

ABERTURA DE CHAMADO NO OTRS COM ZABBIX

1. Nome

Abertura de chamados no OTRS com o Zabbix

2. Introdução

Realizaremos a integração do Zabbix com o OTRS via web service.

3. Pré-requisitos

Para realizar o procedimento de abertura de chamado é necessário:

- 1) Instalação do python no servidor zabbix
- 2) Instalação do python-pip no servidor zabbix
- 3) Instalação do modulo otrs-python no servidor zabbix
- 4) Instalação do CPAN no servidor zabbix
- 5) instalação dos módulos do OTRS no servidor zabbix
- 4) Script de abertura create.py
- 5) A importação do webservice deverá ser realizada
- 6) Realizar previamente as informações abaixo:
 - a) IP do servidor OTRS
 - b) Usuário e senha para conectar no otrs
 - c) Cliente no qual o chamado será aberto
 - d) Título do chamado
 - e) Descrição do chamado

- f) Fila cadastrada no OTRS
- g) Serviço cadastrado no OTRS
- h) Nível de SLA cadastrado no OTRS

4. Procedimento de instalação

1) Instalação do python e python pip

- a) `yum install python python-pip -y`

2) Instalação do python-otrs com pip

- a) `pip install python-otrs -y`

3) Instalação do CPAN

- a) `yum install cpan -y`

4) Instalando os módulos do otrs

- a) `scp /opt/otrs/bin/otrs.CheckModules.pl root@zabbix:/tmp/`
- b) `perl -MCPAN -e otrs.CheckModules.pl`

5) Instalando módulos perl

- a) `yum install "perl(ExtUtils::MakeMaker)" "perl(Sys::Syslog)" -y`
- b) `yum install "perl(Archive::Tar)" "perl(Archive::Zip)" "perl(Crypt::Eksblowfish::Bcrypt)" "perl(Crypt::SSLeay)" "perl(Date::Format)" "perl(DBD::Pg)" "perl(Encode::HanExtra)" "perl(IO::Socket::SSL)" "perl(JSON::XS)" "perl(Mail::IMAPClient)" "perl(IO::Socket::SSL)" "perl(ModPerl::Util)" "perl(Net::DNS)" "perl(Net::LDAP)" "perl(Template)" "perl(Template::Stash::XS)" "perl(Text::CSV_XS)" "perl(Time::Piece)" "perl(XML::LibXML)" "perl(XML::LibXSLT)" "perl(XML::Parser)" "perl(YAML::XS)" -y`
- c) `yum install "perl(Crypt::Eksblowfish::Bcrypt)" "perl(Encode::HanExtra)" "perl(JSON::XS)" "perl(JSON::XS)" "perl(Mail::IMAPClient)" "perl(ModPerl::Util)" "perl(YAML::XS)" -y`

6) Copiar o arquivo "create.py" para o servidor Zabbix

- a) `scp /home/otrs/create.py root@zabbix:/usr/lib/zabbix/alertscripts/`

7) Executar o script com parametro --help para verificar os itens que deverão ser adicionados

```
[root@Zabbix alertscripts]# python create.py --help
```

```
usage: create.py [-h] [--otrs SERVER] [--webservice WEBSERVICE] [--user USER]
                [--pass PASSWORD] [--customer CUSTOMER] [--title TITLE]
                [--desc DESCRICAO] [--fila FILA] [--servico SERVICO]
                [--sla SLA] [--triggerid TRIGGERID] [--host HOST]
                [--status STATUS]
```

Criar chamado.

optional arguments:

```
-h, --help          show this help message and exit
--otrs SERVER       IP OTRS
--webservice WEBSERVICE
                    OTRS Web Service
--user USER        OTRS user
--pass PASSWORD     OTRS pass
--customer CUSTOMER Cliente
--title TITLE       Titulo do chamado
--desc DESCRICAO    Descricao do chamado
--fila FILA         Fila de atendimento
--servico SERVICO   Servico de abertura
--sla SLA           SLA a ser utilizado
--triggerid TRIGGERID
                    Trigger ID do zabbix
--host HOST         Nome do host no zabbix
--status STATUS     Indisponivel
```

5. Execução do script

O script será executado a partir do zabbix server.

6. Teste de conexão e script

Agora que temos o script pronto e as informações coletadas corretamente, faremos um teste, execute o comando abaixo no servidor zabbix.

```
python criar_chamado_zabbix.py --otrs {IP_OTRS} --webservice ZabbixOTRS --user {USUARIO} --pass {SENHA} --customer {CLIENTE} --title "Teste de chamado" --desc "Descrição do Chamado" --fila "{FILAS}" --servico "{SERVICO}" --sla "{SLA}"
```

7. Criando a ação no zabbix

1) Acessar o zabbix em Configuration>Actions e apertar em "New Action"

2) Definir um nome para a ação

3) Em "Type of calculation" selecionar "And/Or"

4) Em "Conditions" selecione:

a) Maintenance status not in maintenance

b)* Trigger = Centos7-Teste: HTTP service is down on Centos7-Teste

* Dependo da trigger que você escolher seja necessário alterar a descrição do chamado, para uma melhor compreensão.

5) Marque a opção "Enabled"

6) Na aba "Operations"

7) Em "Operations" clique em "New"

8) Em "Operation type" selecione "Remote command"

9) Em "Target list" selecione "Current host"

10) Em "Type" selecione "Custom script"

11) Em "Execute on" marque a opção "Zabbix server"

12) Em "Commands" copie e cole o conteúdo a seguir:

```
/bin/python /usr/lib/zabbix/alertscripts/create.py --otrs {IP_OTRS} --webservice
ZabbixOTRS --user {USUARIO} --pass {SENHA} --customer {CLIENTE} --title
"ALERTA: {TRIGGER.SEVERITY} - {TRIGGER.NAME} - {TRIGGER.STATUS}" --desc
"O host abaixo esta com problemas.
```

Nome do host: {HOST.HOST}

Nome do item: {ITEM.NAME}

Nome da trigger: {TRIGGER.NAME}

Status da trigger: {TRIGGER.STATUS}

Severidade da trigger: {TRIGGER.SEVERITY}

Data do evento: {EVENT.DATE} {EVENT.TIME}

Data atual: {DATE} {TIME}

```
Total de tempo com problema: {EVENT.AGE}" --fila "{FILA}" --servico "{SERVICO}" --
sla "{SLA}" --triggerid {TRIGGER.ID} --host {HOST.NAME} --status Indisponivel
```

13) Você deve ficar atento para mudar os campos:

otrs: que contém o IP do servidor

webservice: que contém o nome do webservice que você criou

user: nome do usuário para a conexão no otrs

senha: senha do usuário do otrs

customer: em nome de quem o chamado será aberto (solicitante)

fila: nome da fila para quem vai o chamado criado (Se você tiver mais de uma fila, sugiro criar outros scripts com filas diferentes, nesse caso, você deverá alterar o item 4 do passo 7)

serviço: serviço cadastrado no otrs

sla: nível de sla cadastrado no otrs

14) Clique em "Add"

15) Cliquem em "Add" novamente.

8. Testando script

1) Adicione um host de teste nas condições criadas da ação, consultar o item 4 do passo 7, e cause o incidente.

2) Verifique no OTRS se o chamado foi aberto com as especificações corretas