ZABBIX

Zabbix monitorando autenticação Radius do NPS

Objetivo é Monitorar o serviço de autenticação do NPS(Network Policy Server) do Windows. É necessário criar um usuário, uma chave secret e políticas que permitam a autenticação no Radius, após essa etapa basta implementar um script em Shell no próprio servidor Linux do Zabbix server e capturar o retorno dele através de um item de checagem externa no Zabbix. Nessa implementação de monitoramento já estamos levando em conta que o serviço de autenticação já esta funcionando na estrutura de network em questão.

Implementação: Leonardo De Oliveira Freitas

Ambiente implementado.

S.O: CentOS Linux 7 (Core)

Zabbix: 4.2.6

Para testar a resposta da autenticação radius vamos utilizar a ferramenta radtest do pacote freeradius-utils.

Instalando pacote necessario.

yum install freeradius-utils

Para utilizar o item external check do zabbix é necessário que o script fiquei localizado dentro do diretório de leitura padrão do zabbix. /usr/lib/Zabbix/externalscripts. Lembrando que esse diretório pode variar do CentOS para debian e ubuntu server, ou outras distribuições.



Script para realizar o teste.

PassUser -> Senha do usuário Active Directory.

PassSecret -> Senha da secret gerada no NPS.

DNS do servidor pode ser substituído por \$1 -> Assim conseguimos aplicar o template em mais de um host.

-t mschap -> Criptografia utilizada no envio do pacote. Precisa ser suportada na política de aceite do NPS.



Capturando retorno do teste após execução do script. Quando o serviço de autenticação está funcionando sem problemas o retorno da consulta é Access-Accept.

Podemos analisar que quando o serviço de autenticação não está funcionando o retorno da consulta é diferente, retornando Access-Reject.

```
[root@ZABBIXHUJF externalscripts]# ./Radius.sh
(0) -: Expected Access-Accept got Access-Reject
[root@ZABBIXHUJF externalscripts]# ./Radius.sh
(0) -: Expected Access-Accept got Access-Reject
[root@ZABBIXHUJF externalscripts]# ./Radius.sh
Received Access-Accept Id 141 from 10.2.150.60:1812 to 0.0.0.0:0 length 135
```

Você consegue visualizar o tráfego de rede do pacote Radius utilizando o WireShark.

```
Time Source
                                       Destination
                                                                   Protocol
                                                                             Length
   471 9... 192.168.20.15
                                                                                 212 Access-Request(1) (id=100, l=168)
                                       192.168.20.31
                                                                   RADIUS
  472 9... 192.168.20.15
                                                                   RADIUS
                                                                                239 Access-Request(1) (id=113, l=195)
                                      192.168.20.31
                                                                   RADIUS
                                                                                 64 Accounting-Response(5) (id=112, l=20)
   527 9... 192.168.20.31
                                       192.168.20.15
  682 9... 192.168.20.31
                                       192,168,20,15
                                                                   RADIUS
                                                                                 177 Access-Accept(2) (id=100, l=133)
   743 9...
           192.168.20.31
                                       192.168.20.15
                                                                   RADTUS
                                                                                 159 Access-Accept(2) (id=113.
  Code: Access-Request (1)
  Packet identifier: 0x71 (113)
  Length: 195
  Authenticator: ae8370b7c00fe7d3ce001585526bdbdd
  [The response to this request is in frame 743]

▼ Attribute Value Pairs

  v AVP: l=6 t=NAS-IP-Address(4): 192.168.20.15
       NAS-IP-Address: 192,168,20,15

▼ AVP: l=6 t=NAS-Port(5): 0
      NAS-Port: 0

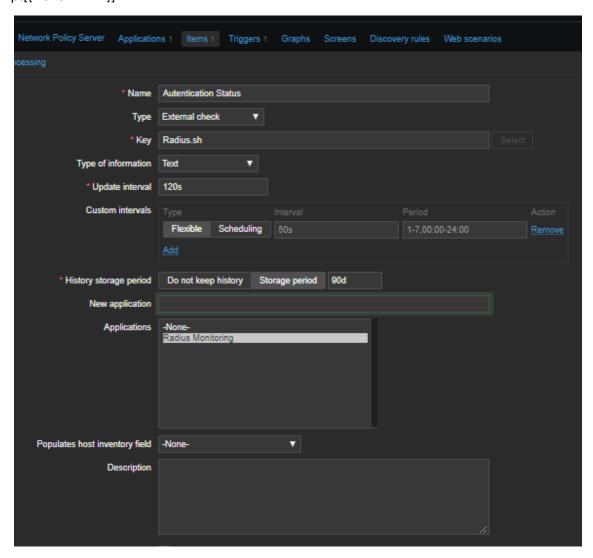
▼ AVP: l=6 t=NAS-Port-Type(61): Wireless-802.11(19)
      NAS-Port-Type: Wireless-802.11 (19)
  w AVP: l=14 t=User-Name(1): 40331acafc99
       User-Name: 40331acafc00
    AVP: l=18 t=User-Password(2): Encrypted
       User-Password (encrypted): 405a8f7d8528788df4c309f1619297ee
      P: t=6 t=Service-Type(6): Call
       Service-Type: Call-Check (10)
```



Zabbix monitorando autenticação Radius do NPS

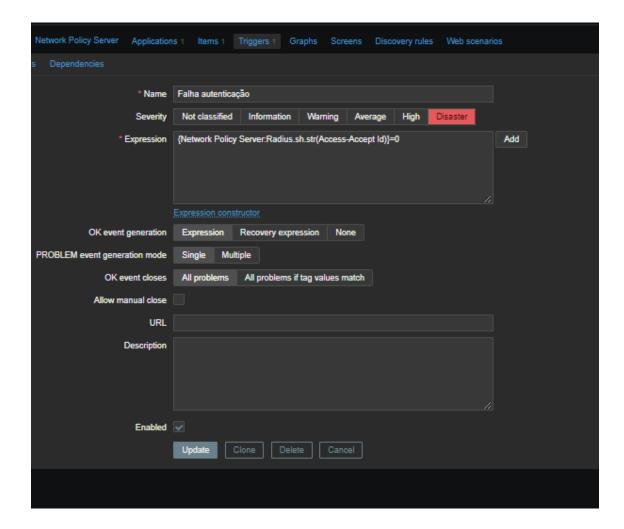
Item external check - No Zabbix basta criar um item agora para ler o arquivo e tratar o retorno, diagnosticando se a autenticação está respondendo corretamente ou não.

Key -> Script[{HOTS.NAME}]



Zabbix monitorando autenticação Radius do NPS

Trigger – Percorremos a string em busca de retorno "Access-Accept Id" pois somente o retorno com sucesso retorna o ID. Você pode livremente tratar essa condição de disparo utilizando outras práticas também, tais como a busca por "Access-Reject" ou expressões regulares.



Nos exemplos da implementação acima ocultei detalhes que desrespeitam a segurança do ambiente pois o objetivo é mostrar o funcionamento da estrutura de implementação e apontar a metodologia utilizada para a necessidade em questão. Porém toda a estrutura foi implementada com sucesso.

ZABBIX

Zabbix monitorando autenticação Radius do NPS

```
<irem>
<name>Autentication Status</name>
<type>10</type>
<snmp_community/>
<snmp_oid/>
<key>Radius.sh[{HOST.IP}]</key>
<delay>2m</delay>
<history>90d</history>

</item>

<triggers>
    <trigger>
    <expression>{Network Policy Server:Radius.sh[{HOST.IP}].str(Access-Accept Id)}=0</expression>
    </trigger>
</trigger>>
</trigger>>
```

Fontes de pesquisa.

https://www.zabbix.com/documentation/3.0/pt/manual/config/items/itemtypes/external

https://www.zabbix.com/documentation/3.4/manual/config/items/item

https://blog.zabbix.com/deep-dive-in-zabbix-preprocessing/8288/

https://www.ietf.org/rfc/rfc2865.txt