Alta disponibilidade para VPNs

Caminhos alternativos

O IPSec impede que muitos dos truques espertos oferecidos pelos produtos de alta disponibilidade funcionem. Mostraremos este mês uma solução que pode ser usada como um caminho alternativo para conexões IPSec.

POR JOHANNES HUBERTZ

Os administradores de sistema insistem em usar conexões de rede que possuam um segundo caminho de "backup", que entra em cena automaticamente caso a conexão principal caia. Mas se usarmos uma VPN com IPsec para proteger o tráfego que deve passar pela Internet, a conexão reserva vai precisar de alguns cuidados especiais.

A razão de ser desses cuidados é o fato de o IPsec [1] precisar de um endereço IP consistente e imutável em ambos os pontos de entrada da VPN. Por isso, se a conexão for comutada para um túnel diferente, o endereço IP deve ser reconfigurado nos dois novos pontos de acesso – caso contrário, qualquer conexão existente será assassinada. O *Border Gateway Protocol* (BGP ou protocolo de *gateways* de fronteira [2]) oferece uma maneira eficaz e confiável de administrar um conjunto de números IP fornecidos por diversos provedores. Infelizmente, a maioria dos contratos de serviço de conexão à Internet proíbe terminantemente o uso do BGP em seus links.

Como paliativo, muitos administradores fazem tudo "à mão", sem qualquer automação – quando o ruim vira péssimo, eles trocam manualmente os cabos no painel de conexões para que o sistema use a conexão auxiliar. Não é exatamente o que podemos chamar de tecnologia de ponta em plena "Era da Informação". Em vez disso, seria preferível que os dispositivos de rede detectassem as falhas em uma

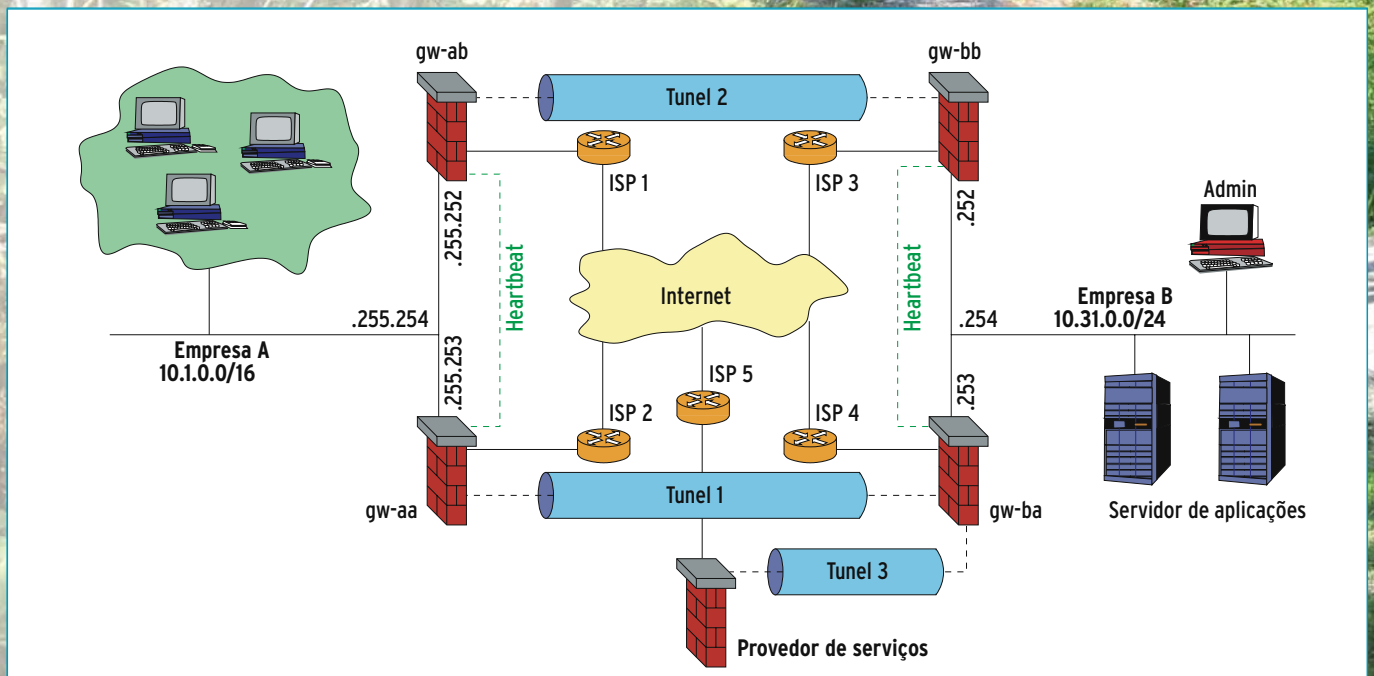

Figura 1: Nesta ilustração, a rede de um cliente está conectada a um prestador de serviços terceirizado por meio de uma VPN. A conexão usa dois caminhos alternativos – se necessário, o Túnel 2 toma o lugar do Túnel 1. Há ainda um provedor de serviços de TI conectado, mas sem alta disponibilidade.

conexão e fizessem automaticamente a comutação para a linha auxiliar. O sistema ideal poderia também gerar automaticamente as configurações para ambos os pontos de acesso, armazenando-as em um local central.

Para gateways que funcionam como firewalls e usam IPsec, a configuração central é o que podemos chamar de tecnologia de ponta. No Linux, o SSPE (*Simple Security Policy Editor* ou editor simplificado de políticas de segurança [3]) pode cuidar dessa tarefa. Entretanto, a solução de alta disponibilidade (ou HA, de *High-Availability*) ainda não é compatível com SSPE.

Linux-HA

O projeto Linux-HA [4] é dedicado a soluções de alta disponibilidade em servidores Linux. Esse software permite que os administradores montem uma VPN de alta disponibilidade que comute automaticamente e rapidamente da conexão principal para a de reserva – e tudo isso sem usar o BGP. Para implementar essa solução, você precisa de dois túneis paralelos e independentes. Um deles sempre será usado em um determinado momento. Cada túnel, em cada rede, possui seu próprio ponto de acesso, que serve como *gateway* para a rede local. O Linux-HA implementa a reconfiguração automática de endereços IP e pode, com isso, auxiliar nesse "pepino".

Ambos os nós HA possuem um endereço IP individual e um endereço IP compartilhado, que é usado por apenas uma das máquinas em um dado momento. O mecanismo pode "hospedar" servidores (Web, Mail ou mesmo o bom e velho Doom) que se conectam através do IP compartilhado. O serviço roda em ambas as máquinas e "escuta" em cada endereço IP. Entretanto, as solicitações de conexão vindas de outras máquinas chegam apenas pelo IP compartilhado. Isso permite que o Linux-HA atribua o endereço externo a uma segunda máquina em caso de emergência; os usuários não perceberão nada – nem que a máquina principal caiu, nem que o endereço externo mudou, já que para eles o endereço que vale é o compartilhado.

Um cabo serial assegura que o *heartbeat* ("batimento cardíaco") das duas máquinas seja monitorado. O *heartbeat* é uma parte importante do subsistema de alta disponibilidade Linux-HA. Os computadores envolvidos verificam, periodicamente, a disponibilidade de seus parceiros. Se um computador entrar em parafuso (ou se alguém chutar o cabo de rede) o computador parceiro "adota" o endereço IP pertencente ao primeiro computador. Isso gera *broadcasts* via

protocolo ARP usando o endereço IP compartilhado e o endereço MAC da *segunda* máquina, a que assumiu o controle – pense no procedimento todo como se fosse um ataque de *ARP-Spoofing* ou *ARP-Poisoning*, só que usado com um propósito legal (quem diria que tecnologia inventada por bandidos seria empregada por nós, que estamos "do lado claro da força").

Além do APR-Spoofing, a máquina define um *alias* (apelido) para a interface de rede. Quando o primeiro nó (o que caiu) voltar à vida (ou seja, seu "coração" voltar a bater) o protocolo *heartbeat* assegura que o segundo servidor desabilitará o alias da interface. Enquanto isso, o servidor principal levanta sua placa de rede e faz um novo *broadcast*, via protocolo ARP, para reassumir o controle sobre aquele endereço IP.

Listagem 1: Supervisor VPN para alta disponibilidade

```
01 #!/bin/bash
02 # Supervisor HA VPN no gateway gw-aa
03
04 # A outra extremidade do túnel
05 TARGET="gw-ba"
06
07 # Número de segundos entre "pings"
08 TIMEOUT=1
09
10 # Espere até que o tempo decorrido seja
11 # MAXFAIL * TIMEOUT antes de habilitar a troca de túneis
12 MAXFAIL=5
13
14 # Espere até que o tempo decorrido seja
15 # HYSTERE * TIMEOUT depois que a máquina principal
16 # entre novamente em operação para, só então, colocar
17 # os túneis na condição inicial
18 HYSTERE=180
19
20 # Considera sem erros o início de operação
21 FAIL=0
22
23 VERBOSE=""
24
25 ACTION_FAIL_START="/root/bin/HA-VPN-action-script start"
26 ACTION_OK_AGAIN="/root/bin/HA-VPN-action-script stop"
27
28 PING=/usr/bin/echoping
29 LOG="/usr/bin/logger -t HA-VPN"
30
31 math () {
32    eval echo "\$(($*))"
33 }
34
35 echo "`date +%Y%m%d%H%M%S` `basename $0` iniciando" | $LOG
36
37 while :
38 do
39 VAL=`$PING ${VERBOSE} -u -t $TIMEOUT -s 5 ${TARGET} 2>&1`
40 ERROR=$?
41 if [ $ERROR -gt 0 ] ; then
42     echo "$DAT $ERROR $FAIL $VAL" | $LOG
43     # Ocorreu um evento de expiração de tempo
44    if [ $FAIL -lt 0 ] ; then
45       # Outro erro durante a fase de recuperação
46       FAIL=`math $MAXFAIL + 1`
47    fi
48    if [ $FAIL -eq $MAXFAIL ] ; then
49       # Iniciar troca para o Túnel 2
50       :
51       FAIL=`math $FAIL + 1`
52       echo "$DAT iniciando troca de túneis: ${ACTION_FAIL_START}" | $LOG
53       ${ACTION_FAIL_START}
54       else
55       if [ $FAIL -lt $MAXFAIL ] ; then
56          FAIL=`math $FAIL + 1`
57       fi
58    fi
59 else
60     # "Ping" detectado com sucesso
61     if [ $FAIL -gt $MAXFAIL ] ; then
62       FAIL=`math 0 - $HYSTERE `
63     fi
64     if [ $FAIL -le $MAXFAIL -a $FAIL -ge 0 ] ; then
65       FAIL=0
66     fi
67     if [ $FAIL -lt 0 ] ; then
68    # Espera pelo período de "histerese" antes de voltar ao Túnel 1
69       echo "$DAT $ERROR $FAIL $VAL" | $LOG
70       FAIL=`math $FAIL + 1`
71       if [ $FAIL -eq 0 ] ; then
72          # Restaura operação normal
73          :
74          echo "$DAT novamente em operação normal: ${ACTION_OK_AGAIN}" | $LOG
75          ${ACTION_OK_AGAIN}
76       fi
77     fi
78 fi
79 #echo "$DAT $ERROR $FAIL $VAL" | $LOG
80 sleep $TIMEOUT
81 done
82 # A instrução abaixo nunca é atingida.
83 # Está aqui somente para fins de documentação
84 exit 0
```

Listagem 2: Script de acionamento da VPN de alta disponibilidade

```
01 #!/bin/bash
02 # Script de acionamento HA-VPN
03
04 #VERBOSE=-v
05 VERBOSE=""
06
07 NAME=`basename $0`
08 LOG="/usr/bin/logger -t HA-VPN"
09
10 PARAMETER_FAULT=0
11
12 if [ $# -ne 1 ] ; then
13     PARAMETER_FAULT=1
14 else
15     PARAMETER=$1
16     case $PARAMETER in
17    start)  ;;
18    stop)   ;;
19    *)      PARAMETER_FAULT=1 ;;
20     esac
21 fi
22
23 if [ $PARAMETER_FAULT -ne 0 ] ; then
24     $LOG " ${NAME}: chamado com :$*: ==> erro no parâmetro informado, abortando"
25     echo "`date +%Y%m%d%H%M%S` ${NAME}: erro no parâmetro informado, abortando"
26     exit 1
27 fi
28
29 ACTION_FAIL_START="/etc/init.d/heartbeat stop"
30 ACTION_OK_AGAIN="/etc/init.d/heartbeat start"
31
32 case $PARAMETER in
33     start)  $LOG ${ACTION_FAIL_START} ;
34        ${ACTION_FAIL_START} ;;
35     stop)   $LOG ${ACTION_OK_AGAIN} ;
36        ${ACTION_OK_AGAIN} ;;
37 esac
38 exit 0
```

Duas instalações do Linux-HA

Na **figura 1** vemos uma empresa qualquer, que chamaremos simplesmente de *cliente*, à esquerda. À direita temos um provedor de serviços terceirizado. Tanto o lado do cliente (esquerda, *gw-aa*) quanto o provedor de serviços (direita, *gw-ba*) usam a tecnologia ESP (*Encapsulating Security Payload* ou *área de dados encapsulada de forma segura*, um protocolo do IPsec) para enviar pacotes pelo túnel. O túnel de reserva (Túnel 2) está configurado nos dois roteadores auxiliares no alto do diagrama (*gw-ab* e *gw-bb*) de uma maneira bastante semelhante à do Túnel 1.

O Linux-HA está em operação interligando *gw-aa* com *gw-ab* e também *gw-ba* com *gw-bb*. O *heartbeat* usa a porta serial, que não é usada para nenhum outro propósito e, por isso, é independente da rede, do IPSec e do IPTables. Ambas as

instalações de alta disponibilidade trabalham de forma independente uma da outra. Em condições normais os *gateways* *gw-aa* e *gw-ba* possuem endereços IP locais (10.1.255.254 à esquerda e 10.31.0.254 à direita). Caso haja alguma falha, esses endereços "migram" para os *gateways* *gw-ab* e *gw-bb*. O problema é que, se os IPs "migram" de um servidor a outro, os administradores perdem a capacidade de se conectar remotamente a eles, pois precisam de um IP fixo e imutável para isso. Para resolver o problema, um segundo endereço IP – esse sim, imutável – é atribuído à placa de rede da cada uma das máquinas envolvidas. Cada máquina possui, portanto, dois IPs: um "migratório" (ou seja, compartilhado) e o outro fixo e exclusivo. Quando o administrador quiser se conectar a um servidor em especial, basta usar o IP exclusivo. Em nosso exemplo, o endereço estático de *gw-bb* é 10.31.0.252.

Escrevendo scripts para uma VPN de alta disponibilidade

Os gateways padrão, *gw-aa* e *gw-ba*, rodam um shell script como o mostrado na **listagem 1**. O script é iniciado como um item no arquivo `/etc/inittab` e monitora a acessibilidade do outro lado, independentemente do túnel IPsec. Para isso, o script manda um pequeno pacote UDP para a porta echo (porta número 7). O protocolo *Echo* é um componente padrão em muitas distribuições Linux e pode usar UDP se assim configurado. Com isso, evitamos usar pacotes ICMP Echo Request – o "ping" normal – que são via de regra bloqueados por firewalls zelosos. Observe que, entretanto, a porta 7 também pode ser usada para realizar uma negação de serviço na máquina, portanto recomendamos cautela e monitoramento constante.

Assim que a conexão entre *gw-aa* e *gw-ba* esteja ativa, o contador `FAIL` será sempre zero. Se não houver resposta a um dos nossos pings, o script vai incrementar, na **linha 56**, a variável `FAIL` em uma unidade – mas só se o valor estiver abaixo do configurado em `MAXFAIL`. Se o próximo ping estiver OK, o script reinicializa o valor de `FAIL` novamente a zero (**linha 65**). Se o valor atinge `MAXFAIL`, a **linha 53** chama o programa definido em `ACTION_FAIL_START` (no caso, o script mostrado na **listagem 2**, ao qual é passado o parâmetro `start`). O programa chamado por `ACTION_FAIL_START` para a "pulsação" local, forçando o gateway de reserva a adotar automaticamente o IP do roteador que "caiu".

Esperando pela operação normal

O loop infinito se mantém rodando enquanto espera que o *gateway* principal volte à vida. Quando isso acontece, espera pelo "ping" da porta echo. Quando a primeira resposta chega, depois de uma falha de comunicação bastante longa, a conexão pode não ter, ainda, se estabilizado. Por isso, o script espera um bocadinho antes de recolocar todos os roteadores na condição normal de operação. Para esperar pelo período de histerese (espera), o script atribui um valor negativo a variável `FAIL` – mais precisamente, o valor presente na variável `HYSTERE` (**linha 62**) – e incrementa o valor de `FAIL` para cada ping que responda corretamente (**linha 70**). Se outro erro ocorrer na fase de recuperação, a **linha 46** coloca em `FAIL` um valor maior que o configurado em `MAXFAIL` – com isso, o sistema continua usando os roteadores de backup até que a conexão principal esteja normalizada e estável.

As coisas não voltam ao normal até que `FAIL` possua o valor zero. Só então o script chama `ACTION_OK_AGAIN` na **linha 75**. Esse programa também reinicia a pulsação (*heartbeat*) e reatribui os endereços IP corretos a todos os roteadores para funcionarem na condição inicial.

Essa abordagem multinível impede o "roteamento flipflop", no qual os *gateways* ficam alternando entre os túneis cíclica e rapidamente. As conexões de Internet são propensas a falhas de poucos segundos. Se você procurar em seus próprios *logs* vai encontrar vários exemplos. Três minutos é um tempo de espera que consideramos ser bastante conveniente.

Nossa experiência em instalações desse tipo mostra que as falhas costumam sempre seguir um certo padrão. Se a pulsação em um lado do túnel cai, ela também será interrompida na outra ponta depois de um segundo – obviamente, os relógios dos dois sistemas têm que estar sincronizados para que consigamos perceber esses eventos olhando os registros (*logs*) do sistema.

Satisfação garantida

A solução descrita no artigo está funcionando desde o início de 2004 e demonstrou repetidamente que os usuários sequer notaram as falhas quando estas ocorreram. Em caso de um cataclisma, os mecanismos de reenvio de dados das pilhas TCP/IP dos servidores e clientes podem facilmente "aguentar" por volta de dez segundos – tempo suficiente para que o sistema de reserva consiga estabilizar-se e entrar em ação.

A conexão de reserva não usa a função de monitoração, o que poderia ser um problema. Na prática, se um provedor diferente for usado para a conexão de reserva, é muito difícil que ambos os provedores saiam do ar ao mesmo tempo. E, mesmo que isso aconteça, nem o protocolo BGP poderá ajudá-lo. ■

Informações

[1] Freeswan: www.freeswan.ca/code/super-freeswan/

[2] RFC 1745: www.ietf.org/rfc/rfc1745.txt

[3] SSPE: sspe.sourceforge.net

[4] Linux-HA: www.linux-ha.org

[5] RFC 2401: www.ietf.org/rfc/rfc2401.txt