separador de nomes e no caso de variáveis com mais de um IP, adicionar um número ao final da variável que deve ser incrementado em 1 para cada IP.
3. Como no Firewall Builder as regras são separadas por interfaces de rede, as regras foram agrupadas de acordo com a interface de entrada delas no Firewall.

Medição de Tráfego

A medição de tráfego é feita através do software ipac-ng, as regras para a medição estão em:

```
/etc/ipac-ng/rules.conf
```

A coleta dos dados é feita com uma periodicidade de 5 minutos, isto significa que o intervalo mínimo que se pode obter uma medição é de 5 minutos.

Hoje os seguintes tráfegos são medidos:

Entrada e saída de http, https, pop3, smtp, notes, ftp e ftp-data entre a Internet e as diversas redes da ABC, ainda é medido o total de entrada e saída.

Total de tráfego das VPNs (protocolo ESP), também é contabilizado.

Configuração das VPNs

O novo Firewall utiliza os módulos de IPSEC padrões do kernel, o que torna desnecessária a aplicação de patchs. Os módulos são os seguintes: NET_KEY, INET_AH, INET_ESP, XFRM_USER, CRYPTO, CRYPTO_HMAC, CRYPTO_NULL, CRYPTO_MD5, CRYPTO_SHA1, CRYPTO_DES e CRYPTO_AES.

As VPNs utilizam o software racoon para a negociação inicial, o racoon é configurado através da racoon-tool o seu arquivo de configuração é o

```
/etc/racoon/racoon-tool.conf
```

Abaixo segue um exemplo de configuração

```
      connection(Morumbi):
          dst_range: 192.1xy.200.0/24
          src_range: 10.x.y.0/8
          dst_ip: 200.1xy.92.147
          src_ip: 200.2xy.162.228
          admin_status: enabled
          lifetime: time 8 hours
          authentication_algorithm: hmac_sha1
          encryption_algorithm: 3des
          compression: disabled


       peer(200.1xy.92.147):
          verify_cert: on
          passive: off
          verify_identifier: off
          lifetime: time 8 hours
          hash_algorithm[0]: sha1
          encryption_algorithm[0]: 3des
```

A autenticação é feita através de PSK, Pre Shared Key, e estas chaves ficam armazenadas no arquivo.

```
/etc/racoon/psk.txt
```

Para gerar uma nova chave sugerimos os seguinte comando:

```
apg -a 1 -n 1 -m 29 -x 29 -M lcn
```

O arquivo psk.txt, tem o seguinte formato:

```
200.2xy.8.67          u2ydN0quUxrctBDtPdtwDz72GbD70
200.1xy.92.147        Hi6dSWu8eqp64l4HTkZ5TC1Yuapuf
200.1xy.228.92        MXvSpkEyi1BmtaIrV1QwsOn24tPto
```

Nas lojas o software freeswan foi mantido e reconfigurado conforme o exemplo abaixo.

```
conn vpn-morumbi-ABC
          left=200.1xy.92.147
          leftsubnet=192.1xy.200.0/24
          leftnexthop=200.2xy.128.1
          right=200.2xy.162.228
          rightsubnet=10.0.0.0/8
          #rightnexthop=200.2xy.162.225
          auto=start
          type=tunnel
          keyexchange=ike
          auth=esp
          authby=secret
          pfs=yes
          keylife=8h
          rekey=yes
          rekeymargin=9m
          keyingtries=0
          ikelifetime=8h
```

Nas lojas a chave é guardada no arquivo

```
/etc/ipsec.secrets
```

No seguinte formato:

```
200.2xy.128.27 200.2xy.162.228 : TSU "ZTYER^#/H%tDr82K|DBMn,#@*j)\?'
```

Sistema para alta disponibilidade

Conceitos básicos

Foram instalados dois servidores com a função de firewall, com configuração idêntica um ao outro. A instalação foi projetada para continuar operando em caso de falha crítica de hardware em um dos equipamentos, e em alguns casos de falha de software. Para que isso seja possível, um dos equipamentos trabalha em modo "principal", como firewall, enquanto o outro fica apenas em modo "reserva", permanentemente verificando se o "principal" está operando corretamente. Além disso, o disco do servidor principal é replicado para o servidor reserva; assim, em caso de ativação do servidor reserva na função de "principal", seus dados e configurações estarão atualizados.

O sistema é composto de três partes principais: replicação de dados, detecção de falha no servidor principal, e escolha de servidor principal/reserva (no momento do boot). Também será explicada a manutenção do sistema.

Replicação de dados

A replicação de dados é feita pelo driver "drbd" do linux. A partição raiz do firewall foi configurada para ser montada num dispositivo de bloco drbd. Com isso, todos os dados escritos nessa partição no servidor principal são replicados no servidor reserva.

A configuração do drbd é feita pelo arquivo /etc/drbd.conf. Nesta as partições hda1 de ambas as máquinas são associadas ao dispositivo em /dev/drbd0.

Foi utilizada uma partição dedicada a meta-dados, em hda3. O particionamento foi feito de tal modo que a partição raiz fica no início do disco, seguida pela de swap, e na sequência fica espaço não-alocado. A partição de meta-dados fica ao final do disco.

A situação da replicação pode ser monitorada com

```
abc-fw:/root# cat /proc/drbd
version: 0.7.7 (api:77/proto:74)
SVN Revision: 1680 build by root@abc-fw, 2004-12-16 15:43:13
 0: cs:Connected st:Primary/Secondary ld:Consistent
    ns:6207160 nr:0 dw:2273332 dr:5030953 al:177 bm:719 lo:0 pe:0
ua:0 ap:0
```

O status do dispositivo drbd0 aparece na linha iniciada por "0". No exemplo acima ele está conectado e ativo. Outras situações podem ser esclarecidas na documentação do drbd.

A replicação é feita por uma conexão Ethernet dedicada, feita com um cabo cross-over entre as duas interfaces eth5 dos servidores.

Detecção de falha

O servidor "reserva" fica permanentemente executando o comando:

```
fping -r<NR_PING> -a <IP1> <IP2>
```

O comando fping é executado por um script shell, localizado em /usr/local/sbin/sec. Este script fica em loop executando fping e atualizando o console, até que o fping indique falha. Quando há indicação de falha o script shell termina, e o servidor, que estava em modo reserva, continua o boot e entra em modo principal.

Este comando verifica a conectividade, com ICMP echo, para os IPs de LAN e de cross-over do servidor "principal". Os IPs são especificados em /etc/drbd-cluster/cluster.conf como PR_IP_CROSS e PR_IP_LAN. O parâmetro NR_PING também é definido no mesmo arquivo. Durante os testes de ativação configuramos para 10, o que significa que o servidor principal deverá parar de responder aos pings por aproximadamente 60 segundos antes que o servidor reserva assuma o papel de principal.

Ambos os IPs (LAN e CROSS) deverão estar em falha para que o servidor reserva assuma. Caso somente um deles falhe, o servidor reserva indica a situação em seu console, mas continua operando como reserva (pois isso pode significar uma falha em uma das placas de rede do próprio servidor reserva, ou em sua conectividade).

Escolha de modo de operação quando do boot

A decisão de operar em modo principal ou reserva deve ser feita quando do boot do equipamento. Esta decisão não pode ser feita simplesmente com o mesmo critério da detecção de falha, pois pode ocorrer dos dois equipamentos estarem sendo ligados simultaneamente, e nesse caso estarão testando se existe um nó ativo ao mesmo tempo. Nesse caso ambos assumiriam o papel de "principal", gerando problemas na rede.

Para fazer a escolha o sistema segue o seguinte algoritmo:

1. Se foi escolhido, no menu de boot do LILO, a opção "Principal", o sistema entra em modo principal sem fazer testes.
2. Se foi escolhido no LILO, "Reserva", o sistema entra em modo reserva sem fazer testes. Adicionalmente o sistema força uma ressincronização completa do disco do servidor reserva, com o comando "drbdadm invalidate all" (veja documentação do drbd).
3. Se foi escolhido, no LILO, a opção "Autodetect", verifica, com o comando fping, se há um servidor em modo principal. Se detectar um, entra em modo reserva de imediato. Este comando "fping" não usa a variável NR_PING, e portanto usa o timeout padrão do fping, que é de pouco mais de 3 segundos (com 3 pacotes ICMP).
4. Se nenhum dos IPs configurados para teste responder, o servidor reserva monitora a interface CROSS por um período configurado (variável ESCUTA_SILENCIOSA no /etc/drbd-cluster/cluster.conf – deixado como 60 segundos). Caso identifique algum tráfego nesse período, assume modo reserva.
5. Se nenhum tráfego for identificado, o servidor envia um pacote pela eth5, ao mesmo tempo que continua monitorando a mesma eth5. Verifica então se o pacote detectado

tem o seu próprio mac-address como origem. Se tiver, assume modo principal. Caso contrário, assume modo reserva.

A idéia por trás desse algoritmo é a seguinte: a "escuta silenciosa" deve ser suficiente para que o outro servidor, se tiver decidido entrar em modo ativo antes de começarmos a monitorar a eth5, gere tráfego (vai gerar quando configurar o drbd, pois este tenta se conectar no servidor reserva).

Caso ambos estejam fazendo "escuta silenciosa" simultaneamente, o primeiro que enviar tráfego assume o papel de principal. Para que isso funcione melhor, a eth5 deve ser configurada como half-duplex, evitando que cada servidor veja o seu pacote transmitido em antes do transmitido pelo outro.

O algoritmo acima é implementado pelo script /usr/local/sbin/setup.

Arquivos de configuração e scripts usados[1]

Diretório	*arquivo*	*função*
/usr/local/sbin	setup*	Executado quando do boot do servidor. Decide se entra em modo principal ou reserva.
	sec*	Executado quando servidor decide entrar em modo reserva. Quando esse script termina, o servidor continua o boot e entra em modo principal.
	rebootd*	Script executado de dentro do ramdisk, quando em modo reserva. Responsável por aceitar comandos do nó principal para reinicializar o reserva.
	reboot-reserva	Script que pode ser executado no principal para forçar uma reinicialização do reserva.
/etc	drbd.conf*	Configuração do software drbd (partição, parâmetros de replicação, Ips usados para replicar, etc).
	lilo.conf	Configuração do bootloader. A secção "disk=/dev/hda" é necessária para que o lilo saiba como criar mapas de boot no dispositivo drbd0. Nesta seção o drbd0 é definido como uma partição que inicia no setor 63 do hda. O valor "63" deve ser obtido na tabela de partição real do hda, e deve ser o início da partição correspondente ao drbd0 (no caso, hda1). As opções append="drbd-cluster=reserva" e append="drbd-cluster=principal" são usadas para informar ao script "setup" que desejamos forçar um determinado modo de operação.

1 Os arquivos/scripts que são copiados para o ramdisk estão assinalados com um *.

Diretório	*arquivo*	*função*
/etc/drbd-cluster	cluster.conf*	Define parâmetros dos algoritmos de escolha de modo de operação e detecção de falha. Define também as interfaces Ethernet, IPs e máscaras usados em modo reserva e principal para comunicação entre os dois servidores. Note que os IPs do modo principal devem *adicionalmente* ser corretamente definidos em /etc/network/interfaces. Define o hostname do servidor quando em modo reserva (é necessário que essa definição esteja de acordo com o que for configurado no drbd.conf, para que o drbd funcione corretamente).
	inetd-secondary.conf*	É a versão do inetd.conf que é copiada para o ramdisk. A definição de escuta na porta 24 não é mais utilizada.
/etc/mkinitrd[2]	exe	Lista de arquivos executáveis copiados para o ramdisk.
	files	Lista de arquivos comuns copiados para o ramdisk.
	modules	Lista de módulos que precisam ser carregados antes de montar a partição raiz definitiva. Neste arquivo é definida a carga dos drivers Ethernet, e aqui foi configurado o modo half-duplex para a eth5. Note que os dois equipamentos não são exatamente iguais, pois um possui uma Realtek e outro uma Intel como eth5, mas essa diferença fica transparente por tentarmos, no randisk, carregar ambos os drivers.
	mkinitrd.conf	Configuração central do mkinitrd.
	scripts/drbd-cluster	Script usado para copiar o "setup" para o diretório correto dentro do ramdisk.
/usr/share/initrd-tools/probe.d[3]	drbd	Esse fragmento de shell script é executado pelo mkinitrd para que o mkinitrd aprenda como lidar com partições raiz em /dev/drbdX. Foi incluído manualmente no sistema.
/usr/src	diversos	Arquivos .deb e outros usados na compilação do kernel em uso pelo firewall.

Sequência de boot

O servidor carrega um ramdisk, e os scripts "setup" e "sec" são executados com esse ramdisk como partição raiz.

O script "setup" é responsável pela escolha de modo de operação, no boot. Caso escolha modo reserva, o script "sec" é acionado, mantendo o ramdisk como partição raiz e executando a replicação. Quando o script "sec" termina de executar, o servidor ativa o drbd como primary e continua o boot,

2 Verifique documentação do mkinitrd para mais detalhes sobre este diretório
3 Este diretório foi criado manualmente, mas o script mkinitrd original do Debian o acessa sem modificações.

entrando em modo principal.

Caso o ramdisk seja danificado, pode ser escolhida a opção de boot "Sem-ramdisk". Neste caso o servidor iniciará em modo principal, mas não fará replicação, pois a partição raiz será /dev/hda1 e não /dev/drbd0. Se esta opção for utilizada é importante forçar uma replicação completa, posteriormente, pois a escrita direta no hda1 impede que o drbd detecte quais blocos foram alterados. Para mais detalhes consulte a documentação do drbd.

Administração do cluster

O sistema foi projetado para minimizar o esforço de administração. Com esse objetivo *todos* os dados do servidor principal são replicados no reserva, incluindo mapa de boot e kernel. Do ponto de vista da administração rotineira de um sistema Linux, o sistema deve ser encarado como um único servidor.

Os servidores têm exatamente a mesma instalação (e, caso um deles tenha sua instalação alterada, esta é replicada, mantendo a verdade da afirmação acima). Dois aspectos, entretanto, não são replicados: tabela de partição e MBR. Caso seja alterado o particionamento do servidor principal, a alteração deverá ser feita manualmente no servidor reserva. O mesmo se aplica a alterações no MBR.

Devem ser tomadas algumas precauções simples em caso de atualização do mapa de boot e seus arquivos – kernel e ramdisk.

O modo reserva permite que seja feito login, com a mesma senha de root do modo principal, usando Alt-F2. Note que este prompt possui muito poucos comandos disponíveis, por estar sendo executado do ramdisk. Os comandos mais úteis são "ip", "nc" e "drbdadm". Consulte suas respectivas páginas de manual.

Atualização de kernel e ramdisk

Caso seja necessário trocar o kernel, ou refazer o arquivo de ramdisk, é importante que o comando lilo seja executado novamente. Isso é válido para qualquer instalação de linux com o bootloader LILO. No entanto, no caso do cluster, deve-se assegurar que as alterações de disco gravadas pelo comando LILO (bem como os blocos alterados quando da criação do novo ramdisk ou kernel) sejam propagadas para o servidor reserva *de forma atômica*. Em outras palavras, *não desligue o servidor reserva pouco tempo após a alteração de kernel, ramdisk ou execução do comando lilo no servidor principal*. Se isso for feito o servidor reserva pode perder a capacidade de dar boot, e precisará ser reparado com uma mídia de boot externa (CD ou disquete). Por "pouco tempo" deve-ser entender aprox. 60 segundos. A execução do comando "sync" no servidor principal reduz esse tempo significativamente, especialmente se não houver carga de I/O no servidor principal no momento (para aprox. 2 segundos, tipicamente).

A criação do ramdisk é feita com o comando:

```
mkinitrd -o /boot/initrd.img-2.6.8.1 2.6.8.1
```

Este comando deve ser seguido pelo comando "lilo".

Note que diversos arquivos de configuração, scripts e executáveis são copiados para o ramdisk quando da execução do mkinitrd. Caso, por exemplo, algum parâmetro do drbd ou dos algoritmos de detecção seja alterado, é necessário recriar o ramdisk e re-executar “lilo” para que o servidor reserva receba as atualizações. Para que o servidor reserva *execute* com as alterações, é necessário um reboot do mesmo. O script /usr/local/sbin/reboot-reserva pode ser usado para reinicializar o servidor reserva (ele está associado ao /usr/local/sbin/rebootd, que é executado no servidor reserva).

Referências na Internet

http://www.debian.org (boa parte traduzida para português)

http://www.fwbuilder.org/

http://www.ipsec-howto.org/t1.html

http://ipsec-tools.sourceforge.net/