3 Traps SNMP

Visão geral

O fluxo de comunicação de uma 'trap SNMP' é o inverso do de uma coleta.

Para este tipo de item o dispositivo monitorado inicia a comunicação enviando um 'trap' que é coletado pelo processo 'Zabbix Trapper' que está presente o Zabbix Server e no Zabbix Proxy.

Normalmente as 'traps' são enviadas quando determinada condição ocorre, a partir desta mudança o Agente SNMP se conecta ao servidor SNMP (no caso o Zabbix Trapper) usando a porta 162 (para as consultas SNMP é utilizada a porta 161). Usando 'traps' você poderá detectar problemas logo que ocorrerem, sem ter que aguardar uma eventual fila de coletas.

O recebimento de 'traps SNMP' no zabbix foi desenvolvido para funcionar em conjunto com o **snmptrapd** chamando um script perl ou SNMPTT.

O 'workflow' de um recebimento de trap:

1. **snmptrapd** recebe a trap
2. **snmptrapd** envia o dado para o SNMPTT ou chama um 'trap receiver' escrito em Perl
3. O SNMPTT ou o 'Perl trap receiver' interpreta, formata e grava o dado em um arquivo
4. O 'Zabbix SNMP trapper' lê e interpreta os dados do arquivo
5. Para cada trap o Zabbix procura todos os itens de “SNMP trapper” com interfaces compatíveis com a origem do dado. Observe que somente o “IP” ou o “DNS” da interface do host será utilizado durante a pesquisa.
6. Para cada item localizado, a trap será comparada com a expressão regular em “snmptrap[regexp]”. O dado da trap será enviado como um novo valor para **todos** os itens compatíveis. Se não for encontrado um item compatível, e existir um item de “snmptrap.fallback” definido, o valor será repassado para ele.
7. Se a trap não conseguir salvar o valor recebido em nenhum item, o Zabbix irá registrar isso no “log de traps não correspondentes

”. (Que pode ser habilitado através da opção “Registrar traps SNMP não correspondentes” disponível em `Administração` → `Geral` → `Outros`.)

3.1 Configurando as traps SNMP

A configuração dos campos a seguir é específica para itens deste tipo:

- Você precisará possuir uma interface SNMP

Em *Configuração* → *Hosts*, no campo **Interface do Host** defina uma interface com o IP ou DNS correto. O endereço de cada trap recebida é comparada com o IP ou com o DNS de todas as interfaces SNMP para localizar os hosts correspondentes.

- Configure o item

Utilize uma das chaves de trap SNMP no campo **Chave** do item:

Chave		
Descrição	**Valor retornado**	**Comentários**
snmptrap[regexp]		
Captura qualquer trap SNMP que corresponda com um endereço definido pela expressão regular informada no campo **regexp**	Trap SNMP	Este item só pode ser definido em interfaces SNMP. Este item é suportado desde o Zabbix 2.0.0. *Observação*: A partir do Zabbix 2.0.5, macros de usuário e expressões globais são suportadas para este tipo de item.
snmptrap.fallback		
Captura todas as traps SNMP de um determinado IP que não tenha sido recebida por nenhum item **snmptrap[]** daquela interface	Trap SNMP	Este item só pode ser definido em interfaces SNMP. Este item é suportado desde o Zabbix **2.0.0.**

Expressões regulares de várias linhas não são suportadas neste momento.

Defina o **Tipo da informação** para 'Log' para que os registros de hora sejam processados. Observe que qualquer outro formato, tal qual o numérico, também são aceitáveis mas requerem um gerenciador de trap personalizado.

Para a monitoração de trap SNMP, você primeiro deverá configura-la.

3.2 Configurando a monitoração via SNMP trap

Configurando o Zabbix Server/Proxy

Para receber as traps, o Zabbix Server/Proxy deverá estar configurado para iniciar o processo de SNMP trapper e apontado para o arquivo de traps que estará sendo alimentado pelo SNMPTT ou pelo 'perl trap receiver'. Para fazer isso, edite o arquivo de configuração (zabbix_server.conf ou zabbix_proxy.conf):

1. StartSNMPTrapper=1
2. SNMPTrapperFile=[TRAP FILE]

Se o parâmetro do systemd **PrivateTmp** tiver sido definido é improvável que funcione no */tmp*.

Configurando o SNMPTT

Primeiramente o 'snmptrapd' precisa estar configurado para utilizar o SNMPTT.

Para uma melhor performance, o SNMPTT precisa estar configurado como um daemon utilizando o **snmptthandler-embedded** para enviar as traps. Veja mais intruções sobre configuração do SNMPTT neste endereço:
http://snmptt.sourceforge.net/docs/snmptt.shtml

Quando o SNMPTT estiver apto a receber as 'traps', ajuste-o para registrar as traps:

1. registre as traps no arquivo de traps que será lido pelo Zabbix:
   log_enable = 1
   log_file = [TRAP FILE]
2. defina o formato de data e hora:
   date_time_format = %H:%M:%S %Y/%m/%d = [DATE TIME FORMAT]

Agora formate as traps de forma a possibilitar que o Zabbix as reconheça (edite o arquivo snmptt.conf):

1. Cada instrução de formato deverá começar com “ZBXTRAP [address]”, onde [address] será o valor usado na comparação com as interfaces (pelo IP ou DNS). Exemplo.:
   EVENT coldStart .1.3.6.1.6.3.1.1.5.1 “Status Events” Normal
   FORMAT ZBXTRAP $aA Device reinitialized (coldStart)
2. Veja mais sobre formato de SNMP trap a seguir

Não utilize traps 'não esperadas' - o Zabbix não estará apto a reconhece-las. Traps não esperadas podem ser tratadas ao definir um evento geral no `snmptt.conf`:
EVENT general .* “General event” Normal

Configurando o ''Perl trap receiver''

Requerimentos: Perl, pacote Net-SNMP compilado com --enable-embedded-perl (já é padrão no Net-SNMP 5.4)

'Perl trap receiver' (olhe em misc/snmptrap/zabbix_trap_receiver.pl) poderá ser utilizado para encaminhar as traps para o Zabbix Server/Proxy diretamente a partir do snmptrapd. Para configurar:

- adicione o script perl no arquivo de configuração do 'snmptrapd' (snmptrapd.conf), Exemplo:
  perl do "[FULL PATH TO PERL RECEIVER SCRIPT]";
- configure o recebedor, exemplo:
  $SNMPTrapperFile = '[TRAP FILE]';
  $DateTimeFormat = '[DATE TIME FORMAT]';

Se o nome do script não estiver entre aspas, o snmptrapd irá se recusar a iniciar com mensagens similares a estas:

```
Regexp modifiers "/l" and "/a" are mutually exclusive at (eval 2) line 1, at
end of line
Regexp modifier "/l" may not appear twice at (eval 2) line 1, at end of line
```

Formato do SNMP trap

Tanto os 'perl trap receivers' quanto o 'SNMPTT trap configuration' precisam formatar a trap conforme o padrão a seguir: **[timestamp] [the trap, part 1] ZBXTRAP [address] [the trap, part 2]**, onde

- [timestamp] - momento de ocorrência do evento
- ZBXTRAP - cabeçalho que indica o início de uma nova linha de trap
- [address] - endereço IP para localizar o host a receber a trap

Observe que o “ZBXTRAP” e o “[address]” serão removidos da mensagem durante o processamento. Se a trap utilizar outro formato, o Zabbix pode analisa-las de forma inesperada.

Exemplo de trap:
11:30:15 2011/07/27 .1.3.6.1.6.3.1.1.5.3 Normal “Status Events” localhost - ZBXTRAP 192.168.1.1 Link down on interface 2. Admin state: 1. Operational state: 2
Esta trap será enviada para um host com a interface SNMP com IP=192.168.1.1:
11:30:15 2011/07/27 .1.3.6.1.6.3.1.1.5.3 Normal “Status Events” localhost - Link down on interface 2. Admin state: 1.

3.3 Requerimentos de sistema

Rotação de logs

O Zabbix não provê rotação de logs que possa ser gerida pelo usuário. A rotação de logs primeiro irá renomear o arquivo antigo e após isso apaga-lo para garantir que nenhuma trap será perdida, conforme processo a seguir:

1. O Zabbix abre o arquivo de traps na última posição conhecida e vai para o passo 3
2. O Zabbix verifica se o arquivo aberto já foi rotacionado ao comparar o número do inode definido para o mesmo. Se não existe arquivo aberto, o Zabbix reseta a informação de última posição conhecida e retorna ao passo 1.
3. O Zabbix lê o dado do arquivo aberto e define uma nova localização.
4. Os novos dados são analisados. Se era o arquivo rotacionado, o arquivo é fechado e retorna ao passo 2.
5. Se não existe novo dado, o Zabbix aguarda 1 segundo e retorna ao passo 2.

O tamanho máximo de um arquivo de log para o Zabbix é de 2GB. O arquivo de log precisa ser rotacionado antes deste limite.

Sistema de arquivos

Devido à forma de implementação, o Zabbix precisa que o sistema de arquivos suporte inodes para diferenciar os arquivos (a informação é obtida através da função `stat()`).

3.4 Exemplo de configuração

Este exemplo utiliza o snmptrapd + SNMPTT para enviar traps ao Zabbix Server:

1. **zabbix_server.conf** - configure o Zabbix para iniciar o SNMP trapper e defina a localização do arquivo file:
   StartSNMPTrapper=1
   SNMPTrapperFile=/tmp/my_zabbix_traps.tmp
2. **snmptrapd.conf** - adicione o SNMPTT como o gerenciador de traps:
   traphandle default snmptt
3. **snmptt.ini** - configure o arquivo de saída e o formato de hora:
   log_file = /tmp/my_zabbix_traps.tmp
   date_time_format = %H:%M:%S %Y/%m/%d
4. **snmptt.conf** - defina o formato padrão de trap:

EVENT general .* "General event" Normal
FORMAT ZBXTRAP $aA $ar

5. Crie um item SNMP TEST:
   IP da Interface SNMP do Host IP: 127.0.0.1
   Chave: snmptrap["General"]
   Formato de hora do log: hh:mm:ss yyyy/MM/dd

Teste o funcionamento:

1. Comando utilizado para enviar uma trap:
   snmptrap -v 1 -c public 127.0.0.1 '.1.3.6.1.6.3.1.1.5.3' '0.0.0.0' 6 33 '55' .1.3.6.1.6.3.1.1.5.3 s “teststring000”
2. Que será recebida:
   15:48:18 2011/07/26 .1.3.6.1.6.3.1.1.5.3.0.33 Normal "General event" localhost - ZBXTRAP 127.0.0.1 127.0.0.1
3. Valor par ao teste do item:
   15:48:18 2011/07/26 .1.3.6.1.6.3.1.1.5.3.0.33 Normal "General event" localhost - 127.0.0.1

Este exemplo simples utiliza o SNMPTT como **traphandle**. Para melhor performance em ambientes de produção, utilize um script Perl para encaminhar as traps do snmptrapd ao SNMPTT ou diretamente ao Zabbix.