

**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 está 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 fique 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.

```
root@ZABBIXHUJF:/usr/lib/zabbix/externalscripts
[root@ZABBIXHUJF externalscripts]# pwd
/usr/lib/zabbix/externalscripts
[root@ZABBIXHUJF externalscripts]#
```

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.

```
root@ZABBIXHUJF:/usr/lib/zabbix/externalscripts
radtest -x -t mschap unimedjf\zabbix PassUser npshujf 1812 PassSecret | grep "Received Access-Accept"
```

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

```
root@ZABBIXHUJF:/usr/lib/zabbix/externalscripts
[root@ZABBIXHUJF externalscripts]# ./Radius.sh
Received Access-Accept Id 49 from 10.2.150.60:1812 to 0.0.0.0:0 length 135
[root@ZABBIXHUJF externalscripts]#
```

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.

No.	Time	Source	Destination	Protocol	Length	Info
471	9...	192.168.20.15	192.168.20.31	RADIUS	212	Access-Request(1) (id=100, l=168)
472	9...	192.168.20.15	192.168.20.31	RADIUS	239	Access-Request(1) (id=113, l=195)
527	9...	192.168.20.31	192.168.20.15	RADIUS	64	Accounting-Response(5) (id=112, l=20)
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	RADIUS	159	Access-Accept(2) (id=113, l=115)

Code: Access-Request (1)  
 Packet identifier: 0x71 (113)  
 Length: 195  
 Authenticator: ae8370b7c00fe7d3ce001585526bdbd  
[\[The response to this request is in frame 743\]](#)

▼ Attribute Value Pairs

- ▼ 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)
- ▼ AVP: l=14 t=User-Name(1): 40331acafc99  
 User-Name: 40331acafc99
- ▼ AVP: l=18 t=User-Password(2): Encrypted  
 User-Password (encrypted): 405a8f7d8528788df4c309f1619297ee
- ▼ AVP: l=6 t=Service-Type(6): Call-Check(10)  
 Service-Type: Call-Check (10)

**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}]

The screenshot shows the Zabbix web interface for configuring a new item. The top navigation bar includes links for Network Policy Server, Applications 1, Items 1, Triggers 1, Graphs, Screens, Discovery rules, and Web scenarios. The main content area is titled 'Processing' and contains the following configuration fields:

- Name:** Authentication Status
- Type:** External check
- Key:** Radius.sh (with a 'Select' button)
- Type of information:** Text
- Update interval:** 120s
- Custom intervals:** A table with columns for Type, Interval, Period, and Action.

Type	Interval	Period	Action
Flexible	Scheduling	50s	1-7,00:00-24:00

An 'Add' link is located below the table.
- History storage period:** Do not keep history (selected), Storage period, 90d
- New application:** (empty text field)
- Applications:** A dropdown menu with '-None-' and 'Radius Monitoring' (selected).
- Populates host inventory field:** -None-
- Description:** (empty text area)

**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.

The screenshot shows the Zabbix web interface for configuring a trigger. The top navigation bar includes links for 'Network Policy Server', 'Applications 1', 'Items 1', 'Triggers 1' (active), 'Graphs', 'Screens', 'Discovery rules', and 'Web scenarios'. Below the navigation bar, the 'Dependencies' tab is selected. The main configuration area for the trigger is as follows:

- Name:** Falha autenticação
- Severity:** Not classified, Information, Warning, Average, High, Disaster (Disaster is selected)
- Expression:** {Network Policy Server:Radius.sh\_str(Access-Accept Id)}=0. An 'Add' button is next to the expression field.
- Expression constructor:** A link to the expression constructor tool.
- OK event generation:** Expression, Recovery expression, None (Expression is selected)
- PROBLEM event generation mode:** Single, Multiple (Single is selected)
- OK event closes:** All problems, All problems if tag values match (All problems is selected)
- Allow manual close:** ☐
- URL:** (Empty text field)
- Description:** (Empty text area)
- Enabled:** ☒
- Buttons:** Update, Clone, Delete, Cancel

Nos exemplos da implementação acima oculteí 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.

```
<item>
  <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>
</triggers>
```

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>