REDES II: Configuração de Traps SNMP

Vitor Bruno de Oliveira Barth
vbob@vbob.com.br

Instituto Federal de Educação, Ciência e Tecnologia de Mato Grosso — March 10, 2019

1 Introdução

Traps SNMP são amplamente utilizadas para o Gerenciamento de Redes. Elas permitem que o Agente comunique ao Gerente atividades indevidas, falhas ou quaisquer outras ocorrências que devam ser analisdas.

Neste trabalho, é apresentada a forma de configuração de duas traps:

1. Mudança de estado de uma interface de rede para *down*.
2. Acesso indevido via SSH

Ao ser recebida pelo gerente, cada uma dessas traps executará um *script*, que servirá para avisar ao usuário do comportamento detectado.

2 Configuração das Traps

Para a configuração das Traps, são utilizadas duas MIBs padrão do pacote *net-snmp*:

1. **IF-MIB::linkDown** para indicar que falha em uma conexão de rede
2. **SNMPv2-MIB::authenticationFailure** para indicar falha na conexão via SSH

Estas MIBs foram projetdas justamente para este propósito, e por esta razão é desnecessária a criação de MIBs adicionais.

2.1 Configuração dos Gatilhos

Ao ser recebida, cada trap deverá executar um *script shell* que realizará ações para avisar ao usuário sobre a chamada da *trap*.

A definição de tais gatilhos é realizada no arquivo */etc/snmp/snmptrapd.conf*. O arquivo editado é apresentado abaixo:

File 1 /etc/snmp/snmptrapd.conf

```
disableAuthorization yes
authCommunity log,execute,net public
outputOption n

traphandle IF-MIB::linkDown /usr/local/etc/snmptrap/linkDown.sh
traphandle SNMPv2-MIB::authenticationFailure /usr/local/etc/snmptrap/authenticationFailure.sh
```

O comando *disableAuthorization* indica que não é necessária autorização para que uma *trap* seja processada.

O comando *authCommunity log,execute,net public* permite à comunidade *public* seja utilizada para o envio de *traps*.

O comando *outputOption n* indica que as OIDs serão armazenadas de forma numérica.

Após ser editado, o *daemon* deve ser reiniciado, através do comando

Command Line

```
# systemctl restart snmptrapd
```

Deste modo, o servidor estará ouvindo às *traps* e configurado para executar os *scripts* desejados.

3 *Scripts*

Com os *scripts* definidos no arquivo */etc/snmp/snmptrapd.conf*, estes devem ser construídos para que sejam executadas ações para avisar o usuário das ocorrências.

No caso da *trap* **IF-MIB::linkDown** será enviado um e-mail ao administrador da rede.

Ao ser recebida a *trap* **SMPv2-MIB::authenticationFailure**, esta ocorrência será salva em um arquivo, onde estarão as últimas 10 tentativas mal sucedidas de *login*.

3.1 IF-MIB::linkDown

Para se enviar um e-mail utilizando *Linux* pode ser utilizado o pacote *sendmail*. O *script* completo está apresentado abaixo.

File 2 /usr/local/etc/snmptrap/linkDown.sh

```
#!/bin/bash

email=sysadmin@enterprise.com
file=/usr/local/etc/snmptrap/email.txt

sendmail $email < $file
```

Onde *email.txt* é o corpo da mensagem que se deseja ter enviada.

3.2 SNMPv2-MIB::authenticationFailure

Para se armazenar em arquivo as últimas 10 tentativas de *login* realizadas por SSH, deve-se obter algumas informações contidas na mensagem enviada pela *trap*. Para isso, será feita a leitura de comandos e analisados usando o comando *grep*

File 3 /usr/local/etc/snmptrap/authenticationFailure.sh

```
file=/home/user/loginAttempts.txt
while read param
do
  echo -e "$param "
    if $param | grep -q "STRING"; then
      param = ${param#*'"'}; param=${param%'"'*}
      echo $param >> $file
      echo $(tail -10 $file) > $file
    fi
exit 0
```

4 Geração das Traps no Agente

Agora que as *traps* estão configuradas no Gerente, podemos configurar o agente para que as envie automáticamente.

4.1 IF-MIB::linkDown

Em computadores *Linux* que utilizam o *daemon NetworkManager* para gerenciar as interface de rede, já existe uma infraestrutura pronta para executar comandos no caso de mudanças no estado da conexão: o *dispatcher.d*.

Todos os arquivos colocados na pasta */etc/NetworkManager/dispatcher.d* são executados sempre que há uma mudança em alguma interface de rede.

Para isso, foi criado o arquivo */etc/NetworkManager/dispatcher.d/99-snmptrap.sh*, que será a fonte da *trap*.

File 4 /etc/NetworkManager/dispatcher.d/99-snmptrap.sh

```
interface=$1
event=$2
address=127.0.0.1
MIB=IF-MIB::linkDown
OID=linkDown

if [[ ( "$interface" == "enp2s0" && "$event" != "up" ) ]]
then
   snmptrap -v 2c -c public $address "" $MIB $OID s $interface
fi
exit 0
```

4.2 SNMPv2-MIB::authenticationFailure

O serviço de autenticação do *Linux* é chamado de PAM. Este serviço possui uma pasta de configurações, */etc/pam.d*, onde estão definidos os diferentes formatos de autenticação. Dentre eles, está o /etc/pam.d/sshd, que permite as autenticações via SSH.

Para que seja executado um *script* ao se executar autenticação via SSH, esse arquivo deve ser alterado, adicionando-se as seguintes linhas:

File 5 /etc/pam.d/sshd

```
auth    [success=2 default=ignore]    pam_unix.so nullok_secure
auth    optional                      pam_exec.so /usr/local/etc/snmptrap/sshFailure.sh
```

File 6 /etc/NetworkManager/dispatcher.d/99-snmptrap.sh

```
address=127.0.0.1
MIB=SNMPv2-MIB::authenticationFailure
OID=authenticationFailure
data=(date '+%Y-%m-%d %H:%M:%S')/$PAM_USER/$PAM_RHOST/InvalidPass

snmptrap -v 2c -c public $address "" $MIB $OID s $data
```