



Sumário

Introdução	8
Conceitos Zabbix	9
Zabbix Server	9
Zabbix Proxy	10
Zabbix Agent	10
Zabbix Sender	11
Zabbix Get	11
Servidores	12
Topologia	14
Instalação	15
LI2454 - Zabbix-Server Master	15
LI2614 - Zabbix-Server Slave	20
LI2455 - Zabbix-WEB	24
LI2458 - Zabbix-Proxy THPP	26
LI2459 - Zabbix-Proxy Windows	29
LI2460 - Zabbix-Proxy NIX	32
LI2461 - Zabbix-Proxy SNMP	35
LI2497 - Zabbix-Proxy VMWARE	38
LI2456 - Zabbix-MySQL Master	44
LI2457 - Zabbix-MySQL Slave	61
LI2616 - Zabbix-MySQL DR Dino Bueno	68
LI2615 - Zabbix-Server+WEB DR (Dino Bueno)	73
LI2617 - Proxy THPP DR (Dino Bueno)	78
LI2618 - Proxy Windows DR (Dino Bueno)	80
LI2619 - Proxy Linux Unix DR (Dino Bueno)	83

LI2620 - Proxy SNMP DR (Dino Bueno)	86
LI2621 - Proxy VMWARE DR (Dino Bueno)	89
Housekeeping	93
Integração com LDAP	96
Integração com SDM	98
Categorias do SDM	98
Media Type	98
CA Open Request - Prod Minor	99
CA Open Request - Prod Major	100
CA Open Request - Prod Critical	101
CA Open Request - THPP Minor	102
CA Open Request - THPP Major	103
CA Open Request - THPP Critical	105
Action	106
Ticket CA SDM THPP - Warning - Minor	106
Ticket CA SDM THPP - Average - Major	106
Ticket CA SDM THPP - HighDisaster - Critical	107
Ticket CA SDM Producao - Warning - Minor	108
Ticket CA SDM Producao - Warning - Major	109
Ticket CA SDM Producao - HighDisaster - Critical	109
Script de integração com SDM	110
Erro na integração	122
Módulo VMBIX	124
Validação	127
Itens disponíveis	128
Converter as OIDs de TRAP	134
Funcionamento SNMPTT	136
Itens customizados - External Scripts	140

Caseirao	140
Parâmetros do caseirao	140
Script caseirao.sh	142
Customização no agente	143
Caseirao_count	143
Parâmetros do caseirao_count	143
Script caseirao_count.sh	145
Customização no agente	147
Caseirao_count_dir	147
Parâmetros do caseirao_count_dir	147
Script caseirao_count_dir.sh	149
Customização no agente	151
Caseirao_ckfile	151
Parâmetros do caseirao_ckfile	151
Script caseirao_ckfile.sh	153
Customização no agente	155
Caseirao_filedate	156
Parâmetros do caseirao_filedate	156
Script caseirao_filedate.sh	157
Customização no agente	159
nt519_log.sh	159
Parâmetros do nt519_log.sh	160
Script nt519_log.sh	161
clusternode.sh	162
clusterresource.sh	162
clustergroup.sh	163
ltnPoolMemberStatServerCurConns.sh	163
sysCmTrafficGroupStatusFailoverStatus.sh	169

snmp_HR_proc_num.sh	173
sdm-check.sh	175
mountpoint.sh	179
Clear TRAP	181
Pré-requisitos	183
Instalação	185
Maintenance	224
Ticket Report	230
Acesso do usuário	230
Filtros	230
Grupos	230
Hosts	231
Triggers	232
Período	233
Funcionamento	234
reportPortoAcompanhamentoTicketsDaily_Insert.sh	234
reportPortoAcompanhamentoTicketsDaily_UpdateAck.sh	235
reportPortoAcompanhamentoTicketsDaily_UpdateEvent.sh	235
reportPortoAcompanhamentoTicketsDaily_DeleteTicket.sh	235
Instalação do módulo	235
Capacity	297
Configuração	297
Configuração dos hosts	297
Configuração do XML	298
Instalação	298
Backups	310
Banco de dados	312
Configuração	312

LI2456 e LI2457	312
LI2616	330
Apêndice	331
A - Instalação agente Zabbix Linux THPP	331
B - Instalação agente Zabbix Linux Produção	331
C - Instalação agente Zabbix Linux DMZ	331
D - Instalação agente Zabbix Windows THPP	331
E - Instalação agente Zabbix Windows Produção	331
F - Instalação agente Zabbix Windows DMZ	331
G - Instalação agente Zabbix Unix THPP	331
H - Instalação agente Zabbix Unix Produção	331
I - Instalação agente Zabbix HP-UX THPP	331
J - Instalação agente Zabbix HP-UX Produção	331
K - Instalação agente Zabbix Xenserver Produção	331
L- Troubleshooting agente Zabbix Linux	331
M- Troubleshooting agente Zabbix Windows	331
N- Troubleshooting agente Zabbix Unix	331
O- Failover Zabbix Proxy THPP	331
P- Failover Zabbix Proxy NIX	331
Q- Failover Zabbix Proxy Windows	331
R- Failover Zabbix Proxy SNMP	331
S- Failover Zabbix Proxy VMWARE	331
T- Failover Zabbix WEB	331
U- Failover Cluster Zabbix Server	331
V- Failover Cluster MySQL	331
W- Planilha Hosts x Interfaces	331
X- Planilha Hosts x Groups	332
Y- Planilha Hosts x Templates	332

Z- Planilha Templates Itens e Triggers	332
AA- Planilha Host x itens avulsos	332
AB- Planilha check WEB	332
AC- Planilha LLD Filters Custom	332
AD- Planilha Macros	332
AE- Planilha Regexp	332
AF- Planilha Value Mappings	332

Introdução

O Zabbix foi criado por Alexei Vladishev, e atualmente é mantido e suportado pela Zabbix SIA.

O Zabbix é uma solução de nível enterprise, de código aberto e com suporte a monitoração distribuída.

O Zabbix é um software que monitora vários parâmetros da rede, dos servidores e da saúde dos serviços. Utiliza-se de um mecanismo flexível de notificação que permite configurar alertas por e-mail para praticamente qualquer evento. As notificações permitem que se reaja rapidamente à problemas no ambiente. O Zabbix oferece excelentes recursos de relatórios e visualização de dados armazenados. Isso faz com que o Zabbix seja a ferramenta ideal para planejamento de capacidade.

O Zabbix suporta tanto “pooling” quanto “trapping”. Os relatórios e estatísticas do Zabbix, e seus parâmetros de configuração, estão acessíveis através de interface web. O uso de uma interface web garante que você possa avaliar o estado de sua rede e a saúde de seus servidores a partir de qualquer local. Quando corretamente configurado o Zabbix pode desempenhar papel importante na infraestrutura de monitoramento de TI. Estas características se aplicam tanto a pequenas organizações com poucos servidores quanto para grandes empresas, com milhares de servidores.

Zabbix é livre de custos. É desenvolvido e distribuído através da licença pública GPLv2. Isso garante que seu código-fonte seja distribuído e esteja disponível para o público em geral.

Ref: <https://www.zabbix.com/documentation/3.0/pt/manual/introduction/about>

Conceitos Zabbix

Zabbix Server

O Zabbix Server é o componente central da solução.

O servidor gerencia a coleta e recebimento de dados, calcula o estado das triggers, envia notificações aos usuários. Ele é o componente para o qual os agentes e proxies enviam dados sobre a disponibilidade, performance e integridade dos sistemas monitorados. O servidor também pode executar por si só verificações remotas nos dispositivos monitorados, estas verificações ocorrem quando se utiliza itens do tipo “verificação simples”.

O servidor gerencia o repositório central de configuração, estatísticas e armazenamento de dados operacionais, é ele quem irá alertar os administradores quando os incidentes ocorrerem.

As funcionalidades básicas de uma solução de monitoração baseada em Zabbix é distribuída em três componentes: Zabbix Server, interface web e banco de dados (SGDB).

Todas as informações de configuração da monitoração são armazenadas no banco de dados, tanto o Servidor quanto a Interface Web do Zabbix interagem com o SGBD. Por exemplo, quando você utiliza a interface web (ou a API) para adicionar itens, eles são salvos em uma tabela do SGDB. Em paralelo a isso o Zabbix Server, uma vez a cada minuto, irá buscar, na tabela de itens, a lista de itens que deverão ser monitorados. É por isso que pode demorar até dois minutos para que uma modificação feita na Interface Web comece a produzir efeitos na tela de dados recentes.

<https://www.zabbix.com/documentation/3.0/pt/manual/concepts/server>

Zabbix Proxy

O Zabbix Proxy é um processo que pode receber dados de um ou mais dispositivos monitorados e enviar ao Zabbix Server, basicamente ele funciona em nome do Zabbix Server (na visão do agente monitorado o Proxy passa a ser o Zabbix Server). Todo os dados recebidos são armazenados temporariamente (bufferizados), transferidos ao Zabbix Server que o Zabbix Proxy pertencer, sendo excluídos na sequência do armazenamento temporário do Proxy.

A utilização deste componente é opcional, mas normalmente é muito benéfica pois distribui a carga de monitoração normalmente atribuída ao Zabbix Server. Se toda a coleta de dados for feita através de Proxies o uso de CPU e de I/O no servidor responsável pelo Zabbix Server reduz significativamente.

O Zabbix Proxy é a solução ideal para a monitoração centralizada de localidades geograficamente dispersas e para redes gerenciadas remotamente.

O Zabbix Proxy requer um banco de dados em separado (normalmente um SQLite).

<https://www.zabbix.com/documentation/3.0/pt/manual/concepts/proxy>

Zabbix Agent

O agente Zabbix é instalado no dispositivo alvo da monitoração. Possui capacidade de monitorar de monitorar ativamente os recursos e aplicações locais (discos e partições, memória, estatísticas do processador, etc).

O agente concentra as informações locais sobre o dispositivo monitorado para posterior envio ao servidor ou proxy Zabbix (dependendo da configuração). Em caso de falhas (como um disco cheio ou a interrupção de um processo) o servidor Zabbix pode alertar ativamente os administradores do ambiente sobre o ocorrido.

Os agentes Zabbix são extremamente eficientes pois utilizam chamadas nativas do sistema operacional para obter as informações estatísticas.

<https://www.zabbix.com/documentation/3.0/pt/manual/concepts/agent>

Zabbix Sender

O Zabbix Sender é um utilitário de linha de comando que pode ser utilizado para enviar dados para o Zabbix Server.

Situações usuais de utilização:

- Traps de início ou finalização de scripts
- Envio de métricas de negócio diretamente a partir dos sistemas que os hospedam (sem coleta periódica)
- Envio de traps de incidentes não monitoráveis diretamente pelo Zabbix

<https://www.zabbix.com/documentation/3.0/pt/manual/concepts/sender>

Zabbix Get

O Zabbix Get é um utilitário de linha de comando que pode ser utilizado para se comunicar com o agente de monitoração do Zabbix e requisitar um dado do agente.

Este utilitário é normalmente utilizado em ações de desenvolvimento ou debug de chaves no agente.

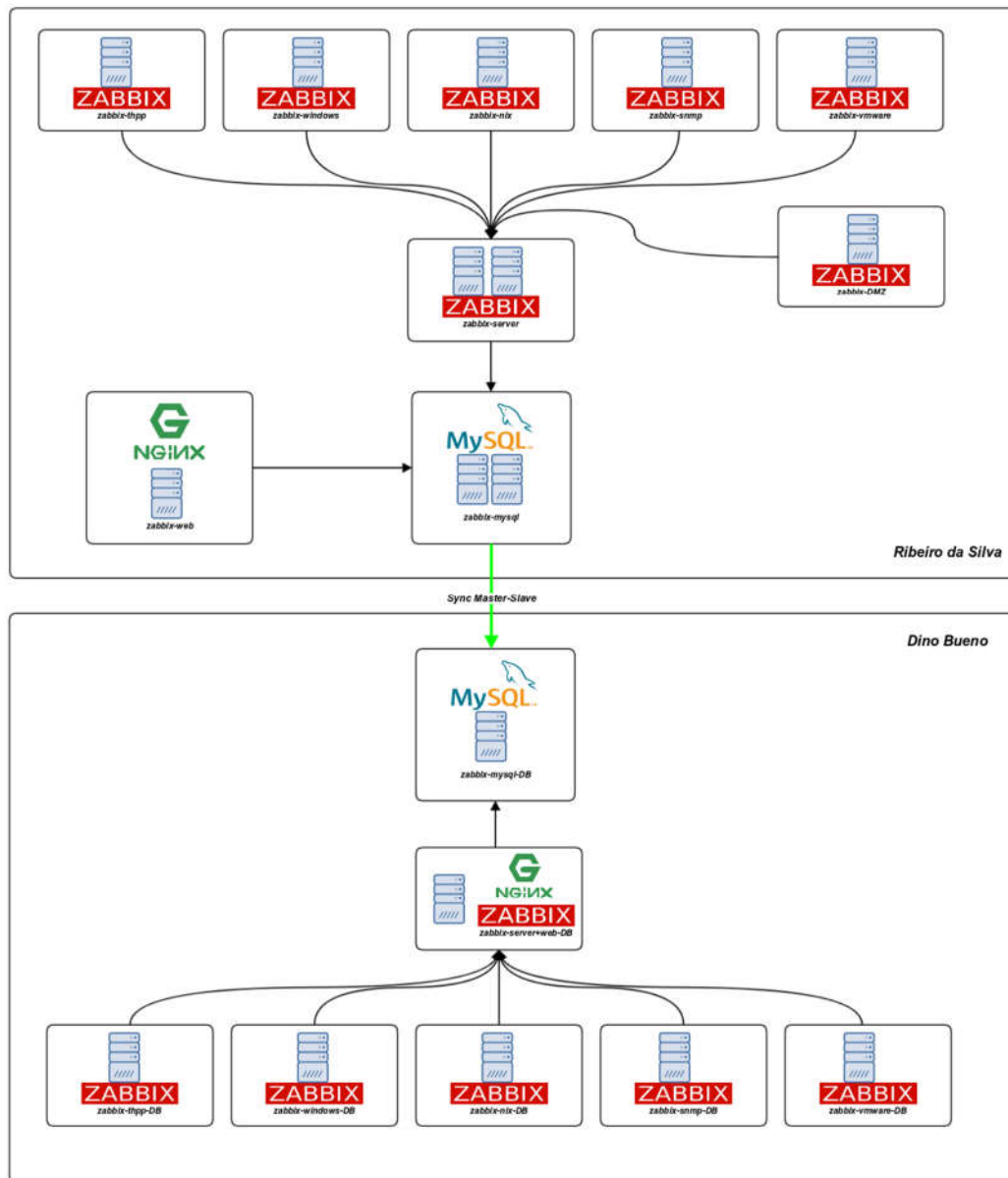
<https://www.zabbix.com/documentation/3.0/pt/manual/concepts/get>

Servidores

Servidor	Funcionalidade	MEM	CPU	Disco	Hostname	IP Publico	IP Servico	DNS/IP
ZABBIX Server Produção	Zabbix Server	16GB	12	120GB	LI2454	172.27.47.88	172.27.49.88	zabbix-server.portoseguro.brasil VIP: 172.27.46.211
ZABBIX Server Produção 2	Zabbix Server	16GB	12	120GB	LI2614	172.27.46.230	172.27.48.31	
Nginx	FrontEnd	20GB	16	60GB	LI2455	172.27.47.89	172.27.49.89	zabbix-web.portoseguro.brasil
MySQL - Master	Banco de dados	64GB	12	2TB**	LI2456	172.27.47.90	172.27.49.90	zabbix-mysql.portoseguro.brasil
MySQL - Slave*	Banco de dados	64GB	12	2TB**	LI2457	172.27.47.91	172.27.49.91	
THPP Proxy	Zabbix Proxy	8GB	4	60GB	LI2458	172.27.47.92	172.27.49.92	zabbix-thpp.portoseguro.brasil
Windows Proxy	Zabbix Proxy	16GB	8	60GB	LI2459	172.27.47.93	172.27.49.93	zabbix-windows.portoseguro.brasil
Linux Unix Proxy	Zabbix Proxy	16GB	8	60GB	LI2460	172.27.47.94	172.27.49.87	zabbix-nix.portoseguro.brasil
SNMP Proxy	Zabbix Proxy	16GB	16	60GB	LI2461	172.27.47.95	172.27.49.86	zabbix-snmpp.portoseguro.brasil
VMWARE Proxy	Zabbix Proxy	16GB	8	60GB	LI2497	172.27.47.79		zabbix-vmware.portoseguro.brasil
DMZ Proxy	Zabbix Proxy	4GB	2	60GB	i410	172.16.234.227		---
ZABBIX Server + Nginx DB	Zabbix Server + FrontEnd	20GB	16	120GB	LI2615	172.26.14.170	172.26.244.204	---
MySQL DB	Banco de dados	64GB	12	2TB	LI2616	172.26.14.171	172.26.244.205	---
THPP Proxy DB	Zabbix Proxy	8GB	4	60GB	LI2617	172.26.14.172	172.26.244.206	---
Windows Proxy DB	Zabbix Proxy	16GB	8	60GB	LI2618	172.26.14.173	172.26.244.207	---

Linux Unix Proxy DB	Zabbix Proxy	16GB	8	60GB	LI2619	172.26.14.174	172.26.244.208	---
SNMP Proxy DB	Zabbix Proxy	16GB	16	60GB	LI2620	172.26.14.175	172.26.244.209	---
VMWARE Proxy DB	Zabbix Proxy	16GB	8	60GB	LI2621	172.26.14.176	172.26.244.215	---
ZABBIX Server Homologacao		16GB	8	60GB	LI2128	172.26.24.86	172.27.48.81	zabbix-server-homologacao.portoseguro.brasil zabbix-homologacao.portoseguro.brasil
ZABBIX DEV		16GB	8	60GB	LI2129	172.26.24.87	172.27.48.82	
Mysql Homologação		16GB	8	500GB	LI2130	172.26.24.88	172.27.48.83	zabbix-mysql-homologacao.portoseguro.brasil

Topologia



Instalação

Para todos os procedimentos de instalação dos servidores, será necessário utilizar o tarball (pacotes-zabbix-portoseguro.tgz) que contém os pacotes utilizados e arquivos de configuração.

LI2454 - Zabbix-Server Master

1- Descompactar o tarball de instalação

```
tar xzvf pacotes-zabbix-portoseguro.tgz
```

2- Instalar o zabbix-agent

```
rpm -i zabbix-agent/*.rpm
```

3- Habilitar o zabbix-agent para iniciar automaticamente com o SO.

```
chkconfig zabbix-agent on
```

4- Substituir o arquivo /etc/zabbix/zabbix_agentd.conf

```
cp config/PROD/LI2454/zabbix_agentd.conf /etc/zabbix/zabbix_agentd.conf
```

5- Iniciar o serviço do zabbix-agent

```
/etc/init.d/zabbix-agent start
```

6- instalar as dependências do zabbix server

```
rpm -i --force zabbix-server/mysql-community-* yum install zabbix-  
server/*.rpm yum install gcc make libxml2-devel.x86_64 net-snmp net-snmp-  
devel.x86_64 libcurl-devel unixODBC unixODBC-devel
```

7- compilar o Zabbix-Server

```
cd zabbix-server/zabbix-3.0.9/  
  
./configure --enable-server --with-mysql --with-net-snmp --with-libcurl --with-  
libxml2 --with-unixodbc  
  
make install  
  
cp /usr/local/sbin/zabbix_server /usr/sbin/zabbix_server
```

8- criar os diretórios do Zabbix-Server e ajustar as permissões

```
mkdir -p /var/log/zabbix /var/run/zabbix /etc/zabbix/alertscripts  
/etc/zabbix/externalscripts/  
  
chown -R zabbix:zabbix /var/log/zabbix /var/run/zabbix /etc/zabbix/alertscripts  
/etc/zabbix/externalscripts/
```

9- copiar o script de inicialização do serviço

```
cp config/PROD/LI2454/initd-zabbix-server /etc/init.d/zabbix-server
```

10- copiar o arquivo de configuração do rotacionamento de logs


```
cp config/PROD/LI2454/logrotate-zabbix-server /etc/logrotate.d/zabbix-server
```

11- Desabilitar o zabbix-server para iniciar com o sistema (serviço será gerenciado pelo Heartbeat)

```
chkconfig zabbix-server off
```

12- Substituir o arquivo de configuração do zabbix-server

```
cp config/PROD/LI2454/zabbix_server.conf /etc/zabbix/zabbix_server.conf
```

13- Instalar o HAproxy

```
rpm -i haproxy/*.rpm
```

14- copiar o arquivo de configuração do HAproxy

```
cp config/PROD/LI2454/haproxy.cfg /etc/haproxy/haproxy.cfg
```

15- Habilitar o serviço do HAproxy para iniciar automaticamente com o SO.

```
chkconfig haproxy on
```

16- iniciar o HAproxy

```
/etc/init.d/haproxy start
```

17- Instalar o heartbeat e o mon

```
yum install heartbeat/*.rpm
```

18- copiar os arquivos de configuração do heartbeat e mon

```
cp config/PROD/LI2454/haresources /etc/ha.d/haresources  
  
cp config/PROD/LI2454/authkeys /etc/ha.d/authkeys  
  
cp config/PROD/LI2454/ha.cf /etc/ha.d/ha.cf  
  
cp config/PROD/LI2454/zabbix-server /etc/ha.d/resource.d/zabbix-server  
  
cp config/PROD/LI2454/mon.cf /etc/mon/mon.cf  
  
cp config/PROD/LI2454/heartbeat.alert /usr/lib64/mon/alert.d/heartbeat.alert  
  
cp config/PROD/LI2454/zabbix-server.monitor /usr/lib64/mon/mon.d/zabbix-server.monitor  
  
cp config/PROD/LI2454/hosts /etc/hosts
```

19- Habilitar o serviço do heartbeat e mon para iniciar automaticamente com o SO.

```
chkconfig heartbeat on  
  
chkconfig mon on
```

20- iniciar o heartbeat e mon

```
/etc/init.d/heartbeat start
```

```
/etc/init.d/mon start
```

LI2614 - Zabbix-Server Slave

1- Descompactar o tarball de instalação

```
tar xzvf pacotes-zabbix-portoseguro.tgz
```

2- Instalar o zabbix-agent

```
rpm -i zabbix-agent/*.rpm
```

3- Habilitar o zabbix-agent para iniciar automaticamente com o SO.

```
chkconfig zabbix-agent on
```

4- Substituir o arquivo /etc/zabbix/zabbix_agentd.conf

```
cp config/PROD/LI2614/zabbix_agentd.conf /etc/zabbix/zabbix_agentd.conf
```

5- Iniciar o serviço do zabbix-agent

```
/etc/init.d/zabbix-agent start
```

6- instalar as dependências do zabbix server

```
rpm -i --force zabbix-server/mysql-community-* yum install zabbix-server/*.rpm yum install gcc make libxml2-devel.x86_64 net-snmp net-snmp-devel.x86_64 libcurl-devel unixODBC unixODBC-devel
```

7- compilar o Zabbix-Server

```
cd zabbix-server/zabbix-3.0.9/

./configure --enable-server --with-mysql --with-net-snmp --with-libcurl --with-
libxml2 --with-unixodbc

make install

cp /usr/local/sbin/zabbix_server /usr/sbin/zabbix_server
```

8- criar os diretórios do Zabbix-Server e ajustar as permissões

```
mkdir -p /var/log/zabbix /var/run/zabbix /etc/zabbix/alertscripts
/etc/zabbix/externalscripts/

chown -R zabbix:zabbix /var/log/zabbix /var/run/zabbix /etc/zabbix/alertscripts
/etc/zabbix/externalscripts/
```

9- copiar o script de inicialização do serviço

```
cp config/PROD/LI2614/initd-zabbix-server /etc/init.d/zabbix-server
```

10- copiar o arquivo de configuração do rotacionamento de logs

```
cp config/PROD/LI2614/logrotate-zabbix-server /etc/logrotate.d/zabbix-server
```

11- Desabilitar o zabbix-server para iniciar com o sistema (serviço será gerenciado pelo Heartbeat)

```
chkconfig zabbix-server off
```

12- Substituir o arquivo de configuração do zabbix-server

```
cp config/PROD/LI2614/zabbix_server.conf /etc/zabbix/zabbix_server.conf
```

13- Instalar o HAproxy

```
rpm -i haproxy/*.rpm
```

14- copiar o arquivo de configuração do HAproxy

```
cp config/PROD/LI2614/haproxy.cfg /etc/haproxy/haproxy.cfg
```

15- Habilitar o serviço do HAproxy para iniciar automaticamente com o SO.

```
chkconfig haproxy on
```

16- iniciar o HAproxy

```
/etc/init.d/haproxy start
```

17- Instalar o heartbeat e o mon

```
yum install heartbeat/*.rpm
```

18- copiar os arquivos de configuração do heartbeat e mon

```
cp config/PROD/LI2614/haresources /etc/ha.d/haresources

cp config/PROD/LI2614/authkeys /etc/ha.d/authkeys

cp config/PROD/LI2614/ha.cf /etc/ha.d/ha.cf

cp config/PROD/LI2614/zabbix-server /etc/ha.d/resource.d/zabbix-server

cp config/PROD/LI2614/mon.cf /etc/mon/mon.cf

cp config/PROD/LI2614/heartbeat.alert /usr/lib64/mon/alert.d/heartbeat.alert

cp config/PROD/LI2614/zabbix-server.monitor /usr/lib64/mon/mon.d/zabbix-server.monitor

cp config/PROD/LI2614/hosts /etc/hosts
```

19- Habilitar o serviço do heartbeat e mon para iniciar automaticamente com o SO.

```
chkconfig heartbeat on

chkconfig mon on
```

20- iniciar o heartbeat e mon

```
/etc/init.d/heartbeat start

/etc/init.d/mon start
```

LI2455 - Zabbix-WEB

1- Descompactar o tarball de instalação

```
tar xzvf pacotes-zabbix-portoseguro.tgz
```

2- Instalar o zabbix-agent

```
rpm -i zabbix-agent/*.rpm
```

3- Habilitar o zabbix-agent para iniciar automaticamente com o SO.

```
chkconfig zabbix-agent on
```

4- Substituir o arquivo /etc/zabbix/zabbix_agentd.conf

```
cp config/PROD/LI2455/zabbix_agentd.conf /etc/zabbix/zabbix_agentd.conf
```

5- Iniciar o serviço do zabbix-agent

```
/etc/init.d/zabbix-agent start
```

6- Instalar o Nginx e PHP

```
yum localinstall zabbix-web/*.rpm
```

7- Copiar o diretório zabbix-web/frontend para /usr/share/zabbix


```
mkdir /usr/share/zabbix ; cp -R zabbix-web/frontend/* /usr/share/zabbix/
```

8- Habilitar os serviços do nginx e php-fpm para iniciar automaticamente com o SO.

```
chkconfig nginx on  
  
chkconfig php-fpm on
```

9- Adicionar o arquivo de configuracao do php e php-fpm e criar o diretório /var/lib/php/zabbix_php-fpm

```
cp config/PROD/LI2455/php.ini /etc/php.ini  
  
cp config/PROD/LI2455/zabbix.conf-php-fpm /etc/php-fpm.d/zabbix.conf  
  
mkdir /var/lib/php/zabbix_php-fpm && chown nginx:nginx  
/var/lib/php/zabbix_php-fpm
```

10- Adicionar o arquivo de configuração do Nginx e renomear o arquivo default.

```
cp config/PROD/LI2455/zabbix.conf /etc/nginx/conf.d/zabbix.conf  
  
cp config/PROD/LI2455/nginx.conf /etc/nginx/nginx.conf  
  
mv /etc/nginx/conf.d/default.conf /etc/nginx/conf.d/default.conf.old
```

11- Adicionar o arquivo de configuração do zabbix-web

```
cp config/PROD/LI2455/zabbix.conf.php
```

```
/usr/share/zabbix/conf/zabbix.conf.php
```

12- Iniciar o processo do Nginx e php-fpm

```
/etc/init.d/nginx start
```

```
/etc/init.d/php-fpm start
```

13- instalar o HAProxy

```
rpm -i haproxy/*.rpm
```

14- Habilitar o serviço do haproxy para iniciar automaticamente com o SO.

```
chkconfig haproxy on
```

15- Configurar o HAProxy

```
cp config/PROD/LI2455/haproxy.cfg /etc/haproxy/haproxy.cfg
```

16- Iniciar o HAProxy

```
/etc/init.d/haproxy start
```

LI2458 - Zabbix-Proxy THPP

1- Descompactar o tarball de instalação

```
tar xzvf pacotes-zabbix-portoseguro.tgz
```

2- Instalar o zabbix-agent

```
rpm -i zabbix-agent/*.rpm
```

3- Habilitar o zabbix-agent para iniciar para iniciar automaticamente com o SO.

```
chkconfig zabbix-agent on
```

4- Substituir o arquivo /etc/zabbix/zabbix_agentd.conf

```
cp config/PROD/LI2458/zabbix_agentd.conf /etc/zabbix/zabbix_agentd.conf
```

5- Iniciar o serviço do zabbix-agent

```
/etc/init.d/zabbix-agent start
```

6- Instalar o zabbix-proxy

```
rpm -i zabbix-proxy/*.rpm
```

7- Habilitar o serviço do zabbix-proxy para iniciar automaticamente com o SO.

```
chkconfig zabbix-proxy on
```

8- criar o diretório do banco de dados e ajustar o permissionamento

```
mkdir /opt/zabbix/ ; chown zabbix:zabbix /opt/zabbix
```

9- Substituir o arquivo /etc/zabbix/zabbix_proxy.conf

```
cp config/PROD/LI2458/zabbix_proxy.conf /etc/zabbix/zabbix_proxy.conf
```

10- Iniciar o serviço do Zabbix-Proxy

```
/etc/init.d/zabbix-proxy start
```

11- Instalar SNMPTT

```
yum install snmptt/*.rpm
```

12- Copiar os arquivos de configuração do SNMPTT e snmptrapd

```
cp config/PROD/LI2458/snmpd.conf /etc/snmp/snmpd.conf
```

```
cp config/PROD/LI2458/snmptt.conf /etc/snmp/snmptt.conf
```

```
cp config/PROD/LI2458/snmptrapd.conf /etc/snmp/snmptrapd.conf
```

```
cp config/PROD/LI2458/snmptt.conf /etc/snmp/snmptt.conf
```

```
cp config/PROD/LI2458/snmptt.ini /etc/snmp/snmptt.ini
```

13- Habilitar os serviços snmptt e snmptrapd para iniciar automaticamente com o SO.

```
chkconfig snmptt on
```

```
chkconfig snmptrapd on
```

14- Editar o arquivo `/etc/logrotate.d/zabbix-proxy` e adicionar o arquivo que receberá as traps para rotacionar.

```
/var/log/zabbix/zabbix_proxy.log /var/log/zabbix/snmptrap.log {  
    weekly  
    rotate 12  
    compress  
    delaycompress  
    missingok  
    notifempty  
    create 0664 zabbix zabbix  
}
```

15- Criar o arquivo que receberá as traps e ajustar o permissionamento.

```
test -e /var/log/zabbix/snmptrap.log || touch /var/log/zabbix/snmptrap.log ;  
chown zabbix:zabbix /var/log/zabbix/snmptrap.log
```

16- iniciar os serviços `snmpd` e `snmptrapd`

```
/etc/init.d/snmpd start  
  
/etc/init.d/snmptrapd start
```

LI2459 - Zabbix-Proxy Windows

1- Descompactar o tarball de instalação

```
tar xzvf pacotes-zabbix-portoseguro.tgz
```

2- Instalar o zabbix-agent

```
rpm -i zabbix-agent/*.rpm
```

3- Habilitar o zabbix-agent para iniciar para iniciar automaticamente com o SO.

```
chkconfig zabbix-agent on
```

4- Substituir o arquivo /etc/zabbix/zabbix_agentd.conf

```
cp config/PROD/LI2459/zabbix_agentd.conf /etc/zabbix/zabbix_agentd.conf
```

5- Iniciar o serviço do zabbix-agent

```
/etc/init.d/zabbix-agent start
```

6- Instalar o zabbix-proxy

```
rpm -i zabbix-proxy/*.rpm
```

7- Habilitar o serviço do zabbix-proxy para iniciar automaticamente com o SO.

```
chkconfig zabbix-proxy on
```

8- criar o diretório do banco de dados e ajustar o permissionamento

```
mkdir /opt/zabbix/ ; chown zabbix:zabbix /opt/zabbix
```

9- Substituir o arquivo /etc/zabbix/zabbix_proxy.conf

```
cp config/PROD/LI2459/zabbix_proxy.conf /etc/zabbix/zabbix_proxy.conf
```

10- Iniciar o serviço do Zabbix-Proxy

```
/etc/init.d/zabbix-proxy start
```

11- Instalar SNMPTT

```
yum install snmptt/*.rpm
```

12- Copiar os arquivos de configuração do SNMPTT e snmptrapd

```
cp config/PROD/LI2459/snmpd.conf /etc/snmp/snmpd.conf
```

```
cp config/PROD/LI2459/snmptt.conf /etc/snmp/snmptt.conf
```

```
cp config/PROD/LI2459/snmptrapd.conf /etc/snmp/snmptrapd.conf
```

```
cp config/PROD/LI2459/snmptt.conf /etc/snmp/snmptt.conf
```

```
cp config/PROD/LI2459/snmptt.ini /etc/snmp/snmptt.ini
```

13- Habilitar os serviços snmptt e snmptrapd para iniciar automaticamente com o SO.

```
chkconfig snmptt on  
chkconfig snmptrapd on
```

14- Editar o arquivo `/etc/logrotate.d/zabbix-proxy` e adicionar o arquivo que receberá as traps para rotacionar.

```
/var/log/zabbix/zabbix_proxy.log /var/log/zabbix/snmptrap.log {  
    weekly  
    rotate 12  
    compress  
    delaycompress  
    missingok  
    notifempty  
    create 0664 zabbix zabbix  
}
```

15- Criar o arquivo que receberá as traps e ajustar o permissionamento.

```
test -e /var/log/zabbix/snmptrap.log || touch /var/log/zabbix/snmptrap.log ;  
chown zabbix:zabbix /var/log/zabbix/snmptrap.log
```

16- iniciar os serviços `snmptt` e `snmptrapd`

```
/etc/init.d/snmptt start  
/etc/init.d/snmptrapd start
```

LI2460 - Zabbix-Proxy NIX

1- Descompactar o tarball de instalação

```
tar xzvf pacotes-zabbix-portoseguro.tgz
```


2- Instalar o zabbix-agent

```
rpm -i zabbix-agent/*.rpm
```

3- Habilitar o zabbix-agent para iniciar automaticamente com o SO.

```
chkconfig zabbix-agent on
```

4- Substituir o arquivo /etc/zabbix/zabbix_agentd.conf

```
cp config/PROD/LI2460/zabbix_agentd.conf /etc/zabbix/zabbix_agentd.conf
```

5- Iniciar o serviço do zabbix-agent

```
/etc/init.d/zabbix-agent start
```

6- Instalar o zabbix-proxy

```
rpm -i zabbix-proxy/*.rpm
```

7- Habilitar o serviço do zabbix-proxy para iniciar automaticamente com o SO.

```
chkconfig zabbix-proxy on
```

8- criar o diretório do banco de dados e ajustar o permissionamento

```
mkdir /opt/zabbix/ ; chown zabbix:zabbix /opt/zabbix
```

9- Substituir o arquivo /etc/zabbix/zabbix_proxy.conf

```
cp config/PROD/LI2460/zabbix_proxy.conf /etc/zabbix/zabbix_proxy.conf
```

10- Iniciar o serviço do Zabbix-Proxy

```
/etc/init.d/zabbix-proxy start
```

11- Instalar SNMPTT

```
yum install snmptt/*.rpm
```

12- Copiar os arquivos de configuração do SNMPTT e snmptrapd

```
cp config/PROD/LI2460/snmpd.conf /etc/snmp/snmpd.conf
```

```
cp config/PROD/LI2460/snmptt.conf /etc/snmp/snmptt.conf
```

```
cp config/PROD/LI2460/snmptrapd.conf /etc/snmp/snmptrapd.conf
```

```
cp config/PROD/LI2460/snmptt.conf /etc/snmp/snmptt.conf
```

```
cp config/PROD/LI2460/snmptt.ini /etc/snmp/snmptt.ini
```

13- Habilitar os serviços snmptt e snmptrapd para iniciar automaticamente com o SO.

```
chkconfig snmptt on
```

```
chkconfig snmptrapd on
```

14- Editar o arquivo `/etc/logrotate.d/zabbix-proxy` e adicionar o arquivo que receberá as traps para rotacionar.

```
/var/log/zabbix/zabbix_proxy.log /var/log/zabbix/snmptrap.log {  
    weekly  
    rotate 12  
    compress  
    delaycompress  
    missingok  
    notifempty  
    create 0664 zabbix zabbix  
}
```

15- Criar o arquivo que receberá as traps e ajustar o permissionamento.

```
test -e /var/log/zabbix/snmptrap.log || touch /var/log/zabbix/snmptrap.log ;  
chown zabbix:zabbix /var/log/zabbix/snmptrap.log
```

16- iniciar os serviços `snmpd` e `snmptrapd`

```
/etc/init.d/snmpd start  
  
/etc/init.d/snmptrapd start
```

LI2461 - Zabbix-Proxy SNMP

1- Descompactar o tarball de instalação

```
tar xzvf pacotes-zabbix-portoseguro.tgz
```

2- Instalar o `zabbix-agent`

```
rpm -i zabbix-agent/*.rpm
```

3- Habilitar o zabbix-agent para iniciar automaticamente com o SO.

```
chkconfig zabbix-agent on
```

4- Substituir o arquivo /etc/zabbix/zabbix_agentd.conf

```
cp config/PROD/LI2461/zabbix_agentd.conf /etc/zabbix/zabbix_agentd.conf
```

5- Iniciar o serviço do zabbix-agent

```
/etc/init.d/zabbix-agent start
```

6- Instalar o zabbix-proxy

```
rpm -i zabbix-proxy/*.rpm
```

7- Habilitar o serviço do zabbix-proxy para iniciar automaticamente com o SO.

```
chkconfig zabbix-proxy on
```

8- criar o diretório do banco de dados e ajustar o permissionamento

```
mkdir /opt/zabbix/ ; chown zabbix:zabbix /opt/zabbix
```

9- Substituir o arquivo /etc/zabbix/zabbix_proxy.conf

```
cp config/PROD/LI2461/zabbix_proxy.conf /etc/zabbix/zabbix_proxy.conf
```

10- Iniciar o serviço do Zabbix-Proxy

```
/etc/init.d/zabbix-proxy start
```

11- Instalar SNMPTT

```
yum install snmptt/*.rpm
```

12- Copiar os arquivos de configuração do SNMPTT e snmptrapd

```
cp config/PROD/LI2461/snmpd.conf /etc/snmp/snmpd.conf
```

```
cp config/PROD/LI2461/snmptt.conf /etc/snmp/snmptt.conf
```

```
cp config/PROD/LI2461/snmptrapd.conf /etc/snmp/snmptrapd.conf
```

```
cp config/PROD/LI2461/snmptt.conf /etc/snmp/snmptt.conf
```

```
cp config/PROD/LI2461/snmptt.ini /etc/snmp/snmptt.ini
```

13- Habilitar os serviços snmptt e snmptrapd para iniciar automaticamente com o SO.

```
chkconfig snmptt on
```

```
chkconfig snmptrapd on
```

14- Editar o arquivo `/etc/logrotate.d/zabbix-proxy` e adicionar o arquivo que receberá as traps para rotacionar.

```
/var/log/zabbix/zabbix_proxy.log /var/log/zabbix/snmptrap.log {  
    weekly  
    rotate 12  
    compress  
    delaycompress  
    missingok  
    notifempty  
    create 0664 zabbix zabbix  
}
```

15- Criar o arquivo que receberá as traps e ajustar o permissionamento.

```
test -e /var/log/zabbix/snmptrap.log || touch /var/log/zabbix/snmptrap.log ;  
chown zabbix:zabbix /var/log/zabbix/snmptrap.log
```

16- iniciar os serviços `snmpd` e `snmptrapd`

```
/etc/init.d/snmpd start  
  
/etc/init.d/snmptrapd start
```

LI2497 - Zabbix-Proxy VMWARE

1- Descompactar o tarball de instalação

```
tar xzvf pacotes-zabbix-portoseguro.tgz
```

2- Instalar o zabbix-agent

```
rpm -i zabbix-agent/*.rpm
```

3- Habilitar o zabbix-agent para iniciar automaticamente com o SO.

```
chkconfig zabbix-agent on
```

4- Substituir o arquivo /etc/zabbix/zabbix_agentd.conf

```
cp config/PROD/LI2497/zabbix_agentd.conf /etc/zabbix/zabbix_agentd.conf
```

5- Iniciar o serviço do zabbix-agent

```
/etc/init.d/zabbix-agent start
```

6- Instalar o zabbix-proxy

```
rpm -i zabbix-proxy/*.rpm
```

7- Habilitar o serviço do zabbix-proxy para iniciar automaticamente com o SO.

```
chkconfig zabbix-proxy on
```

8- criar o diretório do banco de dados e ajustar o permissionamento

```
mkdir /opt/zabbix/ ; chown zabbix:zabbix /opt/zabbix
```

9- Substituir o arquivo /etc/zabbix/zabbix_proxy.conf

```
cp config/PROD/LI2497/zabbix_proxy.conf /etc/zabbix/zabbix_proxy.conf
```

10- Iniciar o serviço do Zabbix-Proxy

```
/etc/init.d/zabbix-proxy start
```

11- Instalar SNMPTT

```
yum install snmptt/*.rpm
```

12- Copiar os arquivos de configuração do SNMPTT e snmptrapd

```
cp config/PROD/LI2497/snmpd.conf /etc/snmp/snmpd.conf
```

```
cp config/PROD/LI2497/snmptt.conf /etc/snmp/snmptt.conf
```

```
cp config/PROD/LI2497/snmptrapd.conf /etc/snmp/snmptrapd.conf
```

```
cp config/PROD/LI2497/snmptt.conf /etc/snmp/snmptt.conf
```

```
cp config/PROD/LI2497/snmptt.ini /etc/snmp/snmptt.ini
```

13- Habilitar os serviços snmptt e snmptrapd para iniciar automaticamente com o SO.

```
chkconfig snmptt on
```

```
chkconfig snmptrapd on
```


14- Editar o arquivo `/etc/logrotate.d/zabbix-proxy` e adicionar o arquivo que receberá as traps para rotacionar.

```
/var/log/zabbix/zabbix_proxy.log /var/log/zabbix/snmptrap.log {  
    weekly  
    rotate 12  
    compress  
    delaycompress  
    missingok  
    notifempty  
    create 0664 zabbix zabbix  
}
```

15- Criar o arquivo que receberá as traps e ajustar o permissionamento.

```
test -e /var/log/zabbix/snmptrap.log || touch /var/log/zabbix/snmptrap.log ;  
chown zabbix:zabbix /var/log/zabbix/snmptrap.log
```

16- iniciar os serviços `snmpd` e `snmptrapd`

```
/etc/init.d/snmpd start  
  
/etc/init.d/snmptrapd start
```

17- criar o diretório `/etc/vmbix` e copiar os arquivos de configuração

```
mkdir /etc/vmbix ; cp pacotes/config/PROD/LI2497/vmbix/* /etc/vmbix
```

18- copiar os scripts de inicialização dos serviços `vmbix`

```
cp pacotes/config/PROD/LI2497/vmbix_init.d/* /etc/init.d/
```

19- copiar os modulos do vmbix para o diretório /usr/lib/zabbix/modules/

```
test -e /usr/lib/zabbix/modules || mkdir /usr/lib/zabbix/modules  
  
cp pacotes/config/PROD/LI2497/vmbix_modulo/* /usr/lib/zabbix/modules/
```

20- copiar os arquivos de configuração dos módulos vmbix para o diretório do zabbix /etc/zabbix

```
cp pacotes/config/PROD/LI2497/vmbix_zabbix_conf/* /etc/zabbix/
```

21- copiar os binários do serviço vmbix

```
cp pacotes/config/PROD/LI2497/vmbix_sbin/* /usr/local/sbin/
```

22- adicionar os serviços vmbix para iniciar automaticamente com o SO.

```
chkconfig --add vmbixd_clesx001 && chkconfig vmbixd_clesx001 on  
  
chkconfig --add vmbixd_clesx002 && chkconfig vmbixd_clesx002 on  
  
chkconfig --add vmbixd_nt1959 && chkconfig vmbixd_nt1959 on
```

23- Iniciar os serviços do zabbix-agent e zabbix-proxy e vmbix

```
/etc/init.d/vmbixd_clesx001 start
```

```
/etc/init.d/vmbixd_clesx002 start
```

```
/etc/init.d/vmbixd_nt1959 start
```

LI2456 - Zabbix-MySQL Master

1- Descompactar o tarball de instalação

```
tar xzvf pacotes-zabbix-portoseguro.tgz
```

2- Instalar o zabbix-agent

```
rpm -i zabbix-agent/*.rpm
```

3- Habilitar o zabbix-agent para iniciar automaticamente com o SO.

```
chkconfig zabbix-agent on
```

4- Substituir o arquivo /etc/zabbix/zabbix_agentd.conf

```
cp config/PROD/LI2497/zabbix_agentd.conf /etc/zabbix/zabbix_agentd.conf
```

5- Iniciar o serviço do zabbix-agent

```
/etc/init.d/zabbix-agent start
```

6- criar o diretório /etc/zabbix/scripts e copiar o script de teste da base.

```
mkdir /etc/zabbix/scripts
```

```
cp config/PROD/LI2456/mysql-lag.php /etc/zabbix/scripts/mysql-lag.php
```

```
cp config/PROD/LI2456/zabbix-agent /etc/init.d/zabbix-agent
```

- o script de inicialização do zabbix-agent sobe o script php (mysql-lag.php) na porta 1234 que monitora o banco de dados (utilizado pelo HAProxy)

7- Instalar o MySQL

```
rpm -Uvh mysql/*.rpm
```

8- Parar o MySQL.

```
/etc/init.d/mysqld stop
```

9- Alterar o conf do MySQL

```
cp config/PROD/LI2456/my.cnf /etc/
```

10- Iniciar o MySQL.

```
/etc/init.d/mysqld start
```

11- Após a instalação será gerado uma senha temporária no arquivo /var/log/mysqld.log.

```
[root@xxxxxxx ~]# grep -i "temporary password" /var/log/mysqld.log
```

```
2017-10-10T20:52:42.055880Z 1 [Note] A temporary password is generated  
for root@localhost: <vgro1foAD-y
```

12- Alterar senha do MySQL

```
mysqladmin -u root -p'<vgro1foAD-y' password 'F1K@%Cxy6Bhf'
```

13- Acessar o Banco com essa senha.

```
mysql -uroot -pF1K@%Cxy6Bhf
```

14- Criar base de dados zabbix_db.

```
mysql> create database zabbix_db;
```

15- Aplicar as devidas permissões para o usuário zabbix_user.

```
mysql> uninstall plugin validate_password;
```

```
mysql> grant all privileges on zabbix_db.* to zabbix_user@'%' identified by  
'mk7UPB70Burv' ;
```

```
mysql> flush privileges;
```

16- Importar o sql para o banco de dados.

```
mysql -uroot zabbix_db -p'F1K@%Cxy6Bhf' < mysql/database/schema.sql
```

```
mysql -uroot zabbix_db -p'F1K@%Cxy6Bhf' < mysql/database/images.sql
```

```
mysql -uroot zabbix_db -p'F1K@%Cxy6Bhf' < mysql/database/data.sql
```

Criar particionamentos da base MySQL

17- Primeiramente execute os seguintes comandos na base

```
use zabbix_db;

ALTER TABLE `acknowledges` DROP PRIMARY KEY, ADD KEY
`acknowledges_0` (`acknowledgeid`);

ALTER TABLE `acknowledges` DROP FOREIGN KEY `c_acknowledges_1`,
DROP FOREIGN KEY `c_acknowledges_2`;

ALTER TABLE `alerts` DROP PRIMARY KEY, ADD KEY `alerts_0` (`alertid`);

ALTER TABLE `alerts` DROP FOREIGN KEY `c_alerts_1`, DROP FOREIGN
KEY `c_alerts_2`, DROP FOREIGN KEY `c_alerts_3`, DROP FOREIGN KEY
`c_alerts_4`;

ALTER TABLE `events` DROP PRIMARY KEY, ADD KEY `events_0`
(`eventid`);

ALTER TABLE `service_alarms` DROP PRIMARY KEY, ADD KEY
`service_alarms_0` (`servicealarmid`);

ALTER TABLE `service_alarms` DROP FOREIGN KEY
`c_service_alarms_1`;

ALTER TABLE `history_log` DROP PRIMARY KEY, ADD INDEX
`history_log_0` (`id`);

ALTER TABLE `history_log` DROP KEY `history_log_2`;

ALTER TABLE `history_text` DROP PRIMARY KEY, ADD INDEX
```

```
`history_text_0` (`id`);
```

```
ALTER TABLE `history_text` DROP KEY `history_text_2`;
```

18- Criar a tabela "manage_partitions".

```
use zabbix_db;
```

```
CREATE TABLE `manage_partitions` ( `tablename` VARCHAR(64) NOT  
NULL COMMENT 'Table name', `period` VARCHAR(64) NOT NULL  
COMMENT 'Period - daily or monthly', `keep_history` INT(3) UNSIGNED NOT  
NULL DEFAULT '1' COMMENT 'For how many days or months to keep the  
partitions', `last_updated` DATETIME DEFAULT NULL COMMENT 'When a  
partition was added last time', `comments` VARCHAR(128) DEFAULT '1'  
COMMENT 'Comments', PRIMARY KEY (`tablename`) ) ENGINE=INNODB;
```

19- Adicione o conteúdo na tabela.

```
INSERT INTO manage_partitions (tablename, period, keep_history,  
last_updated, comments) VALUES ('history', 'day', 8, now(), "");
```

```
INSERT INTO manage_partitions (tablename, period, keep_history,  
last_updated, comments) VALUES ('history_uint', 'day', 8, now(), "");
```

```
INSERT INTO manage_partitions (tablename, period, keep_history,  
last_updated, comments) VALUES ('history_str', 'day', 8, now(), "");
```

```
INSERT INTO manage_partitions (tablename, period, keep_history,  
last_updated, comments) VALUES ('history_text', 'day', 8, now(), "");
```



```
INSERT INTO manage_partitions (tablename, period, keep_history,  
last_updated, comments) VALUES ('history_log', 'day', 8, now(), "");  
  
INSERT INTO manage_partitions (tablename, period, keep_history,  
last_updated, comments) VALUES ('trends', 'month', 13, now(), "");  
  
INSERT INTO manage_partitions (tablename, period, keep_history,  
last_updated, comments) VALUES ('trends_uint', 'month', 13, now(), "");  
  
INSERT INTO manage_partitions (tablename, period, keep_history,  
last_updated, comments) VALUES ('acknowledges', 'month', 13, now(), "");  
  
INSERT INTO manage_partitions (tablename, period, keep_history,  
last_updated, comments) VALUES ('alerts', 'month', 13, now(), "");  
  
INSERT INTO manage_partitions (tablename, period, keep_history,  
last_updated, comments) VALUES ('events', 'month', 13, now(), "");  
  
INSERT INTO manage_partitions (tablename, period, keep_history,  
last_updated, comments) VALUES ('service_alarms', 'month', 13, now(), "");
```

20- Criar a procedure create_next_partitions.

```
DELIMITER $$  
  
USE `zabbix_db`$$  
  
DROP PROCEDURE IF EXISTS `create_next_partitions`$$
```

```
CREATE PROCEDURE `create_next_partitions`(IN_SCHEMANAME
VARCHAR(64))
BEGIN
    DECLARE TABLENAME_TMP VARCHAR(64);
    DECLARE PERIOD_TMP VARCHAR(12);
    DECLARE DONE INT DEFAULT 0;

    DECLARE get_prt_tables CURSOR FOR
        SELECT `tablename`, `period`
        FROM manage_partitions;
    DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;

    OPEN get_prt_tables;

    loop_create_part: LOOP
        IF DONE THEN
            LEAVE loop_create_part;
        END IF;

        FETCH get_prt_tables INTO TABLENAME_TMP, PERIOD_TMP;

        CASE WHEN PERIOD_TMP = 'day' THEN
            CALL `create_partition_by_day`(IN_SCHEMANAME,
TABLENAME_TMP);
            WHEN PERIOD_TMP = 'month' THEN
            CALL `create_partition_by_month`(IN_SCHEMANAME,
TABLENAME_TMP);
        ELSE
```

```
BEGIN
    ITERATE loop_create_part;
END;
END CASE;

UPDATE manage_partitions set last_updated = NOW() WHERE
tablename = TABLENAME_TMP;
END LOOP loop_create_part;

CLOSE get_prt_tables;
END$$

DELIMITER ;
```

21- Criar a procedure create_partition_by_day.

```
DELIMITER $$

USE `zabbix_db`$$

DROP PROCEDURE IF EXISTS `create_partition_by_day`$$

CREATE PROCEDURE `create_partition_by_day`(IN_SCHEMANAME
VARCHAR(64), IN_TABLENAME VARCHAR(64))
BEGIN
DECLARE ROWS_CNT INT UNSIGNED;
DECLARE BEGINTIME INT UNSIGNED;
DECLARE ENDTIME INT UNSIGNED;
DECLARE CKP INT UNSIGNED;
```

```
DECLARE PARTITIONNAME VARCHAR(16);

SET BEGINTIME = UNIX_TIMESTAMP( DATE(NOW()) ) + 24 * 60 * 60;
SET ENDTIME = BEGINTIME + 24 * 60 * 60;
SET PARTITIONNAME = FROM_UNIXTIME( BEGINTIME + 3 * 60 * 60,
'p%Y_%m_%d' );

SET CKP = FROM_UNIXTIME( ENDTIME, '%H' );

IF CKP > 0 THEN
SET ENDTIME = ENDTIME - CKP * 60 * 60;
END IF;

SELECT COUNT(*) INTO ROWS_CNT FROM information_schema.partitions
WHERE table_schema = IN_SCHEMANAME AND table_name =
IN_TABLENAME AND partition_name = PARTITIONNAME;

IF ROWS_CNT = 0 THEN
SET @SQL = CONCAT( 'ALTER TABLE `', IN_SCHEMANAME, '`.`',
IN_TABLENAME, '`',
' ADD PARTITION (PARTITION `', PARTITIONNAME, ' VALUES LESS THAN
( ', ENDTIME, ' ));' );
PREPARE STMT FROM @SQL;
EXECUTE STMT;
DEALLOCATE PREPARE STMT;
ELSE
SELECT CONCAT("partition '", PARTITIONNAME, "' for table
'", IN_SCHEMANAME, "`.`", IN_TABLENAME, "' already exists") AS result;
END IF;
```

```
END$$
```

```
DELIMITER ;
```

22- Criar a procedure create_partition_by_month.

```
DELIMITER $$
```

```
USE `zabbix_db`$$
```

```
DROP PROCEDURE IF EXISTS `create_partition_by_month`$$
```

```
CREATE PROCEDURE `create_partition_by_month`(IN_SCHEMANAME  
VARCHAR(64), IN_TABLENAME VARCHAR(64))
```

```
BEGIN
```

```
    DECLARE ROWS_CNT INT UNSIGNED;
```

```
    DECLARE BEGINTIME TIMESTAMP;
```

```
    DECLARE ENDTIME INT UNSIGNED;
```

```
    DECLARE PARTITIONNAME VARCHAR(16);
```

```
    SET BEGINTIME = DATE(NOW()) - INTERVAL DAY(NOW()) DAY +  
INTERVAL 1 DAY + INTERVAL 1 MONTH);
```

```
    SET PARTITIONNAME = DATE_FORMAT( BEGINTIME, 'p%Y_%m' );
```

```
    SET ENDTIME = UNIX_TIMESTAMP(BEGINTIME + INTERVAL 1  
MONTH);
```

```
    SELECT COUNT(*) INTO ROWS_CNT
```

```
        FROM information_schema.partitions
```

```
        WHERE table_schema = IN_SCHEMANAME AND table_name =
```

```

IN_TABLENAME AND partition_name = PARTITIONNAME;

IF ROWS_CNT = 0 THEN
    SET @SQL = CONCAT( 'ALTER TABLE `', IN_SCHEMANAME,
        '.', IN_TABLENAME, `,
        ' ADD PARTITION (PARTITION `', PARTITIONNAME, `
VALUES LESS THAN (', ENDTIME, ');' );
    PREPARE STMT FROM @SQL;
    EXECUTE STMT;
    DEALLOCATE PREPARE STMT;
ELSE
    SELECT CONCAT("partition `", PARTITIONNAME, "` for table
        '", IN_SCHEMANAME, ".", IN_TABLENAME, "` already exists") AS result;
END IF;
END$$

DELIMITER ;

```

23- Criar a procedure drop_partitions.

```

DELIMITER $$

USE `zabbix_db`$$

DROP PROCEDURE IF EXISTS `drop_partitions`$$

CREATE PROCEDURE `drop_partitions`(IN_SCHEMANAME
    VARCHAR(64))
BEGIN

```

```
DECLARE TABLENAME_TMP VARCHAR(64);
DECLARE PARTITIONNAME_TMP VARCHAR(64);
DECLARE VALUES_LESS_TMP INT;
DECLARE PERIOD_TMP VARCHAR(12);
DECLARE KEEP_HISTORY_TMP INT;
DECLARE KEEP_HISTORY_BEFORE INT;
DECLARE DONE INT DEFAULT 0;
DECLARE get_partitions CURSOR FOR
    SELECT p.`table_name`, p.`partition_name`,
LTRIM(RTRIM(p.`partition_description`)), mp.`period`, mp.`keep_history`
    FROM information_schema.partitions p
    JOIN manage_partitions mp ON mp.tablename = p.table_name
    WHERE p.table_schema = IN_SCHEMANAME
    ORDER BY p.table_name, p.subpartition_ordinal_position;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;

OPEN get_partitions;

loop_check_prt: LOOP
    IF DONE THEN
        LEAVE loop_check_prt;
    END IF;

    FETCH get_partitions INTO TABLENAME_TMP,
PARTITIONNAME_TMP, VALUES_LESS_TMP, PERIOD_TMP,
KEEP_HISTORY_TMP;
    CASE WHEN PERIOD_TMP = 'day' THEN
        SET KEEP_HISTORY_BEFORE =
```

```

UNIX_TIMESTAMP(DATE(NOW()) - INTERVAL KEEP_HISTORY_TMP
DAY));
    WHEN PERIOD_TMP = 'month' THEN
        SET KEEP_HISTORY_BEFORE =
UNIX_TIMESTAMP(DATE(NOW()) - INTERVAL KEEP_HISTORY_TMP
MONTH - INTERVAL DAY(NOW())-1 DAY));
    ELSE
    BEGIN
        ITERATE loop_check_prt;
    END;
END CASE;

IF KEEP_HISTORY_BEFORE >= VALUES_LESS_TMP THEN
    CALL drop_old_partition(IN_SCHEMANAME, TABLENAME_TMP,
PARTITIONNAME_TMP);
END IF;
END LOOP loop_check_prt;

CLOSE get_partitions;
END$$

DELIMITER ;

```

24- Criar a procedure drop_old_partition.

```

DELIMITER $$

USE `zabbix_db`$$

```



```
DROP PROCEDURE IF EXISTS `drop_old_partition`$$

CREATE PROCEDURE `drop_old_partition`(IN_SCHEMANAME
VARCHAR(64), IN_TABLENAME VARCHAR(64), IN_PARTITIONNAME
VARCHAR(64))
BEGIN
    DECLARE ROWS_CNT INT UNSIGNED;

    SELECT COUNT(*) INTO ROWS_CNT
        FROM information_schema.partitions
        WHERE table_schema = IN_SCHEMANAME AND table_name =
IN_TABLENAME AND partition_name = IN_PARTITIONNAME;

    IF ROWS_CNT = 1 THEN
        SET @SQL = CONCAT( 'ALTER TABLE `', IN_SCHEMANAME,
        ".", IN_TABLENAME, "`",
        ' DROP PARTITION `', IN_PARTITIONNAME, '`' );
        PREPARE STMT FROM @SQL;
        EXECUTE STMT;
        DEALLOCATE PREPARE STMT;
    ELSE
        SELECT CONCAT("partition `", IN_PARTITIONNAME, "` for table `",
IN_SCHEMANAME, ".", IN_TABLENAME, "` not exists") AS result;
    END IF;
END$$

DELIMITER ;
```

25- altere as tabelas para funcionar com o particionamento

```
ALTER TABLE `acknowledges` PARTITION BY RANGE ( clock) (PARTITION  
p2010_10 VALUES LESS THAN (UNIX_TIMESTAMP("2010-11-01  
00:00:00")) ENGINE = InnoDB);
```

```
ALTER TABLE `alerts` PARTITION BY RANGE ( clock) (PARTITION  
p2010_10 VALUES LESS THAN (UNIX_TIMESTAMP("2010-11-01  
00:00:00")) ENGINE = InnoDB);
```

```
ALTER TABLE `events` PARTITION BY RANGE ( clock) (PARTITION  
p2010_10 VALUES LESS THAN (UNIX_TIMESTAMP("2010-11-01  
00:00:00")) ENGINE = InnoDB);
```

```
ALTER TABLE `service_alarms` PARTITION BY RANGE ( clock)  
(PARTITION p2010_10 VALUES LESS THAN (UNIX_TIMESTAMP("2010-11-  
01 00:00:00")) ENGINE = InnoDB);
```

```
ALTER TABLE `trends` PARTITION BY RANGE ( clock) (PARTITION  
p2010_10 VALUES LESS THAN (UNIX_TIMESTAMP("2010-11-01  
00:00:00")) ENGINE = InnoDB);
```

```
ALTER TABLE `trends_uint` PARTITION BY RANGE ( clock) (PARTITION  
p2010_10 VALUES LESS THAN (UNIX_TIMESTAMP("2010-11-01  
00:00:00")) ENGINE = InnoDB);
```

```
ALTER TABLE `history` PARTITION BY RANGE ( clock) (PARTITION  
p2011_10_23 VALUES LESS THAN (UNIX_TIMESTAMP("2011-10-24  
00:00:00")) ENGINE = InnoDB);
```

```
ALTER TABLE `history_log` PARTITION BY RANGE ( clock) (PARTITION
p2011_10_23 VALUES LESS THAN (UNIX_TIMESTAMP("2011-10-24
00:00:00")) ENGINE = InnoDB);
```

```
ALTER TABLE `history_str` PARTITION BY RANGE ( clock) (PARTITION
p2011_10_23 VALUES LESS THAN (UNIX_TIMESTAMP("2011-10-24
00:00:00")) ENGINE = InnoDB);
```

```
ALTER TABLE `history_text` PARTITION BY RANGE ( clock) (PARTITION
p2011_10_23 VALUES LESS THAN (UNIX_TIMESTAMP("2011-10-24
00:00:00")) ENGINE = InnoDB);
```

```
ALTER TABLE `history_uint` PARTITION BY RANGE ( clock) (PARTITION
p2011_10_23 VALUES LESS THAN (UNIX_TIMESTAMP("2011-10-24
00:00:00")) ENGINE = InnoDB);
```

26- Execute as procedures para criar e deletar as novas partições:

```
CALL zabbix_db.create_next_partitions('zabbix_db');
CALL zabbix_db.drop_partitions('zabbix_db');
```

27- verifique se a partição foi criada com o mes e ano atual.

```
mysql> select table_name, partition_name, table_rows from
information_schema.partitions where table_name='trends_uint';
```

table_name	partition_name	table_rows
trends_uint	p2017_11	0

```
+-----+-----+-----+
1 row in set (0.11 sec)

mysql> select table_name, partition_name, table_rows from
information_schema.partitions where table_name='history';
+-----+-----+-----+
| table_name | partition_name | table_rows |
+-----+-----+-----+
| history   | p2017_10_11   |          0 |
+-----+-----+-----+
1 row in set (0.01 sec)
```

28- Agendar a execução (crontab) das procedures para manutenção das partições.

```
ZABBIX_PARTITION_LOG="/var/log/zabbix/zabbix_db_partition.log"
#particionamento do banco
00 */6 * * * root HOME=/root ; echo "$(date) Create" >>
$ZABBIX_PARTITION_LOG ; mysql --login-path=local zabbix_db -e "CALL
zabbix_db.create_next_partitions('zabbix_db');" &>>
$ZABBIX_PARTITION_LOG
00 */6 * * * root HOME=/root ; echo "$(date) Delete" >>
$ZABBIX_PARTITION_LOG ; mysql --login-path=local zabbix_db -e "CALL
zabbix_db.drop_partitions('zabbix_db');" &>> $ZABBIX_PARTITION_LOG
```

Replicação

29- Adicione a permissão de acesso para o Server Slave.

```
GRANT REPLICATION SLAVE ON *.* TO slave@172.27.47.91 identified by  
'88uKqUv1W7n7';
```

30- Realize o Backup da base inteira do MySQL.

```
mysqldump -uroot -p'F1K@%Cxy6Bhf' --all-databases --single-transaction  
2>/dev/null > all-db.sql
```

31- Transfira o backup para o servidor LI2457 (MySQL Slave)

```
scp all-db.sql root@172.27.47.91:
```

- Os próximos passos estarão no procedimento de instalação do servidor MySQL slave (LI2457).

LI2457 - Zabbix-MySQL Slave

1- Descompactar o tarball de instalação

```
tar xzvf pacotes-zabbix-portoseguro.tgz
```

2- Instalar o zabbix-agent

```
rpm -i zabbix-agent/*.rpm
```

3- Habilitar o zabbix-agent para iniciar automaticamente com o SO.

```
chkconfig zabbix-agent on
```

4- Substituir o arquivo /etc/zabbix/zabbix_agentd.conf

```
cp config/PROD/LI2497/zabbix_agentd.conf /etc/zabbix/zabbix_agentd.conf
```

5- Iniciar o serviço do zabbix-agent

```
/etc/init.d/zabbix-agent start
```

6- Instalar o MySQL

```
rpm -Uvh mysql/*.rpm
```

7- Parar o MySQL.

```
/etc/init.d/mysqld stop
```

8- Alterar o conf do MySQL padrão para o personalizado.

```
cp -fax config/PROD/LI2457/my.cnf /etc/
```

9- Iniciar o MySQL.

```
/etc/init.d/mysqld start
```

10- Após a instalação será gerado uma senha temporária no arquivo

/var/log/mysqld.log.

```
[root@xxxxxxx ~]# grep -i "temporary password" /var/log/mysqld.log 2017-10-10T20:52:42.055880Z 1 [Note] A temporary password is generated for root@localhost: <vgro1foAD-y
```

11- Alterar senha do MySQL

```
mysqladmin -u root -p'<vgro1foAD-y' password 'F1K@%Cxy6Bhf'
```

12- Realize o import do banco de dados do master copiado para o home do root.

```
mysql -uroot -p'F1K@%Cxy6Bhf' < all-db.sql
```

Configuração do MySQL Slave

1- Acesse o MySQL e execute o comando apontando para o servidor Master (LI2456)

```
CHANGE MASTER TO MASTER_HOST = '172.27.47.90', MASTER_USER = 'slave', MASTER_PASSWORD = '88uKqUv1W7n7', MASTER_LOG_FILE = '?', MASTER_LOG_POS = ?;
```

Alterando os '?' pelo valor real do MASTER_LOG_FILE e MASTER_LOG_POSITION, referente ao servidor Master, isso é obtido executando o seguinte comando no MASTER.

```
mysql> show master status;
```

2 -Inicie o Slave.

```
mysql> start slave
```

3- Verifique se a replicação está correta.

```
mysql> show slave status \G
***** 1. row *****

Slave_IO_State: Waiting for master to send event
Master_Host: 172.27.47.90
Master_User: slave
Master_Port: 3306
Connect_Retry: 60
Master_Log_File: mysql-bin.000288
Read_Master_Log_Pos: 579977324
Relay_Log_File: li2457-relay-bin.000794
Relay_Log_Pos: 579840042
Relay_Master_Log_File: mysql-bin.000288
Slave_IO_Running: Yes
Slave_SQL_Running: Yes
Replicate_Do_DB:
Replicate_Ignore_DB:
Replicate_Do_Table:
Replicate_Ignore_Table:
Replicate_Wild_Do_Table:
Replicate_Wild_Ignore_Table:
Last_Errno: 0
Last_Error:
Skip_Counter: 0
Exec_Master_Log_Pos: 579977324
Relay_Log_Space: 579840297
Until_Condition: None
Until_Log_File:
Until_Log_Pos: 0
Master_SSL_Allowed: No
```



```

Master_SSL_CA_File:
Master_SSL_CA_Path:
Master_SSL_Cert:
Master_SSL_Cipher:
Master_SSL_Key:
Seconds_Behind_Master: 0
Master_SSL_Verify_Server_Cert: No
Last_IO_Errno: 0
Last_IO_Error:
Last_SQL_Errno: 0
Last_SQL_Error:
Replicate_Ignore_Server_Ids:
Master_Server_Id: 1
Master_UUID: fa170eb7-c2c9-11e6-9354-005056b40f5a
Master_Info_File: /var/lib/mysql/master.info
SQL_Delay: 0
SQL_Remaining_Delay: NULL
Slave_SQL_Running_State: Slave has read all relay log; waiting for more
updates
Master_Retry_Count: 86400
Master_Bind:
Last_IO_Error_Timestamp:
Last_SQL_Error_Timestamp:
Master_SSL_Crl:
Master_SSL_Crlpath:
Retrieved_Gtid_Set:
Executed_Gtid_Set:
Auto_Position: 0
Replicate_Rewrite_DB:

```

```
Channel_Name:
Master_TLS_Version:
1 row in set (0,00 sec)
```

As informações mais importante acima são:

```
Slave_IO_Running: Yes
Slave_SQL_Running: Yes
Seconds_Behind_Master: 0
```

Depois configurar o slave, voltaremos ao servidor LI2457 e realizaremos a mesma configuração do passo 1, 2 e 3 para que a replicação seja bidirecional, alterado o MASTER_HOST, MASTER_LOG_FILE e MASTER_LOG_POSITION referente as informações do Slave.

Replicação MySQL DR

1- Crie o usuário da replicação

```
mysql> GRANT REPLICATION SLAVE ON *.* TO slave_dr@172.26.14.171
identified by 'il36dFi3JjZw';
```

2- Realize o Backup da base inteira do MySQL.

```
mysqldump -uroot -p'F1K@%Cxy6Bhf' --all-databases --single-transaction
2>/dev/null > all-db.sql
```

3- Transfira o backup para o servidor LI2616.

```
scp all-db.sql root@172.26.14.171:
```

- Os próximos passos estarão no procedimento de instalação do servidor mysql de DR (LI2616).

LI2616 - Zabbix-MySQL DR Dino Bueno

1- Descompactar o tarball de instalação

```
tar xzvf pacotes-zabbix-portoseguro.tgz
```

2- Instalar o zabbix-agent

```
rpm -i zabbix-agent/*.rpm
```

3- Habilitar o zabbix-agent para iniciar automaticamente com o SO.

```
chkconfig zabbix-agent on
```

4- Substituir o arquivo /etc/zabbix/zabbix_agentd.conf

```
cp config/DR/LI2616/zabbix_agentd.conf /etc/zabbix/zabbix_agentd.conf
```

5- Iniciar o serviço do zabbix-agent

```
/etc/init.d/zabbix-agent start
```

6- Instalar o MySQL

```
rpm -Uvh mysql/*.rpm
```

7- Parar o MySQL.

```
/etc/init.d/mysqld stop
```

8- Alterar o conf do MySQL padrão para o personalizado.

```
cp -fax config/PROD/LI2616/my.cnf /etc/
```

9- Iniciar o MySQL.

```
/etc/init.d/mysqld start
```

10- Após a instalação será gerado uma senha temporária no arquivo
/var/log/mysqld.log.

```
[root@xxxxxxx ~]# grep -i "temporary password" /var/log/mysqld.log 2017-10-10T20:52:42.055880Z 1 [Note] A temporary password is generated for root@localhost: <vgro1foAD-y
```

11- Alterar senha do MySQL

```
mysqladmin -u root -p'<vgro1foAD-y' password 'F1K@%Cxy6Bhf'
```

12- Realize o import do banco de dados do master copiado para o home do root.

```
mysql -uroot -p'F1K@%Cxy6Bhf' < all-db.sql
```

Configuração do MySQL Slave

13- Acesse o MySQL e execute o comando apontando para o servidor MasterSecundario (LI2457)

```
CHANGE MASTER TO MASTER_HOST = '172.27.47.91', MASTER_USER =  
'slave_dr', MASTER_PASSWORD = 'il36dFi3JjZw', MASTER_LOG_FILE =  
'?', MASTER_LOG_POS = ?;
```

Alterando os '?' pelo valor real do MASTER_LOG_FILE e MASTER_LOG_POSITION, referente ao servidor Master, isso é obtido executando o seguinte comando no MasterSecundario (LI2457).

```
mysql> show master status;
```

14 -Inicie o Slave.

```
mysql> start slave
```

15- Verifique se a replicação está correta.

```
mysql> show slave status \G  
***** 1. row *****  
  
Slave_IO_State: Waiting for master to send event  
Master_Host: 172.27.47.91  
Master_User: slave  
Master_Port: 3306  
Connect_Retry: 60  
Master_Log_File: mysql-bin.000288  
Read_Master_Log_Pos: 579977324  
Relay_Log_File: li2457-relay-bin.000794  
Relay_Log_Pos: 579840042  
Relay_Master_Log_File: mysql-bin.000288
```

```

Slave_IO_Running: Yes
Slave_SQL_Running: Yes
Replicate_Do_DB:
Replicate_Ignore_DB:
Replicate_Do_Table:
Replicate_Ignore_Table:
Replicate_Wild_Do_Table:
Replicate_Wild_Ignore_Table:
    Last_Errno: 0
    Last_Error:
    Skip_Counter: 0
Exec_Master_Log_Pos: 579977324
Relay_Log_Space: 579840297
Until_Condition: None
Until_Log_File:
Until_Log_Pos: 0
Master_SSL_Allowed: No
Master_SSL_CA_File:
Master_SSL_CA_Path:
Master_SSL_Cert:
Master_SSL_Cipher:
Master_SSL_Key:
Seconds_Behind_Master: 0
Master_SSL_Verify_Server_Cert: No
    Last_IO_Errno: 0
    Last_IO_Error:
    Last_SQL_Errno: 0
    Last_SQL_Error:
Replicate_Ignore_Server_Ids:

```

```
Master_Server_Id: 1
Master_UUID: fa170eb7-c2c9-11e6-9354-005056b40f5a
Master_Info_File: /var/lib/mysql/master.info
SQL_Delay: 0
SQL_Remaining_Delay: NULL
Slave_SQL_Running_State: Slave has read all relay log; waiting for more
updates
Master_Retry_Count: 86400
Master_Bind:
Last_IO_Error_Timestamp:
Last_SQL_Error_Timestamp:
Master_SSL_Crl:
Master_SSL_Crlpath:
Retrieved_Gtid_Set:
Executed_Gtid_Set:
Auto_Position: 0
Replicate_Rewrite_DB:
Channel_Name:
Master_TLS_Version:
1 row in set (0,00 sec)
```

As informações mais importante acima são:

```
Slave_IO_Running: Yes
Slave_SQL_Running: Yes
Seconds_Behind_Master: 0
```


LI2615 - Zabbix-Server+WEB DR (Dino Bueno)

1- Descompactar o tarball de instalação

```
tar xzvf pacotes-zabbix-portoseguro.tgz
```

2- Instalar o zabbix-agent

```
rpm -i zabbix-agent/*.rpm
```

3- Habilitar o zabbix-agent para iniciar automaticamente com o SO.

```
chkconfig zabbix-agent on
```

4- Substituir o arquivo /etc/zabbix/zabbix_agentd.conf

```
cp config/DR/LI2615/zabbix_agentd.conf /etc/zabbix/zabbix_agentd.conf
```

5- Iniciar o serviço do zabbix-agent

```
/etc/init.d/zabbix-agent start
```

6- instalar as dependências do zabbix server

```
rpm -i --force zabbix-server/mysql-community-* yum install zabbix-server/*.rpm yum install gcc make libxml2-devel.x86_64 net-snmp net-snmp-devel.x86_64 libcurl-devel unixODBC unixODBC-devel
```

7- compilar o Zabbix-Server

```
cd zabbix-server/zabbix-3.0.9/

./configure --enable-server --with-mysql --with-net-snmp --with-libcurl --with-
libxml2 --with-unixodbc

make install

cp /usr/local/sbin/zabbix_server /usr/sbin/zabbix_server
```

8- criar os diretórios do Zabbix-Server e ajustar as permissões

```
mkdir -p /var/log/zabbix /var/run/zabbix /etc/zabbix/alertscripts
/etc/zabbix/externalscripts/

chown -R zabbix:zabbix /var/log/zabbix /var/run/zabbix /etc/zabbix/alertscripts
/etc/zabbix/externalscripts/
```

9- copiar o script de inicialização do serviço

```
cp config/DR/LI2615/initd-zabbix-server /etc/init.d/zabbix-server
```

10- copiar o arquivo de configuração do rotacionamento de logs

```
cp config/DR/LI2615/logrotate-zabbix-server /etc/logrotate.d/zabbix-server
```

11- Desabilitar o zabbix-server para iniciar com o sistema (serviço será gerenciado pelo Heartbeat)

```
chkconfig zabbix-server off
```

12- Substituir o arquivo de configuração do zabbix-server

```
cp config/DR/LI2615/zabbix_server.conf /etc/zabbix/zabbix_server.conf
```

13- Instalar o HAproxy

```
rpm -i haproxy/*.rpm
```

14- copiar o arquivo de configuração do HAproxy

```
cp config/DR/LI2615/haproxy.cfg /etc/haproxy/haproxy.cfg
```

15- Habilitar o serviço do HAproxy para iniciar automaticamente com o SO.

```
chkconfig haproxy on
```

16- iniciar o HAproxy

```
/etc/init.d/haproxy start
```

17- Instalar o Nginx e PHP

```
yum localinstall zabbix-web/*.rpm
```

18- Copiar o diretório zabbix-web/frontend para /usr/share/zabbix

```
mkdir /usr/share/zabbix ; cp -R zabbix-web/frontend/* /usr/share/zabbix/
```

19- Habilitar os serviços do nginx e php-fpm para iniciar automaticamente com o SO.

```
chkconfig nginx on  
  
chkconfig php-fpm on
```

20- Adicionar o arquivo de configuração do php e php-fpm e criar o diretório /var/lib/php/zabbix_php-fpm

```
cp config/DR/LI2615/php.ini /etc/php.ini  
  
cp config/DR/LI2615/zabbix.conf-php-fpm /etc/php-fpm.d/zabbix.conf  
  
mkdir /var/lib/php/zabbix_php-fpm && chown nginx:nginx  
/var/lib/php/zabbix_php-fpm
```

10- Adicionar o arquivo de configuração do Nginx e renomear o arquivo default.

```
cp config/DR/LI2615/zabbix.conf /etc/nginx/conf.d/zabbix.conf  
  
cp config/DR/LI2615/nginx.conf /etc/nginx/nginx.conf  
  
mv /etc/nginx/conf.d/default.conf /etc/nginx/conf.d/default.conf.old
```

11- Adicionar o arquivo de configuração do zabbix-web

```
cp config/DR/LI2615/zabbix.conf.php /usr/share/zabbix/conf/zabbix.conf.php
```

12- Iniciar o processo do Nginx e php-fpm

```
/etc/init.d/nginx start
```

```
/etc/init.d/php-fpm start
```

LI2617 - Proxy THPP DR (Dino Bueno)

1- Descompactar o tarball de instalação

```
tar xzvf pacotes-zabbix-portoseguro.tgz
```

2- Instalar o zabbix-agent

```
rpm -i zabbix-agent/*.rpm
```

3- Habilitar o zabbix-agent para iniciar automaticamente com o SO.

```
chkconfig zabbix-agent on
```

4- Substituir o arquivo /etc/zabbix/zabbix_agentd.conf

```
cp config/DR/LI2617/zabbix_agentd.conf /etc/zabbix/zabbix_agentd.conf
```

5- Iniciar o serviço do zabbix-agent

```
/etc/init.d/zabbix-agent start
```

6- Instalar o zabbix-proxy

```
rpm -i zabbix-proxy/*.rpm
```

7- Habilitar o serviço do zabbix-proxy para iniciar automaticamente com o SO.

```
chkconfig zabbix-proxy on
```

8- criar o diretório do banco de dados e ajustar o permissionamento

```
mkdir /opt/zabbix/ ; chown zabbix:zabbix /opt/zabbix
```

9- Substituir o arquivo /etc/zabbix/zabbix_proxy.conf

```
cp config/DR/LI2617/zabbix_proxy.conf /etc/zabbix/zabbix_proxy.conf
```

10- Iniciar o serviço do Zabbix-Proxy

```
/etc/init.d/zabbix-proxy start
```

11- Instalar SNMPTT

```
yum install snmptt/*.rpm
```

12- Copiar os arquivos de configuração do SNMPTT e snmptrapd

```
cp config/DR/LI2617/snmpd.conf /etc/snmp/snmpd.conf
```

```
cp config/DR/LI2617/snmptt.conf /etc/snmp/snmptt.conf
```

```
cp config/DR/LI2617/snmptrapd.conf /etc/snmp/snmptrapd.conf
```

```
cp config/DR/LI2617/snmptt.conf /etc/snmp/snmptt.conf
```

```
cp config/DR/LI2617/snmptt.ini /etc/snmp/snmptt.ini
```

13- Habilitar os serviços snmptt e snmptrapd para iniciar automaticamente com o SO.

```
chkconfig snmptt on  
  
chkconfig snmptrapd on
```

14- Editar o arquivo `/etc/logrotate.d/zabbix-proxy` e adicionar o arquivo que receberá as traps para rotacionar.

```
/var/log/zabbix/zabbix_proxy.log /var/log/zabbix/snmptrap.log {  
    weekly  
    rotate 12  
    compress  
    delaycompress  
    missingok  
    notifempty  
    create 0664 zabbix zabbix  
}
```

15- Criar o arquivo que receberá as traps e ajustar o permissionamento.

```
test -e /var/log/zabbix/snmptrap.log || touch /var/log/zabbix/snmptrap.log ;  
chown zabbix:zabbix /var/log/zabbix/snmptrap.log
```

16- iniciar os serviços snmptt e snmptrapd

```
/etc/init.d/snmptt start  
  
/etc/init.d/snmptrapd start
```

LI2618 - Proxy Windows DR (Dino Bueno)

1- Descompactar o tarball de instalação


```
tar xzvf pacotes-zabbix-portoseguro.tgz
```

2- Instalar o zabbix-agent

```
rpm -i zabbix-agent/*.rpm
```

3- Habilitar o zabbix-agent para iniciar para iniciar automaticamente com o SO.

```
chkconfig zabbix-agent on
```

4- Substituir o arquivo /etc/zabbix/zabbix_agentd.conf

```
cp config/DR/LI2618/zabbix_agentd.conf /etc/zabbix/zabbix_agentd.conf
```

5- Iniciar o serviço do zabbix-agent

```
/etc/init.d/zabbix-agent start
```

6- Instalar o zabbix-proxy

```
rpm -i zabbix-proxy/*.rpm
```

7- Habilitar o serviço do zabbix-proxy para iniciar automaticamente com o SO.

```
chkconfig zabbix-proxy on
```

8- criar o diretório do banco de dados e ajustar o permissionamento

```
mkdir /opt/zabbix/ ; chown zabbix:zabbix /opt/zabbix
```

9- Substituir o arquivo /etc/zabbix/zabbix_proxy.conf

```
cp config/DR/LI2618/zabbix_proxy.conf /etc/zabbix/zabbix_proxy.conf
```

10- Iniciar o serviço do Zabbix-Proxy

```
/etc/init.d/zabbix-proxy start
```

11- Instalar SNMPTT

```
yum install snmptt/*.rpm
```

12- Copiar os arquivos de configuração do SNMPTT e snmptrapd

```
cp config/DR/LI2618/snmpd.conf /etc/snmp/snmpd.conf
```

```
cp config/DR/LI2618/snmptt.conf /etc/snmp/snmptt.conf
```

```
cp config/DR/LI2618/snmptrapd.conf /etc/snmp/snmptrapd.conf
```

```
cp config/DR/LI2618/snmptt.conf /etc/snmp/snmptt.conf
```

```
cp config/DR/LI2618/snmptt.ini /etc/snmp/snmptt.ini
```

13- Habilitar os serviços snmptt e snmptrapd para iniciar automaticamente com o SO.

```
chkconfig snmptt on
```

```
chkconfig snmptrapd on
```

14- Editar o arquivo `/etc/logrotate.d/zabbix-proxy` e adicionar o arquivo que receberá as traps para rotacionar.

```
/var/log/zabbix/zabbix_proxy.log /var/log/zabbix/snmptrap.log {  
    weekly  
    rotate 12  
    compress  
    delaycompress  
    missingok  
    notifempty  
    create 0664 zabbix zabbix  
}
```

15- Criar o arquivo que receberá as traps e ajustar o permissionamento.

```
test -e /var/log/zabbix/snmptrap.log || touch /var/log/zabbix/snmptrap.log ;  
chown zabbix:zabbix /var/log/zabbix/snmptrap.log
```

16- iniciar os serviços `snmpd` e `snmptrapd`

```
/etc/init.d/snmpd start  
  
/etc/init.d/snmptrapd start
```

LI2619 - Proxy Linux|Unix DR (Dino Bueno)

1- Descompactar o tarball de instalação

```
tar xzvf pacotes-zabbix-portoseguro.tgz
```

2- Instalar o zabbix-agent

```
rpm -i zabbix-agent/*.rpm
```

3- Habilitar o zabbix-agent para iniciar para iniciar automaticamente com o SO.

```
chkconfig zabbix-agent on
```

4- Substituir o arquivo /etc/zabbix/zabbix_agentd.conf

```
cp config/DR/LI2619/zabbix_agentd.conf /etc/zabbix/zabbix_agentd.conf
```

5- Iniciar o serviço do zabbix-agent

```
/etc/init.d/zabbix-agent start
```

6- Instalar o zabbix-proxy

```
rpm -i zabbix-proxy/*.rpm
```

7- Habilitar o serviço do zabbix-proxy para iniciar automaticamente com o SO.

```
chkconfig zabbix-proxy on
```

8- criar o diretório do banco de dados e ajustar o permissionamento

```
mkdir /opt/zabbix/ ; chown zabbix:zabbix /opt/zabbix
```

9- Substituir o arquivo /etc/zabbix/zabbix_proxy.conf

```
cp config/DR/LI2619/zabbix_proxy.conf /etc/zabbix/zabbix_proxy.conf
```

10- Iniciar o serviço do Zabbix-Proxy

```
/etc/init.d/zabbix-proxy start
```

11- Instalar SNMPTT

```
yum install snmptt/*.rpm
```

12- Copiar os arquivos de configuração do SNMPTT e snmptrapd

```
cp config/DR/LI2619/snmpd.conf /etc/snmp/snmpd.conf
```

```
cp config/DR/LI2619/snmptt.conf /etc/snmp/snmptt.conf
```

```
cp config/DR/LI2619/snmptrapd.conf /etc/snmp/snmptrapd.conf
```

```
cp config/DR/LI2619/snmptt.conf /etc/snmp/snmptt.conf
```

```
cp config/DR/LI2619/snmptt.ini /etc/snmp/snmptt.ini
```

13- Habilitar os serviços snmptt e snmptrapd para iniciar automaticamente com o SO.

```
chkconfig snmptt on
```

```
chkconfig snmptrapd on
```

14- Editar o arquivo `/etc/logrotate.d/zabbix-proxy` e adicionar o arquivo que receberá as traps para rotacionar.

```
/var/log/zabbix/zabbix_proxy.log /var/log/zabbix/snmptrap.log {  
    weekly  
    rotate 12  
    compress  
    delaycompress  
    missingok  
    notifempty  
    create 0664 zabbix zabbix  
}
```

15- Criar o arquivo que receberá as traps e ajustar o permissionamento.

```
test -e /var/log/zabbix/snmptrap.log || touch /var/log/zabbix/snmptrap.log ;  
chown zabbix:zabbix /var/log/zabbix/snmptrap.log
```

16- iniciar os serviços `snmpd` e `snmptrapd`

```
/etc/init.d/snmpd start  
  
/etc/init.d/snmptrapd start
```

LI2620 - Proxy SNMP DR (Dino Bueno)

1- Descompactar o tarball de instalação

```
tar xzvf pacotes-zabbix-portoseguro.tgz
```

2- Instalar o `zabbix-agent`

```
rpm -i zabbix-agent/*.rpm
```

3- Habilitar o zabbix-agent para iniciar automaticamente com o SO.

```
chkconfig zabbix-agent on
```

4- Substituir o arquivo /etc/zabbix/zabbix_agentd.conf

```
cp config/DR/LI2620/zabbix_agentd.conf /etc/zabbix/zabbix_agentd.conf
```

5- Iniciar o serviço do zabbix-agent

```
/etc/init.d/zabbix-agent start
```

6- Instalar o zabbix-proxy

```
rpm -i zabbix-proxy/*.rpm
```

7- Habilitar o serviço do zabbix-proxy para iniciar automaticamente com o SO.

```
chkconfig zabbix-proxy on
```

8- criar o diretório do banco de dados e ajustar o permissionamento

```
mkdir /opt/zabbix/ ; chown zabbix:zabbix /opt/zabbix
```

9- Substituir o arquivo /etc/zabbix/zabbix_proxy.conf

```
cp config/DR/LI2620/zabbix_proxy.conf /etc/zabbix/zabbix_proxy.conf
```

10- Iniciar o serviço do Zabbix-Proxy

```
/etc/init.d/zabbix-proxy start
```

11- Instalar SNMPTT

```
yum install snmptt/*.rpm
```

12- Copiar os arquivos de configuração do SNMPTT e snmptrapd

```
cp config/DR/LI2620/snmpd.conf /etc/snmp/snmpd.conf
```

```
cp config/DR/LI2620/snmptt.conf /etc/snmp/snmptt.conf
```

```
cp config/DR/LI2620/snmptrapd.conf /etc/snmp/snmptrapd.conf
```

```
cp config/DR/LI2620/snmptt.conf /etc/snmp/snmptt.conf
```

```
cp config/DR/LI2620/snmptt.ini /etc/snmp/snmptt.ini
```

13- Habilitar os serviços snmptt e snmptrapd para iniciar automaticamente com o SO.

```
chkconfig snmptt on
```

```
chkconfig snmptrapd on
```


14- Editar o arquivo `/etc/logrotate.d/zabbix-proxy` e adicionar o arquivo que receberá as traps para rotacionar.

```
/var/log/zabbix/zabbix_proxy.log /var/log/zabbix/snmptrap.log {  
    weekly  
    rotate 12  
    compress  
    delaycompress  
    missingok  
    notifempty  
    create 0664 zabbix zabbix  
}
```

15- Criar o arquivo que receberá as traps e ajustar o permissionamento.

```
test -e /var/log/zabbix/snmptrap.log || touch /var/log/zabbix/snmptrap.log ;  
chown zabbix:zabbix /var/log/zabbix/snmptrap.log
```

16- iniciar os serviços `snmpd` e `snmptrapd`

```
/etc/init.d/snmpd start  
  
/etc/init.d/snmptrapd start
```

LI2621 - Proxy VMWARE DR (Dino Bueno)

1- Descompactar o tarball de instalação

```
tar xzvf pacotes-zabbix-portoseguro.tgz
```

2- Instalar o `zabbix-agent`

```
rpm -i zabbix-agent/*.rpm
```

3- Habilitar o zabbix-agent para iniciar automaticamente com o SO.

```
chkconfig zabbix-agent on
```

4- Substituir o arquivo /etc/zabbix/zabbix_agentd.conf

```
cp config/DR/LI2621/zabbix_agentd.conf /etc/zabbix/zabbix_agentd.conf
```

5- Iniciar o serviço do zabbix-agent

```
/etc/init.d/zabbix-agent start
```

6- Instalar o zabbix-proxy

```
rpm -i zabbix-proxy/*.rpm
```

7- Habilitar o serviço do zabbix-proxy para iniciar automaticamente com o SO.

```
chkconfig zabbix-proxy on
```

8- criar o diretório do banco de dados e ajustar o permissionamento

```
mkdir /opt/zabbix/ ; chown zabbix:zabbix /opt/zabbix
```

9- Substituir o arquivo /etc/zabbix/zabbix_proxy.conf

```
cp config/DR/LI2621/zabbix_proxy.conf /etc/zabbix/zabbix_proxy.conf
```

10- Iniciar o serviço do Zabbix-Proxy

```
/etc/init.d/zabbix-proxy start
```

11- Instalar SNMPTT

```
yum install snmptt/*.rpm
```

12- Copiar os arquivos de configuração do SNMPTT e snmptrapd

```
cp config/DR/LI2621/snmpd.conf /etc/snmp/snmpd.conf
```

```
cp config/DR/LI2621/snmptt.conf /etc/snmp/snmptt.conf
```

```
cp config/DR/LI2621/snmptrapd.conf /etc/snmp/snmptrapd.conf
```

```
cp config/DR/LI2621/snmptt.conf /etc/snmp/snmptt.conf
```

```
cp config/DR/LI2621/snmptt.ini /etc/snmp/snmptt.ini
```

13- Habilitar os serviços snmptt e snmptrapd para iniciar automaticamente com o SO.

```
chkconfig snmptt on
```

```
chkconfig snmptrapd on
```

14- Editar o arquivo `/etc/logrotate.d/zabbix-proxy` e adicionar o arquivo que receberá as traps para rotacionar.

```
/var/log/zabbix/zabbix_proxy.log /var/log/zabbix/snmptrap.log {  
    weekly  
    rotate 12  
    compress  
    delaycompress  
    missingok  
    notifempty  
    create 0664 zabbix zabbix  
}
```

15- Criar o arquivo que receberá as traps e ajustar o permissionamento.

```
test -e /var/log/zabbix/snmptrap.log || touch /var/log/zabbix/snmptrap.log ;  
chown zabbix:zabbix /var/log/zabbix/snmptrap.log
```

16- iniciar os serviços `snmpd` e `snmptrapd`

```
/etc/init.d/snmpd start  
  
/etc/init.d/snmptrapd start
```

17- criar o diretório `/etc/vmbix` e copiar os arquivos de configuração

```
mkdir /etc/vmbix ; cp pacotes/config/DR/LI2621/vmbix/* /etc/vmbix
```

18- copiar os scripts de inicialização dos serviços `vmbix`

```
cp pacotes/config/DR/LI2621/vmbix_init.d/* /etc/init.d/
```

19- copiar os modulos do vmbix para o diretório /usr/lib/zabbix/modules/

```
test -e /usr/lib/zabbix/modules || mkdir /usr/lib/zabbix/modules  
  
cp pacotes/config/DR/LI2621/vmbix_modulo/* /usr/lib/zabbix/modules/
```

20- copiar os arquivos de configuração dos módulos vmbix para o diretório do zabbix /etc/zabbix

```
cp pacotes/config/DR/LI2621/vmbix_zabbix_conf/* /etc/zabbix/
```

21- copiar os binários do serviço vmbix

```
cp pacotes/config/DR/LI2621/vmbix_sbin/* /usr/local/sbin/
```

22- adicionar os servicos vmbix para iniciar automaticamente com o SO.

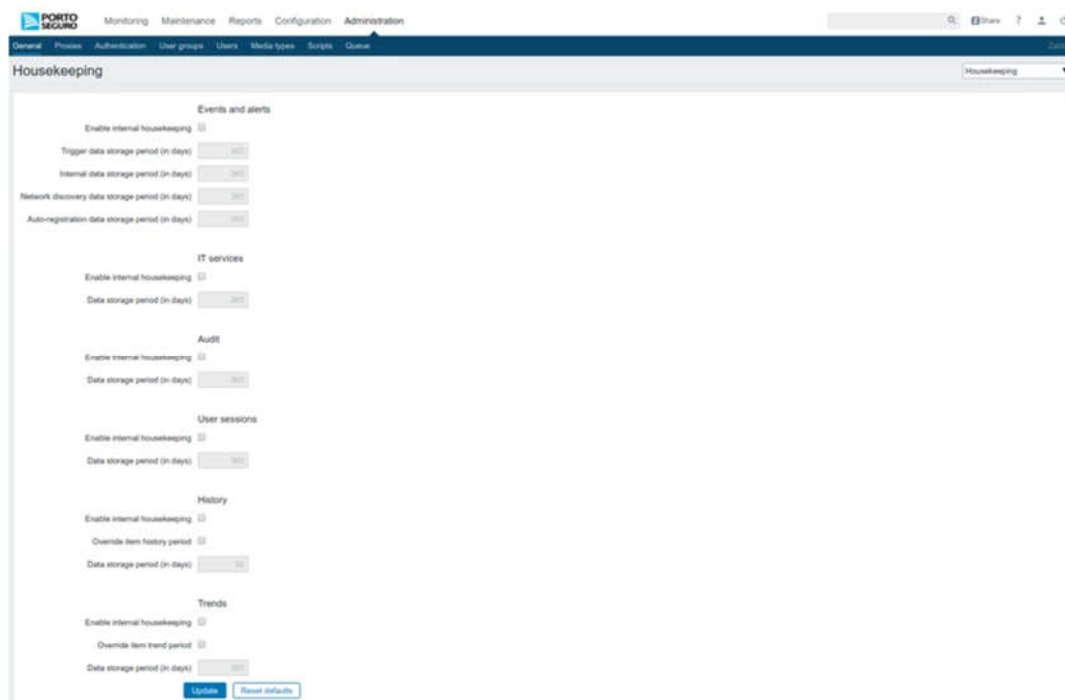
```
chkconfig --add vmbixd_clesx001 && chkconfig vmbixd_clesx001 on  
  
chkconfig --add vmbixd_clesx002 && chkconfig vmbixd_clesx002 on  
  
chkconfig --add vmbixd_nt1959 && chkconfig vmbixd_nt1959 on
```

Housekeeping

O processo de housekeeping é responsável pela limpeza dos dados antigos das tabelas do zabbix. Esse processo de limpeza sobrecarrega muito o banco

de dados causando lentidão, podendo gerar diversos falsos positivos, por este motivo optamos por desabilitar esse processo e trabalhar com o particionamento de tabelas.

Para configurar o particionamento de tabelas é necessário desabilitar o housekeeping no frontend do zabbix em Administration → General → Housekeeping, e no arquivo de configuração do zabbix-server /etc/zabbix/zabbix_server.conf.



Alterar o parâmetro HousekeepingFrequency no arquivo /etc/zabbix/zabbix_server.conf para 0.

```
HousekeepingFrequency=0
```

Os passos para configuração das partições, procedures e agendamentos estão listados no procedimento de instalação do banco de dados.

Integração com LDAP

A integração com o LDAP é configurado em Administration → Authentication, selecionar o Default authentication para LDAP e configurar os campos com os valores abaixo:

Host	portoseguro.brasil
Port	3268
BaseDN	dc=portoseguro,dc=brasil
Search attribute	sAMAccountName
Bind DN	cn=Conta de Serviço do CA EEM,ou=Contas de Serviço,ou=Administração,ou=Porto,dc=portoseguro,dc=brasil
Bind Password	caeem01
Test Login	usrcaqos
Test User Pass	canetqos01

PORTO SEGURO Monitoring Maintenance Reports Configuration **Administration**

General Proxies **Authentication** User groups Users Media types Scripts Queue Zabbix

Authentication

Default authentication: Internal **LDAP** HTTP

LDAP host: portoseguro.brasil

Port: 3268

Base DN: dc=portoseguro,dc=brasil

Search attribute: sAMAccountName

Bind DN: cn=Conta de Serviço do CA EEM,ou=Contas de Serviço,ou=Administração,ou=Por

Bind password: Change password

Test authentication: [must be a valid LDAP user]

Login: Admin

User password:

Update Test

https://www.zabbix.com/documentation/3.0/pt/manual/web_interface/frontend_sections/administration/authentication

Integração com SDM

Os tickets serão abertos no PortoSDM após a trigger permanecer por 3 minutos com o status de PROBLEM, e caso exista um ticket do mesmo problema (triggerID) com o status OPEN no período de 8 horas (configurável), será adicionado um comentário no mesmo ticket.

Categorias do SDM

Os tickets devem ser abertos nas categorias e prioridades corretas baseado na severidade do alerta e o tipo de ambiente:

Ambiente	Severidade Zabbix	Categoria	ID Categoria	Prioridade	ID Prioridade
Produção	Warning	Monitoração.Alerta.Minor	pcat:403911	2- Baixa	pri:501
Produção	Average	Monitoração.Alerta.Major	pcat:403910	3- Media	pri:502
Produção	High e Disaster	Monitoração.Alerta.Critical	pcat:403909	4- Alta	pri:503
Homologação	Warning	Infraestrutura.Monitoração.Homologação.Alert.Minor	pcat:416591	2- Baixa	pri:501
Homologação	Average	Infraestrutura.Monitoração.Homologação.Alert.Major	pcat:416590	3- Media	pri:502
Homologação	High e Disaster	Infraestrutura.Monitoração.Homologação.Alert.Critical	pcat:416589	4- Alta	pri:503

Media Type

Para cada tipo de severidade e ambiente foi criado uma media type. Essas media type devem conter os 10 parâmetros abaixo:

1	Assunto do alerta (Nao alterar)
2	Mensagem do alerta (Nao alterar)
3	Categoria que o ticket será aberto
4	tempo de SLA do ticket (segundos)
5	IP do CA SDM
6	Usuário do CA SDM
7	Senha do usuario CA SDM
8	Usuário para integração com API Zabbix
9	Senha do usuário para integração com API Zabbix.
10	Prioridade do ticket

CA Open Request - Prod Minor

Configuração da media type.

Name	CA Open Request - Prod Minor
Type	Script
Script Name	ca-api-create.sh

Parâmetros do script

1	{ALERT.SUBJECT}
2	{ALERT.MESSAGE}
3	pcat:403911
4	28800
5	172.26.26.13
6	usrcaspc
7	caspectrum01
8	CA-SDM
9	W5nqgDWITH8
10	pri:501

CA Open Request - Prod Major

Configuração da media type.

Name	CA Open Request - Prod Major
Type	Script

Script Name	ca-api-create.sh
-------------	------------------

Parâmetros do script

1	{ALERT.SUBJECT}
2	{ALERT.MESSAGE}
3	pcat:403910
4	28800
5	172.26.26.13
6	usrcaspc
7	caspectrum01
8	CA-SDM
9	W5nqgDWITH8
10	pri:502

CA Open Request - Prod Critical

Configuração da media type.

Name	CA Open Request - Prod Critical
------	---------------------------------

Type	Script
Script Name	ca-api-create.sh

Parâmetros do script

1	{ALERT.SUBJECT}
2	{ALERT.MESSAGE}
3	pcat:403909
4	28800
5	172.26.26.13
6	usrcaspc
7	caspectrum01
8	CA-SDM
9	W5nqgDWTH8
10	pri:503

CA Open Request - THPP Minor

Configuração da media type.

Name	CA Open Request - THPP Minor
Type	Script
Script Name	ca-api-create.sh

Parâmetros do script

1	{ALERT.SUBJECT}
2	{ALERT.MESSAGE}
3	pcat:416591
4	28800
5	172.26.26.13
6	usrcaspc
7	caspectrum01
8	CA-SDM
9	W5nqgDWITH8
10	pri:501

CA Open Request - THPP Major

Configuração da media type.

Name	CA Open Request - THPP Major
Type	Script
Script Name	ca-api-create.sh

Parâmetros do script

1	{ALERT.SUBJECT}
2	{ALERT.MESSAGE}
3	pcat:416590
4	28800
5	172.26.26.13
6	usrcaspc
7	caspectrum01
8	CA-SDM
9	W5nqgDWTH8
10	pri:502

CA Open Request - THPP Critical

Configuração da media type.

Name	CA Open Request - THPP Critical
Type	Script
Script Name	ca-api-create.sh

Parâmetros do script

1	{ALERT.SUBJECT}
2	{ALERT.MESSAGE}
3	pcat:416589
4	28800
5	172.26.26.13
6	usrcaspc
7	caspectrum01
8	CA-SDM
9	W5nqgDWITH8
10	pri:503

Action

Existem 3 actions para abertura de tickets para cada ambiente, Produção e THPP

Ticket CA SDM THPP - Warning - Minor

Condição para o disparo da Action:

1	Manutenção	Host não esteja em manutenção
2	Status da trigger	PROBLEM
3	Trigger severity	Warning
4	Host Group Igual	Hosts Homologacao
5	Host Group Igual	Servidores_Unix_-_Homologacao
6	Host Group Igual	Servidores_NT_-_Homologacao
7	Host Group Igual	Servidores_Linux_-_Homologacao
8	Host Group Diferente	Projetos

Caso essas condições sejam satisfeitas, a action será executada após 3 minutos na media type "**CA Open Request - THPP Minor**"

Ticket CA SDM THPP - Average - Major

Condição para o disparo da Action:

1	Manutenção	Host não esteja em manutenção
2	Status da trigger	PROBLEM
3	Trigger severity	Average
4	Host Group Igual	Hosts Homologacao
5	Host Group Igual	Servidores_Unix_-_Homologacao
6	Host Group Igual	Servidores_NT_-_Homologacao
7	Host Group Igual	Servidores_Linux_-_Homologacao
8	Host Group Diferente	Projetos

Caso essas condições sejam satisfeitas, a action será executada após 3 minutos na media type **"CA Open Request - THPP Major"**

Ticket CA SDM THPP - HighDisaster - Critical

Condição para o disparo da Action:

1	Manutenção	Host não esteja em manutenção
2	Status da trigger	PROBLEM
3	Trigger severity	Maior igual a HIGH
4	Host Group Igual	Hosts Homologacao

5	Host Group Igual	Servidores_Unix_-_Homologacao
6	Host Group Igual	Servidores_NT_-_Homologacao
7	Host Group Igual	Servidores_Linux_-_Homologacao
8	Host Group Diferente	Projetos

Caso essas condições sejam satisfeitas, a action será executada após 3 minutos na media type **"CA Open Request - THPP Critical"**

Ticket CA SDM Producao - Warning - Minor

Condição para o disparo da Action:

1	Manutenção	Host não esteja em manutenção
2	Status da trigger	PROBLEM
3	Trigger severity	Warning
4	Host Group Igual	Hosts Producao
5	Host Group Igual	Servidores_Unix_-_Prod
6	Host Group Igual	Servidores_NT_-_Prod
7	Host Group Igual	Servidores_Linux_-_Prod
9	Host Group Diferente	Projetos

Caso essas condições sejam satisfeitas, a action será executada após 3 minutos na media type "**CA Open Request - Prod Minor**"

Ticket CA SDM Producao - Warning - Major

Condição para o disparo da Action:

1	Manutenção	Host não esteja em manutenção
2	Status da trigger	PROBLEM
3	Trigger severity	Average
4	Host Group Igual	Hosts Producao
5	Host Group Igual	Servidores_Unix_-_Prod
6	Host Group Igual	Servidores_NT_-_Prod
7	Host Group Igual	Servidores_Linux_-_Prod
9	Host Group Diferente	Projetos

Caso essas condições sejam satisfeitas, a action será executada após 3 minutos na media type "**CA Open Request - Prod Major**"

Ticket CA SDM Producao - HighDisaster - Critical

Condição para o disparo da Action:

1	Manutenção	Host não esteja em manutenção
---	------------	-------------------------------

2	Status da trigger	PROBLEM
3	Trigger severity	Maior igual a HIGH
4	Host Group Igual	Hosts Producao
5	Host Group Igual	Servidores_Unix_-_Prod
6	Host Group Igual	Servidores_NT_-_Prod
7	Host Group Igual	Servidores_Linux_-_Prod
9	Host Group Diferente	Projetos

Caso essas condições sejam satisfeitas, a action será executada após 3 minutos na media type "**CA Open Request - Prod Critical**"

Script de integração com SDM

Esse script deve estar em todos os Zabbix-servers no diretório **/etc/zabbix/alertscripts**.

```
#!/bin/bash
```

```
SDM_SUMMARY=$(echo $1 | sed 's/\r\n/g')
```

```
SDM_DESCRIPTION=$(echo $2 | sed 's/\r\n/g' | sed 's/&/g' | sed
```

```
's:<::g;s:>::g' )
SDM_LOG="/var/log/zabbix/sdm.log"
SDM_CATEGORY="$3"
SDM_SLA="$4"
SDM_ENDPOINT="$5"
SDM_USER="$6"
SDM_PASS="$7"
ENDPOINT="http://${SDM_ENDPOINT}:8080/axis/services/USD_R11_WebService"
SDM_TIMEOUT="20s"
SDM_PRIORITY=${10}

ZABBIX_USER="$8"
ZABBIX_PASS="$9"
ZABBIX_API='http://zabbix-web.portoseguro.brasil/api_jsonrpc.php'

test -e $SDM_LOG || touch $SDM_LOG
rpm -q jq &> /dev/null || echo "Dependencia: Instalar o comando \"jq\" >>
$SDM_LOG

# AUTENTICACAO
porto_sdm_auth(){
timeout ${SDM_TIMEOUT} curl --silent \
  --data \
  @- \
  --header 'Content-Type: application/soap+xml; charset=utf-8' \
  --header 'SOAPAction: ""' \
  --user-agent "" \
```

```

    ${ENDPOINT} <<EOF | xmllint --format -
    <?xml version="1.0" encoding="utf-8"?>
    <soapenv:Envelope
    xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:ser="http://www.ca.com/UnicenterServicePlus/ServiceDesk">
      <soapenv:Header/>
      <soapenv:Body>
        <ser:login>
          <username>${SDM_USER}</username>
          <password>${SDM_PASS}</password>
        </ser:login>
      </soapenv:Body>
    </soapenv:Envelope>
    EOF
  }

# LOGOUT
porto_sdm_logout(){
curl --silent \
  --data \
  @- \
  --header 'Content-Type: application/soap+xml; charset=utf-8' \
  --header 'SOAPAction: ""' \
  --user-agent "" \
  ${ENDPOINT} <<EOF | xmllint --format -
  <?xml version="1.0" encoding="utf-8"?>
  <soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ser="http://www.ca.com/UnicenterServicePlus/ServiceDesk">

```



```

<soapenv:Header/>
<soapenv:Body>
  <ser:logout>
    <sid>${SDM_LOGIN_RETURN}</sid>
  </ser:logout>
</soapenv:Body>
</soapenv:Envelope>
EOF
}

# CRIAR TICKET_REQUEST PARA A TRIGGERID
porto_sdm_create(){

timeout ${SDM_TIMEOUT} curl --silent \
  --data-binary \
  @- \
  --header 'Content-Type: application/soap+xml; charset=utf-8' \
  --header 'SOAPAction: ""' \
  --user-agent "" \
  ${ENDPOINT} <<EOF | xmllint --format -
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://www.ca.com/UnicenterServicePlus/ServiceDesk">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:createRequest>
      <sid>${SDM_LOGIN_RETURN}</sid>
      <creatorHandle></creatorHandle>

```

```
<attrVals>
  <string>customer</string>
  <string>cnt:B603608B61CD894CB13A9E38109B77CC</string>

  <string>description</string>
  <string>${SDM_DESCRIPTION}</string>

  <string>summary</string>
  <string>${SDM_SUMMARY}</string>

  <string>status</string>
  <string>OP</string>

  <string>priority</string>
  <string>${SDM_PRIORITY}</string>

  <string>type</string>
  <string>R</string>

  <string>z_str_telefone</string>
  <string>${ZABBIX_TRIGGERID}</string>

  <string>z_srl_loc</string>
  <string>AC927405A78FFA40A0AB0C2800D6B4AA</string>

  <string>z_str_email</string>
  <string>spectrum</string>

  <string>category</string>
```

```

    <string>${SDM_CATEGORY}</string>

    </attrVals>
    <propertyValues>
    <!-- The value Yes below is the value of the first (and only) property
dropdown -->
    <string>Yes</string>
    </propertyValues>
    <template></template>
    <attributes>
    <string>ref_num</string>
    </attributes>
    <newRequestHandle></newRequestHandle>
    <newRequestNumber></newRequestNumber>
    </ser:createRequest>
  </soapenv:Body>
</soapenv:Envelope>

EOF

}

# CHECK DO TICKET PARA A TRIGGERID
porto_sdm_check(){

timeout ${SDM_TIMEOUT} curl --silent \
  --data-binary \
  @- \

```

```

--header 'Content-Type: application/soap+xml; charset=utf-8' \
--header 'SOAPAction: "" \
--user-agent "" \
${ENDPOINT} <<EOF | xmllint --format -
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://www.ca.com/UnicenterServicePlus/ServiceDesk">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:doSelect>
      <sid>${SDM_LOGIN_RETURN}</sid>
      <objectType>cr</objectType>
      <whereClause>open_date > ${SDM_TIME_QUERY} AND status NOT
LIKE 'RE' AND status NOT LIKE 'CL' AND z_str_telefone LIKE
'${ZABBIX_TRIGGERID}'</whereClause>
      <maxRows>1</maxRows>
      <attributes>
      </attributes>
    </ser:doSelect>
  </soapenv:Body>
</soapenv:Envelope>

EOF

}

# ADICIONA UMA ATIVIDADE NO CA PARA O TICKET / TRIGGERID
porto_sdm_create_activityLog(){

```

```

timeout ${SDM_TIMEOUT} curl --silent \
  --data-binary \
  @- \
  --header 'Content-Type: application/soap+xml; charset=utf-8' \
  --header 'SOAPAction: ""' \
  --user-agent "" \
  ${ENDPOINT} <<EOF | xmllint --format -
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://www.ca.com/UnicenterServicePlus/ServiceDesk">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:createActivityLog>
      <sid>${SDM_LOGIN_RETURN}</sid>
      <creator>cnt:$PORTO_SDM_TICKET_check_log_agent</creator>

<objectHandle>${PORTO_SDM_TICKET_check_persistent_id}</objectHandle
>
    <description>
      ${SDM_DESCRIPTION}
    </description>
    <LogType>LOG</LogType>
    <TimeSpent>0</TimeSpent>
    <Internal>false</Internal>
  </ser:createActivityLog>
</soapenv:Body>
</soapenv:Envelope>
EOF

```

```

}

zabbix_ack_ticket(){

ZABBIX_EVENTID=$(echo "$SDM_DESCRIPTION" | grep "event ID" | awk -
F":" '{print $2}' | sed s:\ ::)
ZABBIX_TRIGGERID=$(echo "$SDM_DESCRIPTION" | grep "Trigger ID" |
awk -F":" '{print $2}' | sed s:\ ::)

ZABBIX_AUTH_TOKEN=$(curl -s -H 'Content-Type: application/json-rpc' -d
{"jsonrpc":
"2.0", "method": "user.login", "params": {"user": "${ZABBIX_USER}", "pa
ssword": "${ZABBIX_PASS}"}, "auth": null, "id": 0} $ZABBIX_API | jq -r
.result)

curl -s -H 'Content-Type: application/json-rpc' -d {"jsonrpc":
"2.0", "method": "event.acknowledge", "params": {"eventids": "${ZABBIX_
EVENTID}", "message": "${PORTO_SDM_TICKET}"}, "auth": "${ZABBIX_
AUTH_TOKEN}", "id": 0} $ZABBIX_API
}

# LOGIN
SDM_LOGIN_RETURN=$(porto_sdm_auth | grep loginReturn | cut -d> -f2 |
cut -d< -f1)

ZABBIX_EVENTID=$(echo "$SDM_DESCRIPTION" | grep "event ID" | awk -
F":" '{print $2}' | sed s:\ ::)

```

```

ZABBIX_TRIGGERID=$(echo "$SDM_DESCRIPTION" | grep "Trigger ID" |
awk -F":" '{print $2}' | sed s:\ ::)

SDM_TIME_QUERY=$(echo $(( $(date +%s) - ${SDM_SLA} )) )
# CHECK TICKET
PORTO_SDM_TICKET_check=$(porto_sdm_check)
PORTO_SDM_TICKET_check_ref_num=$( echo
$PORTO_SDM_TICKET_check | sed 's/</</g' | sed 's/>/>/g' | sed
's/<\AttrName> <AttrValue>//g' | tr -s '[:blank:]' ' ' | awk -F'ref_num' '{print $2}' |
awk -F '</AttrValue>' '{print $1}')
PORTO_SDM_TICKET_check_persistent_id=$( echo
$PORTO_SDM_TICKET_check | sed 's/</</g' | sed 's/>/>/g' | sed
's/<\AttrName> <AttrValue>//g' | tr -s '[:blank:]' ' ' | awk -F'persistent_id' '{print
$2}' | awk -F '</AttrValue>' '{print $1}' | tr -d '>' | tr -d '<' | tr -d '/')
PORTO_SDM_TICKET_check_log_agent=$( echo
$PORTO_SDM_TICKET_check | sed 's/</</g' | sed 's/>/>/g' | sed
's/<\AttrName> <AttrValue>//g' | tr -s '[:blank:]' ' ' | awk -F'log_agent' '{print
$2}' | awk -F '</AttrValue>' '{print $1}' | tr -d '>' | tr -d '<' | tr -d '/')

# OP:ABERTO CL:FECHADO | RE:RESOLVIDO

if [ ! -z $PORTO_SDM_TICKET_check_ref_num ];then
    # INSERT ACTIVITY_LOG
    porto_sdm_create_activityLog
    PORTO_SDM_TICKET=$PORTO_SDM_TICKET_check_ref_num
else

    # CREAT TICKET

```

```
    PORTO_SDM_TICKET=$(porto_sdm_create | grep newRequestNumber
| cut -d">" -f2 | cut -d"<" -f1)

fi

#ACK IN ZABBIX
if [ $SDM_LOGIN_RETURN -eq "1005" ] || [ -z $PORTO_SDM_TICKET ] ;
then
    PORTO_SDM_TICKET=$(echo Ticket ERRO na integracao com SDM)
    zabbix_ack_ticket
    zabbix_sender -z zabbix-server.portoseguro.brasil -s "zabbix-server" -k
integracao.sdm -o "http://zabbix-
web/tr_events.php?triggerid=${ZABBIX_TRIGGERID}&eventid=${ZABBIX_E
VENTID}"
else
    PORTO_SDM_TICKET=$(echo "Ticket $PORTO_SDM_TICKET")
    zabbix_ack_ticket
fi

echo
"$(date);"$ZABBIX_EVENTID;"$ZABBIX_TRIGGERID;"$SDM_ENDPOINT";
"$SDM_LOGIN_RETURN;"$PORTO_SDM_TICKET"" >> $SDM_LOG

porto_sdm_logout

#prereq: comando jq
```


Os campos "Original event ID" e "Trigger ID" da mensagem da Action são utilizados pelo script de integração. Qualquer alteração nestes campos podem indisponibilizar a integração.

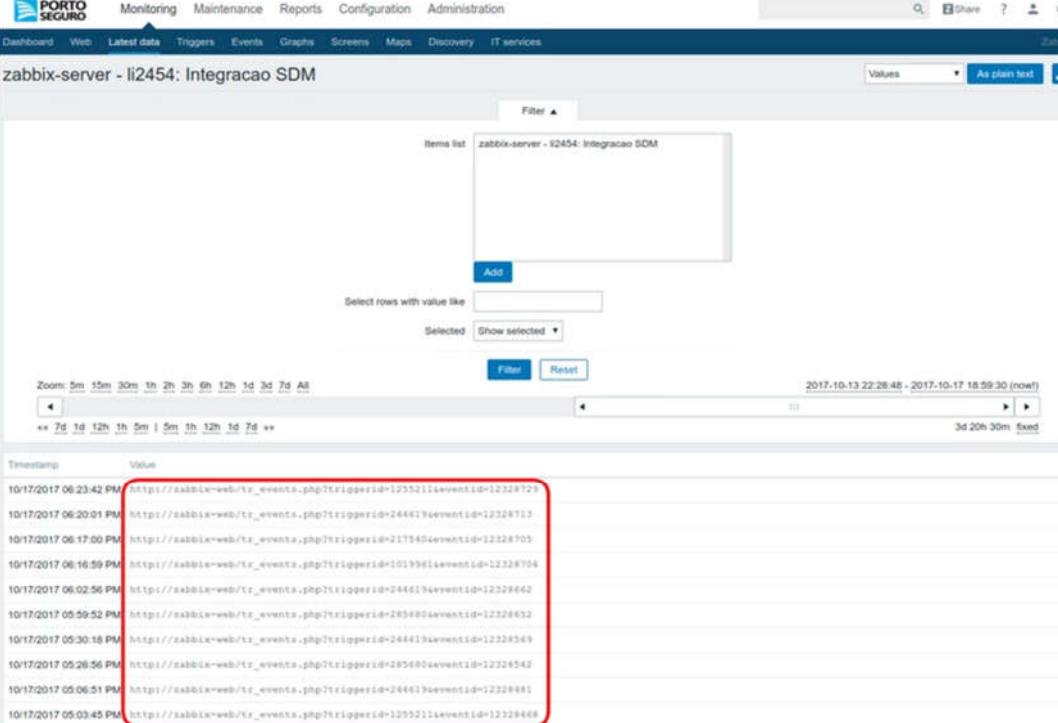
The screenshot shows the Zabbix web interface for configuring an Action. The top navigation bar includes 'Monitoring', 'Maintenance', 'Reports', 'Configuration', and 'Administration'. Below this, a secondary bar shows 'Host groups', 'Templates', 'Hosts', 'Clear_Traps', 'Actions', 'Discovery', and 'IT services'. The 'Actions' tab is selected, and the 'Action' sub-tab is active. The configuration form includes fields for 'Name' (Ticket CA SDM THPP - Warning - Minor), 'Default subject' ({HOST.NAME1} : {TRIGGER.NAME}), and 'Default message'. The 'Default message' field contains a template with the following placeholders: 'Original event ID: {EVENT.ID}', 'Trigger ID: {TRIGGER.ID}', 'Trigger Name: {TRIGGER.NAME}', 'Trigger severity: {TRIGGER.SEVERITY}', 'Trigger URL: {TRIGGER.URL}', 'Item values: 1. {ITEM.NAME1} ({HOST.NAME1}:{ITEM.KEY1}): {ITEM.VALUE1}', and 'Trigger Description: {TRIGGER.DESCRPTION}'. The 'Original event ID' and 'Trigger ID' placeholders are circled in red. At the bottom, there are checkboxes for 'Recovery message' (unchecked) and 'Enabled' (checked), and buttons for 'Update', 'Clone', 'Delete', and 'Cancel'.

O usuário CA-SDM é utilizado para inserir o comentário (ack) no evento com o nome do ticket, portanto, este usuário não deve ser deletado ou alterado. Caso tenha uma alteração, é necessário ajustar os parâmetros do media type com as novas credenciais.

O campo telefone (z_str_telefone) do ticket é populado com a triggerID do evento e é utilizado na regra de tempo do ticket, caso este campo seja excluído/alterado é necessário realizar os ajustes no script de integração.

Erro na integração

Em caso de erro na abertura de um ticket no PortoSDM uma mensagem é enviada para um item no zabbix-server, que informará a URL de todos os eventos que geraram erro na integração.



Timestamp	Value
10/17/2017 06:23:42 PM	http://zabbix-web/tr_events.php?triggerid=1255211&eventid=12328729
10/17/2017 06:20:01 PM	http://zabbix-web/tr_events.php?triggerid=244619&eventid=12328713
10/17/2017 06:17:00 PM	http://zabbix-web/tr_events.php?triggerid=217540&eventid=12328705
10/17/2017 06:16:59 PM	http://zabbix-web/tr_events.php?triggerid=1019501&eventid=12328704
10/17/2017 06:02:56 PM	http://zabbix-web/tr_events.php?triggerid=244619&eventid=12328662
10/17/2017 05:59:52 PM	http://zabbix-web/tr_events.php?triggerid=283680&eventid=12328632
10/17/2017 05:30:18 PM	http://zabbix-web/tr_events.php?triggerid=244619&eventid=12328569
10/17/2017 05:26:56 PM	http://zabbix-web/tr_events.php?triggerid=283680&eventid=12328542
10/17/2017 05:06:51 PM	http://zabbix-web/tr_events.php?triggerid=244619&eventid=12328481
10/17/2017 05:03:45 PM	http://zabbix-web/tr_events.php?triggerid=1255211&eventid=12328469

Para este item existe uma action (Alerta erro de integracao SDM) que envia um E-mail para os endereços centro.decomando@portoseguro.com.br e gestaoconfiguracao.ferramentas@portoseguro.com.br com o link direto do evento que gerou problema.

https://www.zabbix.com/documentation/3.0/manual/web_interface/frontend_sections/configuration/actions

Módulo VMBIX

A monitoração do ambiente VMware está sendo feito com o plugin nativo do zabbix e um módulo externo chamado VMBIX. Esse módulo foi implementado no ambiente por conta da monitoração dos datastores dos hosts VMware. Por padrão o zabbix mapeia todos os datastores associados aos hosts VMware, desta forma, caso tenha um datastore apresentado para mais de um host VMware e o mesmo ultrapasse o threshold o alerta será gerado em todos os hosts que aquele datastore estiver associado, gerando múltiplos eventos do mesmo datastore.

Por padrão o módulo VMBIX não suporta a monitoração de múltiplos vCenters por um mesmo proxy, portanto, é necessário seguir o procedimento abaixo para que isso seja possível:

1- fazer o download do código fonte do zabbix no endereço

<https://sourceforge.net/projects/zabbix/files/ZABBIX%20Latest%20Stable/>,

descompactar o tarball e executar o comando abaixo:

```
POC zabbix-server+WEB (LI2128)[zabbix-3.0.8]# ./configure --with-openssl
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
configure: Configuring Zabbix 3.0.8
checking whether make sets $(MAKE)... (cached) yes
checking build system type... x86_64-unknown-linux-gnu
```

```
checking host system type... x86_64-unknown-linux-gnu
...
...
```

2- fazer o download do código fonte do VMBIX no endereço https://github.com/dav3860/vmbix_zabbix_module/archive/master.zip e descompactar dentro do diretório <source_zabbix>/src/modules/

3- entrar no diretório do módulo e editar o arquivo vmbix-3.0.c substituindo a linha abaixo:

De:

```
#define CONFIG_FILE "/etc/zabbix/vmbix_module.conf"
```

Para:

```
#define CONFIG_FILE "/etc/zabbix/vmbix_module_NOMEDOVMMWARE.conf"
```

***Alterar o NOMEDOVMMWARE para o nome do vCenter. Ex: clesx001, clesx002 ou nt1959**

será necessário ajustar o nome dos itens no mesmo arquivo:

De:

```
/* KEY      FLAG      FUNCTION      TEST PARAMETERS */
{
    {"vmbix",    CF_HAVEPARAMS, zbx_module_vmbix, NULL},
    {"vmbix.ping", CF_HAVEPARAMS, zbx_module_vmbix_ping, NULL},
    {NULL}
};
```

Para:

```
/* KEY      FLAG      FUNCTION      TEST PARAMETERS */
{
    {"vmbix_NOMEDOVMMWARE",    CF_HAVEPARAMS,
    zbx_module_vmbix, NULL},
    {"vmbix.ping_NOMEDOVMMWARE", CF_HAVEPARAMS,
    zbx_module_vmbix_ping, NULL},
    {NULL}
};
```

***Alterar o NOMEDOVMMWARE para o nome do vCenter. Ex: clesx001, clesx002 ou nt1959**

4- após o ajuste é necessário compilar o módulo com o comando abaixo:

```
POC zabbix-server+WEB (LI2128)[vmbix_zabbix_module-master]# make
vmbix-3.0
gcc -shared -o vmbix.so vmbix-3.0.c -I../..../include -fPIC
```

5- renomear o nome do módulo para o mesmo nome do vCenter.

```
POC zabbix-server+WEB (LI2128)[vmbix_zabbix_module-master]# mv
vmbix.so vmbix_NOMEDOVMMWARE.so
```

***Alterar o NOMEDOVMMWARE para o nome do vCenter. Ex: clesx001, clesx002 ou nt1959**

6- Repita os passos 3, 4 e 5 para todos os vCenter.

O procedimento de instalação e configuração deste módulo estão listados no procedimento de instalação dos servidores do proxy VMware.

Validação

Para verificar se os módulos estão corretamente configurados, é necessário reiniciar o zabbix-proxy e verificar no log se o módulo foi carregado:

```
26426:20170424:204040.504 Starting Zabbix Server. Zabbix 3.0.8 (revision
65976).
26426:20170424:204040.504 ***** Enabled features *****
26426:20170424:204040.504 SNMP monitoring:      YES
26426:20170424:204040.504 IPMI monitoring:      YES
26426:20170424:204040.504 Web monitoring:      YES
26426:20170424:204040.504 VMware monitoring:      YES
26426:20170424:204040.504 SMTP authentication:    NO
26426:20170424:204040.504 Jabber notifications:    YES
26426:20170424:204040.504 Ez Texting notifications: YES
26426:20170424:204040.504 ODBC:                  YES
26426:20170424:204040.504 SSH2 support:          YES
26426:20170424:204040.504 IPv6 support:          YES
26426:20170424:204040.504 TLS support:           YES
26426:20170424:204040.505 *****
26426:20170424:204040.505 using configuration file:
/etc/zabbix/zabbix_server.conf
26426:20170424:204040.505 Loading VmBix module configuration file
/etc/zabbix/vmbix_module.conf
26426:20170424:204040.505 loaded modules: vmbix.so
26426:20170424:204040.526 current database version (mandatory/optional):
03000000/03000000
26426:20170424:204040.526 required mandatory version: 03000000
```

A linha "loaded modules: " deve aparecer no log para todos os módulos configurados.

Validando o carregamento do módulo no arquivo de log do zabbix-proxy, podemos testar a coleta das informações com o comando abaixo:

```
proxy-vmware (LI2497)[~]# zabbix_get -s 127.0.0.1 -p 12051 -k
"cluster.discovery"
{"data":[{"#CLUSTER":"THPP_Linux2"},{"#CLUSTER":"CTI-DEV"},{"#CLUSTER":"DMZ-DC"},{"#CLUSTER":"THPp_Windows"},{"#CLUSTER":"Corporativo_Linux-DC"},{"#CLUSTER":"PA_RISC-DC"},{"#CLUSTER":"THPp_Linux"},{"#CLUSTER":"Gravadores-BF"},{"#CLUSTER":"COL-DC"},{"#CLUSTER":"Corporativo_Windows-DC"},{"#CLUSTER":"BI-TABLEAU-DC"},{"#CLUSTER":"Telecom-DC"},{"#CLUSTER":"Corporativo_Linux2-DC"}]}
```

Esse comando lista todos os clusters do vCenter.

*A porta de cada vCenter é definida dentro do arquivo
/etc/vmbix/vmbix_NOMEDOVMWARE.conf

Itens disponíveis

Segue uma lista com todos os itens disponibilizados pelo módulo VMBIX:

```
Available methods :
vmbix.ping
vmbix.version
vmbix.stats[threads]
vmbix.stats[queue]
vmbix.stats[requests]
vmbix.stats[cachesize,(vm|esxi|ds|perf|counter|hri|cluster)]
vmbix.stats[hirate,(vm|esxi|ds|perf|counter|hri|cluster)]
about
```



```
cluster.discovery
cluster.cpu[name,free]
cluster.cpu[name,total]
cluster.cpu[name,usage]
cluster.cpu.num[name,threads]
cluster.cpu.num[name,cores]
cluster.mem[name,free]
cluster.mem[name,total]
cluster.mem[name,usage]
cluster.hosts[name,online]
cluster.hosts[name,maint]
cluster.hosts[name,total]
datacenter.discovery
datacenter.status[name,(overall|config)]
datastore.discovery
datastore.local[(uuid|name)]
datastore.size[(uuid|name),free]
datastore.size[(uuid|name),total]
datastore.size[(uuid|name),provisioned]
datastore.size[(uuid|name),uncommitted]
esx.connection[(uuid|name)]
esx.uptime[(uuid|name)]
esx.cpu.load[(uuid|name),cores]
esx.cpu.load[(uuid|name),total]
esx.cpu.load[(uuid|name),used]
esx.discovery
esx.maintenance[(uuid|name)]
esx.memory[(uuid|name),total]
esx.memory[(uuid|name),used]
```

```
esx.path[(uuid|name),active]
esx.path[(uuid|name),dead]
esx.path[(uuid|name),disabled]
esx.path[(uuid|name),standby]
esx.status[(uuid|name)]
esx.vms.count[(uuid|name)]
esx.vms.memory[(uuid|name),active]
esx.vms.memory[(uuid|name),ballooned]
esx.vms.memory[(uuid|name),compressed]
esx.vms.memory[(uuid|name),consumed]
esx.vms.memory[(uuid|name),overheadConsumed]
esx.vms.memory[(uuid|name),private]
esx.vms.memory[(uuid|name),shared]
esx.vms.memory[(uuid|name),swapped]
esx.counter[(uuid|name),counter,[instance,interval]]
esx.counter.discovery[(uuid|name),counter,[interval]]
esx.counter.list[(uuid|name)]
event.latest[*]
vm.consolidation[(uuid|name),needed]
vm.cpu.load[(uuid|name),cores]
vm.cpu.load[(uuid|name),total]
vm.cpu.load[(uuid|name),used]
vm.discovery[*]
vm.discovery.full[*]
vm.folder[(uuid|name)]
vm.uptime[(uuid|name)]
vm.name[(uuid|name)]
vm.annotation[(uuid|name)]
vm.guest.disk.discovery[(uuid|name)]
```

```
vm.guest.disk.capacity[(uuid|name),disk]
vm.guest.disk.free[(uuid|name),disk]
vm.guest.ip[(uuid|name)]
vm.guest.(uuid|name)[(uuid|name)]
vm.guest.os[(uuid|name)]
vm.guest.os.short[(uuid|name)]
vm.guest.tools.mounted[(uuid|name)]
vm.guest.tools.running[(uuid|name)]
vm.guest.tools.version[(uuid|name)]
vm.host[(uuid|name)]
vm.memory[(uuid|name),active]
vm.memory[(uuid|name),ballooned]
vm.memory[(uuid|name),compressed]
vm.memory[(uuid|name),consumed]
vm.memory[(uuid|name),overheadConsumed]
vm.memory[(uuid|name),private]
vm.memory[(uuid|name),shared]
vm.memory[(uuid|name),swapped]
vm.memory[(uuid|name),total]
vm.counter[(uuid|name),counter,[instance,interval]]
vm.counter.discovery[(uuid|name),counter,[interval]]
vm.counter.list[(uuid|name)]
vm.powerstate[(uuid|name)]
vm.status[(uuid|name)]
vm.storage.committed[(uuid|name)]
vm.storage.uncommitted[(uuid|name)]
vm.storage.unshared[(uuid|name)]
vm.snapshot[(uuid|name)]
```

Para maiores informações de cada item, acesso o site da documentação:

<https://github.com/dav3860/vmbix/wiki>

Quando configuramos esses itens no zabbix, é necessário adicionar a key configurada no arquivo vmbix-3.0.c.

Exemplo:

A coleta via linha de comando é feita com a key "event.latest, *"

```
proxy-vmware (LI2497)[~]# zabbix_get -s 127.0.0.1 -p 12051 -k  
"event.latest, *"  
Alarm 'Virtual machine CPU usage' on LI1154 changed from Red to Green
```

já no frontend do zabbix a key é configurada da seguinte forma:

vmbix_clesx001[event.latest, *]

PORTO SEGURO Monitoring Maintenance Reports Configuration Administration

Host groups Templates Hosts Clear_Traps Actions Discovery IT services

Items

All templates / Template VmBix vCenter Loadable Module clesx001 Applications 5 Items 3 Triggers 1 Graphs Screens Discovery rules 3 Web scenarios

Name Latest event ⓘ

Type Simple check ▼

Key vmbox_clesx001[event.latest,*] Select

User name

Password ⓘ

Type of information Character ▼

Update interval (in sec) 60

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

[Add](#)

History storage period (in days) 7

Show value As is ▼ [show value mappings](#)

New application

Applications

- None-
- CPU
- Datastore
- ESX
- Memory
- Performance
- Storage
- vCenter
- Virtual Machine

Populates host inventory field -None- ▼

Description

Enabled ☒

[Update](#) [Clone](#) [Delete](#) [Cancel](#)

Converter as OIDs de TRAP

Copiar o arquivo de MIB para um proxy e executar o comando abaixo:

```
proxy-snmpp (LI2461)[tmp]# snmpttconvertmib --in=ECS-MIB-v2.mib --
out=ECS-MIB-v2.txt --format=4

***** Processing MIB file *****

snmptranslate version: NET-SNMP version: 5.5
severity: Normal

File to load is:      ./ECS-MIB-v2.mib
File to APPEND TO:    ./ECS-MIB-v2.txt

MIBS environment var: ./ECS-MIB-v2.mib
mib name: ECS-MIB

Processing MIB:      ECS-MIB
#
skipping a TRAP-TYPE / NOTIFICATION-TYPE line - probably an import line.
#
Line: 65
NOTIFICATION-TYPE: trapAlarmNotification
Variables: notifyTimestamp notifySeverity notifyType notifyDescription
Enterprise: notificationTrap
Looking up via snmptranslate: ECS-MIB::trapAlarmNotification
add_mibdir: strings scanned in from /root/.snmp/mibs/.index are too large.
count = 1209
```

```
OID: .1.3.6.1.4.1.1139.102.1.1
```

```
Done
```

```
Total translations:      1
```

```
Successful translations:  1
```

```
Failed translations:      0
```

Neste exemplo apenas uma OID de TRAP foi encontrada na MIB ECS-MIB-v2.mib com a OID .1.3.6.1.4.1.1139.102.1.1. As informações convertidas foram escritas no arquivo ECS-MIB-v2.txt.

```
proxy-snmp (LI2461)[tmp]# cat ./ECS-MIB-v2.txt
```

```
#
```

```
#
```

```
#
```

```
#
```

```
MIB: ECS-MIB (file:./ECS-MIB-v2.mib) converted on Thu Oct 19 16:59:27  
2017 using snmpttconvertmib v1.4
```

```
#
```

```
#
```

```
#
```

```
EVENT trapAlarmNotification .1.3.6.1.4.1.1139.102.1.1 "Status Events"
```

```
Normal
```

```
FORMAT trapAlarmNotification - notifyTimestamp:$1 notifySeverity:$2  
notifyType:$3 notifyDescription:$4
```

```
SDESC
```

```
This trap identifies a problem on the ECS. The description can be used to
```

describe the nature of the change

Variables:

1: notifyTimestamp

2: notifySeverity

3: notifyType

4: notifyDescription

EDESC

Após esta conversão é necessário copiar as linhas que começam com EVENT e FORMAT para o arquivo `/etc/snmp/snmpd.conf` com o formato do zabbix:

o 'SNMPD trap configuration' precisam formatar a trap conforme o padrão a seguir: **[timestamp] [the trap, part 1] ZBXTRAP [address] [the trap, part 2]**, onde

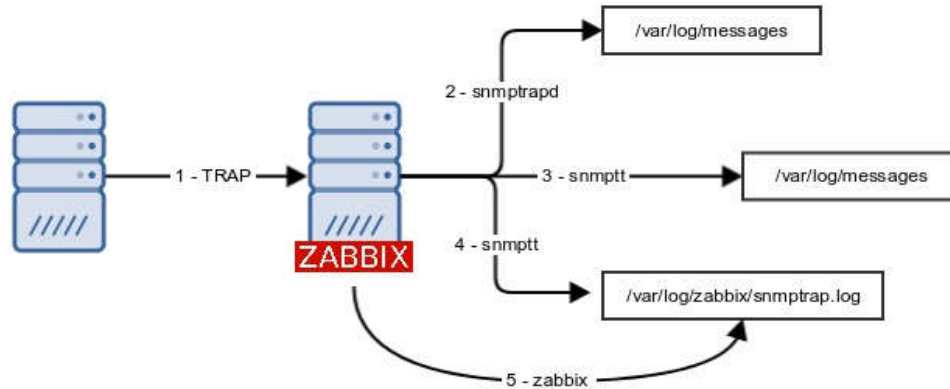
- [timestamp] - momento de ocorrência do evento
- ZBXTRAP - cabeçalho que indica o início de uma nova linha de trap
- [address] - endereço IP ou DNS para localizar o host a receber a trap

O campo address pode ser usado a variável \$aA que representa o IP de origem da TRAP ou o \$A que representa o DNS do host de origem.

No link <http://snmpd.sourceforge.net/docs/snmpd.shtml#SNMPD.CONF-FORMAT> é possível encontrar todas as variáveis que podem ser usadas no campo FORMAT.

Após o ajuste do arquivo `/etc/snmp/snmpd.conf` é necessário reiniciar o serviço snmpd com o comando **`"/etc/init.d/snmpd restart"`**.

Funcionamento SNMPD



1- trap enviada para o zabbix-proxy na porta 162 (snmptrapd)

2- serviço snmptrapd recebe a trap e escreve no log /var/log/messages no formato "puro"

Ex:

```

Oct 16 07:58:10 li2461 snmptrapd[3473]: 2017-10-16 07:58:10
sw326.portoseguro.brasil [172.26.74.116] (via UDP: [172.26.74.116]:32772-
>[172.27.47.95]) TRAP, SNMP v1, community public#012#011SNMPv2-
SMI::enterprises.1588.2.1.1.1 Enterprise Specific Trap (3) Uptime: 352 days,
8:13:31.00#012#011SNMPv2-SMI::enterprises.1588.2.1.1.1.6.2.1.4.88 =
INTEGER: 2#011SNMPv2-SMI::enterprises.1588.2.1.1.1.6.2.1.1.88 =
INTEGER: 88#011SNMPv2-SMI::enterprises.1588.2.1.1.1.6.2.1.36.88 =
STRING: "VHXEN043"#011SNMPv2-
SMI::enterprises.1588.2.1.1.1.6.2.1.34.88 = Hex-STRING: 20 57 00 05 33 63
18 00 #011SNMPv2-SMI::enterprises.1588.2.1.1.1.1.35.88 = INTEGER:
4#011SNMPv2-SMI::enterprises.1588.2.1.1.1.6.2.1.39.88 = INTEGER:
6#011SNMPv2-SMI::enterprises.1588.2.1.1.1.1.10.0 = STRING:
"BRCAFX1935G028"#011SNMPv2-SMI::enterprises.1588.2.1.1.1.6.2.1.38.88
= Hex-STRING: 00 #011SNMPv2-SMI::enterprises.1588.2.1.1.1.6.2.1.40.88 =
  
```

INTEGER: 2

3 e 4- serviço snmptrapd encaminha a trap para o snmptt, que trata a mensagem e escreve nos arquivos de log /var/log/messages e /var/log/zabbix/snmptrap.log

Log no arquivo /var/log/messages

```
Oct 16 07:58:11 li2461 snmptt[12976]: .1.3.6.1.4.1.1588.2.1.1.1.0.3 Normal
"Brocade_swFCPortScn" sw326.portoseguro.brasil - ZBXTRAP
172.26.74.116 swFCPortScn swFCPortOpStatus 2 swFCPortIndex 88
swFCPortName VHXEN043 swSsn 20 57 00 05 33 63 18 00 swFCPortFlag 4
swGroupName 6 swGroupType BRCAFX1935G028 swGroupMemPos 00
swVfld 2
```

Log no arquivo /var/log/zabbix/snmptrap.log

```
07:58:10 2017/10/16 .1.3.6.1.4.1.1588.2.1.1.1.0.3 Normal
"Brocade_swFCPortScn" sw326.portoseguro.brasil - ZBXTRAP
172.26.74.116 swFCPortScn swFCPortOpStatus 2 swFCPortIndex 88
swFCPortName VHXEN043 swSsn 20 57 00 05 33 63 18 00 swFCPortFlag 4
swGroupName 6 swGroupType BRCAFX1935G028 swGroupMemPos 00
swVfld 2
```

5- zabbix verifica as traps escritas no arquivo /var/log/zabbix/snmptrap.log e associa o log escrito para um item, se existir, caso não exista um mensagem é escrita no arquivo /var/log/zabbix/zabbix_proxy.log (unmatched trap received).

<https://www.zabbix.com/documentation/3.0/pt/manual/config/items/itemtypes/snmptrap>

<http://snmptt.sourceforge.net/docs/snmptt.shtml#SNMPTT.CONF-Configuration-file-format>

Itens customizados - External Scripts

Caseirao

O item caseirao testa se um ponto de montagem está acessível.

Parâmetros do caseirao

Key: caseirao[*]

Este item precisa dos 5 parâmetros abaixo para funcionar.

N	Parametro
1	Timeout em segundos
2	Host
3	Ponto de montagem
4	Usuario
5	Senha

Exemplo: caseirao[15,apl-corp01,/fincobpdf_book/ps,usrcaspc,caspectrum01]

Name

Type

Key

Type of information

Data type

Units

Use custom multiplier ☐

Update interval (in sec)

History storage period (in days)

Trend storage period (in days)

Store value

Show value [show value mappings](#)

New application

Applications

- None-
- Caseirao
- caseirao_ckfile
- caseirao_count
- caseirao_filedat
- Cohesion
- CPU
- Disk Operations/s
- Disk Sector/s
- Filesystems

Populates host inventory field

Description

Enabled ☒

- Este item utiliza o tipo de dado Numeric (unsigned)

Script caseirao.sh

Salvar o script no caminho: /etc/zabbix/scripts/caseirao.sh

```
#!/bin/bash
TIMEOUT=$1
SERVIDOR=$2
COMPARTILHAMENTO=$3
USUARIO=$4
SENHA=$5
P_MONTAGEM="/tmp${COMPARTILHAMENTO}"
DEBUG=1 # 0 ou 1

if [ $DEBUG -eq 0 ]; then
    LOG=/dev/null
else
    LOG=/tmp/caseirao.log
fi

test -d "${P_MONTAGEM}" || mkdir -p "${P_MONTAGEM}"

echo "timeout ${TIMEOUT}s mount -t cifs -o
username=${USUARIO},password=${SENHA}
//${SERVIDOR}${COMPARTILHAMENTO} ${P_MONTAGEM}" 2>&1>>
${LOG}

timeout ${TIMEOUT}s mount -t cifs -o
username=${USUARIO},password=${SENHA}
"${SERVIDOR}${COMPARTILHAMENTO}" "${P_MONTAGEM}" 2>&1>>
${LOG}

RESULTADO=$?
```

```

echo $RESULTADO

umount "${P_MONTAGEM}" 2>> ${LOG}

echo
"${date} ${TIMEOUT}; ${SERVIDOR}; ${COMPARTILHAMENTO}; ${USUARIO}
${SENHA}; ${RESULTADO}" 2>&1>> ${LOG}
echo "-----" 2>&1>> ${LOG}

```

Customização no agente

Para que este item seja reconhecido pelo zabbix_agentd, a linha abaixo deve ser inserida no arquivo de configuração /etc/zabbix/zabbix_agentd.conf.

```
UserParameter=caseirao[*],/etc/zabbix/scripts/caseirao.sh $1 $2 $3 $4 $5
```

- Após essa alteração será necessário reiniciar o agente do zabbix
(*/etc/init.d/zabbix-agent restart*)

Caseirao_count

O item caseirao_count testa se um ponto de montagem está acessível e conta quantos arquivos existem no diretório.

Parâmetros do caseirao_count

Key: caseirao_count[*]

Este item precisa dos 6 parâmetros abaixo para funcionar.

N	Parametro
---	-----------

1	Timeout em segundos
2	Host
3	Ponto de montagem
4	Usuario
5	Senha
6	*Teste Recursivo

- o parâmetro 6 aceita apenas o valor "R" ou vazio. O Valor R deve ser configurado quando a contagem de dados considerar os subdiretórios (recursivo).

Exemplo:

```
caseirao_count[15,pfs08,/PortoSeg/WebSysin/ROBOSYS/LOGIP,usrcaspc,caspectrum01,R]
```


Name

Type

Key

Type of information

Units

Use custom multiplier ☐

Update interval (in sec)

History storage period (in days)

Trend storage period (in days)

Store value

Show value [show value mappings](#)

New application

Applications

- None-
- Caseirao
- caseirao_ckfile
- caseirao_count
- caseirao_filedat
- Cohesion
- CPU
- Disk Operations/s
- Disk Sector/s
- Filesystems

Populates host inventory field

Description

Enabled ☒

- Este item utiliza o tipo de dado Numeric (float)

Script caseirao_count.sh

Salvar o script no caminho: /etc/zabbix/scripts/caseirao_count.sh

```
#!/bin/bash
TIMEOUT="$1"
SERVIDOR="$2"
COMPARTILHAMENTO="$3"
USUARIO="$4"
SENHA="$5"
RECURSIVO="$6"
P_MONTAGEM="/tmp${COMPARTILHAMENTO}"
DEBUG=1 # 0 ou 1

if [ $DEBUG -eq 0 ]; then
    LOG=/dev/null
else
    LOG=/tmp/caseirao_count.log
fi

test -d "${P_MONTAGEM}" || mkdir -p "${P_MONTAGEM}"

echo "timeout ${TIMEOUT}s mount -t cifs -o
username=${USUARIO},password=${SENHA}
//${SERVIDOR}${COMPARTILHAMENTO} ${P_MONTAGEM}" 2>&1>>
${LOG}

timeout ${TIMEOUT}s mount -t cifs -o
username=${USUARIO},password=${SENHA}
"//${SERVIDOR}${COMPARTILHAMENTO}" "${P_MONTAGEM}" 2>&1>>
${LOG}

RESULTADO=$?
```

```

if [ ${RESULTADO} -eq 0 ] ; then
    ls -lh${RECURSIVO} "${P_MONTAGEM}" | grep "^-" | wc -l
else
    echo -1
fi

umount "${P_MONTAGEM}" 2>> ${LOG}

echo
"$(date);${TIMEOUT};${SERVIDOR};${COMPARTILHAMENTO};${USUARIO}
;${SENHA};${RESULTADO}" 2>&1>> ${LOG}
echo "-----" 2>&1>> ${LOG}

```

Customização no agente

Para que este item seja reconhecido pelo zabbix_agentd, a linha abaixo deve ser inserida no arquivo de configuração /etc/zabbix/zabbix_agentd.conf.

```

UserParameter=caseirao_count[*],/etc/zabbix/scripts/caseirao_count.sh $1 $2
$3 $4 $5 $6

```

- Após essa alteração será necessário reiniciar o agente do zabbix
(*/etc/init.d/zabbix-agent restart*)

Caseirao_count_dir

O item caseirao_count_dir testa se um ponto de montagem está acessível e conta quantos diretórios existem no share.

Parâmetros do caseirao_count_dir

Key: caseirao_count_dir[*]

Este item precisa dos 6 parâmetros abaixo para funcionar.

N	Parametro
1	Timeout em segundos
2	Host
3	Ponto de montagem
4	Usuario
5	Senha
6	*Teste Recursivo

- o parâmetro 6 aceita apenas o valor "R" ou vazio. O Valor R deve ser configurado quando a contagem de dados considerar os subdiretórios (recursivo).

Exemplo:

```
caseirao_count_dir[15,aplbenner,/Producao/Arquivos/Solicitacao/2,usrcaspc,ca  
spectrum01]
```

Name

Type

Key

Type of information

Units

Use custom multiplier ☐

Update interval (in sec)

History storage period (in days)

Trend storage period (in days)

Store value

Show value [show value mappings](#)

New application

- Este item utiliza o tipo de dado Numeric (float)

Script caseirao_count_dir.sh

Salvar o script no caminho: /etc/zabbix/scripts/caseirao_count_dir.sh

```
#!/bin/bash
TIMEOUT="$1"
SERVIDOR="$2"
COMPARTILHAMENTO="$3"
USUARIO="$4"
SENHA="$5"
RECURSIVO="$6"
P_MONTAGEM="/tmp${COMPARTILHAMENTO}"
DEBUG=1 # 0 ou 1

if [ $DEBUG -eq 0 ]; then
```

```
LOG=/dev/null
else
    LOG=/tmp/caseirao_count.log
fi

test -d "${P_MONTAGEM}" || mkdir -p "${P_MONTAGEM}"

echo "timeout ${TIMEOUT}s mount -t cifs -o
username=${USUARIO},password=${SENHA}
//${SERVIDOR}${COMPARTILHAMENTO} ${P_MONTAGEM}" 2>&1>>
${LOG}
timeout ${TIMEOUT}s mount -t cifs -o
username=${USUARIO},password=${SENHA}
"//${SERVIDOR}${COMPARTILHAMENTO}" "${P_MONTAGEM}" 2>&1>>
${LOG}

RESULTADO=$?

if [ ${RESULTADO} -eq 0 ] ; then
    ls -lh${RECURSIVO} "${P_MONTAGEM}" | grep "^d" | wc -l
else
    echo -1
fi

umount "${P_MONTAGEM}" 2>> ${LOG}

echo
"${date};"${TIMEOUT};"${SERVIDOR};"${COMPARTILHAMENTO};"${USUARIO}
;"${SENHA};"${RESULTADO}" 2>&1>> ${LOG}
```

```
echo "-----" 2>&1>> ${LOG}
```

Customização no agente

Para que este item seja reconhecido pelo zabbix_agentd, a linha abaixo deve ser inserida no arquivo de configuração /etc/zabbix/zabbix_agentd.conf.

```
UserParameter=caseirao_count_dir[*],/etc/zabbix/scripts/caseirao_count_dir.s  
h $1 $2 $3 $4 $5 $6
```

- Após essa alteração será necessário reiniciar o agente do zabbix
(*/etc/init.d/zabbix-agent restart*)

Caseirao_ckfile

O item caseirao_ckfile testa se um ponto de montagem está acessível e verifica o cksum de um arquivo.

Parâmetros do caseirao_ckfile

Key: caseirao_ckfile[*]

Este item precisa dos 6 parâmetros abaixo para funcionar.

N	Parametro
1	Timeout em segundos
2	Host
3	Ponto de montagem

4	Usuario
5	Senha
6	Nome do arquivo

Exemplo:

```
caseirao_ckfile[15,img01,/vdi/iso_bdm,usrcaspc,caspectrum01,ISO_784_DB.iso]
```


Name

Type

Key

Type of information

Units

Use custom multiplier ☐

Update interval (in sec)

History storage period (in days)

Trend storage period (in days)

Store value

Show value [show value mappings](#)

New application

Applications

- None-
- Caseirao
- caseirao_ckfile
- caseirao_count
- caseirao_filedat
- Cohesion
- CPU
- Disk Operations/s
- Disk Sector/s
- Filesystems

Populates host inventory field

Description

Enabled ☒

- Este item utiliza o tipo de dado Numeric (float)

Script caseirao_ckfile.sh

Salvar o script no caminho: /etc/zabbix/scripts/caseirao_ckfile.sh

```
#!/bin/bash
TIMEOUT="$1"
SERVIDOR="$2"
COMPARTILHAMENTO="$3"
USUARIO="$4"
SENHA="$5"
ARQUIVO="$6"
P_MONTAGEM="/tmp${COMPARTILHAMENTO}"
DEBUG=1 # 0 ou 1

if [ $DEBUG -eq 0 ]; then
    LOG=/dev/null
else
    LOG=/tmp/caseirao_ckfile.log
fi

test -d "${P_MONTAGEM}" || mkdir -p "${P_MONTAGEM}"

echo "timeout ${TIMEOUT}s mount -t cifs -o
username=${USUARIO},password=${SENHA}
//${SERVIDOR}${COMPARTILHAMENTO} ${P_MONTAGEM}" 2>&1>>
${LOG}
timeout ${TIMEOUT}s mount -t cifs -o
username=${USUARIO},password=${SENHA}
"//${SERVIDOR}${COMPARTILHAMENTO}" "${P_MONTAGEM}" 2>&1>>
${LOG}

RESULTADO=$?
```

```

if [ ${RESULTADO} -eq 0 ] ; then
    ARQUIVO_TESTE=${P_MONTAGEM}/${ARQUIVO}
    ls $ARQUIVO_TESTE >> ${LOG} 2>&1 && cksum $ARQUIVO_TESTE |
    awk {'print $1'} || echo -1

else
    echo -1
fi

umount ${P_MONTAGEM} 2>> ${LOG}

echo
"$(date);${TIMEOUT};${SERVIDOR};${COMPARTILHAMENTO};${USUARIO}
;${SENHA};${RESULTADO}" 2>&1>> ${LOG}
echo "-----" 2>&1>> ${LOG}

```

Customização no agente

Para que este item seja reconhecido pelo zabbix_agentd, a linha abaixo deve ser inserida no arquivo de configuração /etc/zabbix/zabbix_agentd.conf.

```

UserParameter=caseirao_ckfile[*],/etc/zabbix/scripts/caseirao_ckfile.sh $1 $2
$3 $4 $5 $6

```

- Após essa alteração será necessário reiniciar o agente do zabbix
(*/etc/init.d/zabbix-agent restart*)

Caseirao_filedate

O item caseirao_filedate testa se um ponto de montagem está acessível e verifica quantos arquivos anteriores a x dias existem no diretório.

Parâmetros do caseirao_filedate

Key: caseirao_filedate[*]

Este item precisa dos 6 parâmetros abaixo para funcionar.

N	Parametro
1	Timeout em segundos
2	Host
3	Ponto de montagem
4	Usuario
5	Senha
6	quantidade de dias

Exemplo:

caseirao_filedate[15,pfs07,/conectividade/tiss,usrcaspc,caspectrum01,90]

Name

Type

Key

Type of information

Units

Use custom multiplier ☐

Update interval (in sec)

History storage period (in days)

Trend storage period (in days)

Store value

Show value [show value mappings](#)

New application

Applications

- None-
- Caseirao
- caseirao_ckfile
- caseirao_count
- caseirao_filedate
- Cohesion
- CPU
- Disk Operations/s
- Disk Sector/s
- Filesystems

Populates host inventory field

Description

Enabled ☒

- Este item utiliza o tipo de dado Numeric (float)

Script caseirao_filedate.sh

Salvar o script no caminho: /etc/zabbix/scripts/caseirao_filedate.sh

```
#!/bin/bash
TIMEOUT="$1"
SERVIDOR="$2"
COMPARTILHAMENTO="$3"
USUARIO="$4"
SENHA="$5"
TEMPO="$6"
P_MONTAGEM="/tmp${COMPARTILHAMENTO}"
DEBUG=1 # 0 ou 1

if [ $DEBUG -eq 0 ]; then
    LOG=/dev/null
else
    LOG=/tmp/caseirao_filedat.log
fi

test -d "${P_MONTAGEM}" || mkdir -p "${P_MONTAGEM}"

echo "timeout ${TIMEOUT}s mount -t cifs -o
username=${USUARIO},password=${SENHA}
//${SERVIDOR}${COMPARTILHAMENTO} ${P_MONTAGEM}" 2>&1>>
${LOG}

timeout ${TIMEOUT}s mount -t cifs -o
username=${USUARIO},password=${SENHA}
"//${SERVIDOR}${COMPARTILHAMENTO}" "${P_MONTAGEM}" 2>&1>>
${LOG}

RESULTADO=$?
```

```

if [ ${RESULTADO} -eq 0 ] ; then
    find $P_MONTAGEM -mtime +${TEMPO} | wc -l

else
    echo -1
fi

umount ${P_MONTAGEM} 2>> ${LOG}

echo
"$(date);${TIMEOUT};${SERVIDOR};${COMPARTILHAMENTO};${USUARIO}
;${SENHA};${RESULTADO}" 2>&1>> ${LOG}
echo "-----" 2>&1>> ${LOG}

```

Customização no agente

Para que este item seja reconhecido pelo zabbix_agentd, a linha abaixo deve ser inserida no arquivo de configuração /etc/zabbix/zabbix_agentd.conf.

```

UserParameter=caseirao_filedate[*],/etc/zabbix/scripts/caseirao_filedate.sh
$1 $2 $3 $4 $5 $6

```

- Após essa alteração será necessário reiniciar o agente do zabbix
(*/etc/init.d/zabbix-agent restart*)

nt519_log.sh

O item nt519_log.sh checa o tempo de atualização de um arquivo com o nome dinâmico baseado na data (f:\ROBOSYS\LOGIP\LOG_SYS_YYYYMMDD.TXT).

Parâmetros do nt519_log.sh

Este item não possui parâmetros.

Key: nt519_log.sh

Exemplo: nt519_log.sh

Name: Atualizacao Log F:\ROBOSYS\LOGIP\LOG_SYS_201

Type: External check

Key: nt519_log.sh Select

Host interface: NT519.portoseguro.brasil : 10050

Type of information: Numeric (unsigned)

Data type: Decimal

Units: unixtime

Use custom multiplier: ☐ 1

Update interval (in sec): 120

Custom intervals

TYPE	INTERVAL	PERIOD	ACTION
Flexible	Scheduling	50	1-7,00:00-24:00 Remove

[Add](#)

History storage period (in days): 90

Trend storage period (in days): 365

Store value: As is

Show value: As is [show value mappings](#)

New application:

Applications:

- None-
- Cohesion
- CPU
- Filesystems
- General
- ITCM Client
- Log
- Memory
- Network interfaces
- OS

Populates host inventory field: -None-

Description:

- Este item utiliza o tipo de dado Numeric (Unsigned)
- Este item deve ser External check

Script nt519_log.sh

O script deve ser adicionado no diretório ExternalScripts definido no arquivo de configuração do Server ou Proxy.

```
DATA_LOG=$(date "+%Y%m%d")
zabbix_get -s nt519 -p 10050 -k
vfs.file.time[f:\ROBOSYS\LOGIP\LOG_SYS_"$DATA_LOG".TXT] | grep -v
"No such file or directory"
```

clusternode.sh

Este script é utilizado pelo template "**Template Cluster Windows - LLD - Passive -External**" para coletar todos os nodes do cluster e retornar um Json para a criação dos itens de LLD. O script deve ser adicionado no diretório ExternalScripts definido no arquivo de configuração do Server ou Proxy.

```
#!/bin/bash

echo -e "{\"data\":[ "
zabbix_get -s $1 -p 10050 -k system.run["powershell.exe -command \"Get-
WmiObject -class MSCluster_node -namespace root\\mscluster | select
name\""] | sed 1,3d |dos2unix | tr -s " " | sed 's/[[:blank:]]*/ /' | sed
"s:^\{\\\"#CLUSTERNODE\\\"\\\":\\\": \" | sed s:\\\":\\\": | sed '$s/,/}'"
```

clusterresource.sh

Este script é utilizado pelo template "**Template Cluster Windows - LLD - Passive -External**" para coletar todos os recursos do cluster e retornar um Json para a criação dos itens de LLD. O script deve ser adicionado no diretório ExternalScripts definido no arquivo de configuração do Server ou Proxy.

```
#!/bin/bash
```

```
echo -e "{\"data\":[ "
```

```
zabbix_get -s $1 -p 10050 -k system.run["powershell.exe -command \"Get-
WmiObject -class MSCluster_Resource -namespace root\\mscluster | select
name\""] | sed 1,3d |dos2unix | tr -s " " | sed 's/[[:blank:]]*$// ' | sed
"s:^:\\\"{#CLUSTERRESOURCE}\\\"\\\".\" | sed s:$:\\\"\\\",: |sed '$s/,/}/' | sed
's:\\:\\\\\\\\\\\\\\\\:g'
```

clustergroup.sh

Este script é utilizado pelo template "**Template Cluster Windows - LLD - Passive -External**" para coletar todos os grupos do cluster e retornar um Json para a criação dos itens de LLD. O script deve ser adicionado no diretório ExternalScripts definido no arquivo de configuração do Server ou Proxy.

```
#!/bin/bash
```

```
echo -e "{\"data\":[ "
```

```
zabbix_get -s $1 -p 10050 -k system.run["powershell.exe -command \"Get-
WmiObject -class MSCluster_ResourceGroup -namespace root\\mscluster |
select name\""] | sed 1,3d |dos2unix | tr -s " " | sed 's/[[:blank:]]*$// ' | sed
"s:^:\\\"{#CLUSTERGROUP}\\\"\\\".\" | sed s:$:\\\"\\\",: |sed '$s/,/}/' | sed
```

ItmPoolMemberStatServerCurConns.sh

Script utilizado no Template F5 para coletar as informações de conexões no pool.

Type: External check

Key:ltmPoolMemberStatServerCurConns.sh[{HOST.HOST},{HOST.CONN},{\$\$NMP_COMMUNITY}]

Name

Type

Key

```
#!/bin/bash

#ltmPoolMemberStatServerCurConns.sh[{HOST.CONN},{$$NMP_COMMUNITY},{#MEMBER_NAME},{#MEMBER_PORT},{#MEMBER_NODE}]
```

```
HOST=$1;
CONN=$2;
COMM=$3
ZABBIX_PROXY="zabbix-snmp";

#discovery[{#MEMBER_NAME},1.3.6.1.4.1.3375.2.2.5.4.3.1.1,{#MEMBER_P
ORT},1.3.6.1.4.1.3375.2.2.5.4.3.1.4,{#MEMBER_NODE},1.3.6.1.4.1.3375.2.2.
5.4.3.1.28]
#MEMBER_NAME=$3;
#MEMBER_PORT=$4;
#MEMBER_NODE=$5;

function exe(){
    ItmPoolMemberStatPoolName="1.3.6.1.4.1.3375.2.2.5.4.3.1.1";
    ItmPoolMemberStatPort="1.3.6.1.4.1.3375.2.2.5.4.3.1.4";
    ItmPoolMemberStatNodeName="1.3.6.1.4.1.3375.2.2.5.4.3.1.28";
    ItmPoolMemberStatServerCurConns="1.3.6.1.4.1.3375.2.2.5.4.3.1.11";

    SNMPWALK="snmpwalk -v 2c -c $COMM $CONN";
    ZABBIX_SENDER="zabbix_sender -z $ZABBIX_PROXY -s $HOST -p
10051";

    err=0;

    d1=$(date +"%s.%N");
    mapfile -t member < <($SNMPWALK -Onq
$ItmPoolMemberStatPoolName | tr -d '"');
    if [ $? -ne 0 ]
```

```
then
    echo "0";
    exit 1;
fi
d2=$(date +"%s.%N");
mapfile -t port < <($SNMPWALK -Ovq $ItmPoolMemberStatPort);
if [ $? -ne 0 ]
then
    echo "0";
    exit 1;
fi
d3=$(date +"%s.%N");
mapfile -t node < <($SNMPWALK -Ovq
$ItmPoolMemberStatNodeName | tr -d '"');
if [ $? -ne 0 ]
then
    echo "0";
    exit 1;
fi
d4=$(date +"%s.%N");
mapfile -t conn < <($SNMPWALK -Ovq
$ItmPoolMemberStatServerCurConns);
if [ $? -ne 0 ]
then
    echo "0";
    exit 1;
fi
d5=$(date +"%s.%N");
```

```

for ((i=0; i<${#member[@]}; i++))
do
    index=$(echo ${member[$i]} | cut -d' ' -f1' | cut -d'.' -f2-');
    name=$(echo ${member[$i]} | cut -d' ' -f2');

    key="ItnPoolMemberStatServerCurConns[${name},${port[$i]},{node[$
i]}]";

    send="$ZABBIX_SENDER -k $key -o ${conn[$i]}";
    echo $send;
    $send >> /dev/null 2>&1;

    if [ $? -ne 0 ]
    then
        err=$((err+1));
    fi

done

d6=$(date +"%s.%N");

script_status="$ZABBIX_SENDER -k
trap_ItnPoolMemberStatServerCurConns_status -o $err";
#    echo $script_status;
    $script_status;

:<<'LOG_OUT'
    echo "";
    echo "EXEC: $(date +"%Y-%m-%d %H:%M:%S")";

```

```

echo "walk1: $(echo "scale=9; $d2 - $d1" | bc)";
echo "walk2: $(echo "scale=9; $d3 - $d2" | bc)";
echo "walk3: $(echo "scale=9; $d4 - $d3" | bc)";
echo "walk4: $(echo "scale=9; $d5 - $d4" | bc)";
echo " calc: $(echo "scale=9; $d6 - $d5" | bc)";
echo "";
echo "total lines: ${#member[@]}"
echo "total error: $err";
echo "total delta: $(echo "scale=9; $d6 - $d1" | bc)";
echo "";
LOG_OUT

}

#exe $HOST $CONN $COMM $ZABBIX_PROXY >>
/tmp/script_ltmPoolMemberStatServerCurConns_${HOST}.log 2>&1 &
exe $HOST $CONN $COMM $ZABBIX_PROXY >> /dev/null 2>&1 &

echo "1";

```

Esse script alimentará os itens criados pelo LLD na regra de discovery "***Pool member connections discovery***".

Type: SNMPv2 agent

Key: ltmPoolMemberStatPoolName

OID:discovery[{#MEMBER_NAME},1.3.6.1.4.1.3375.2.2.5.4.3.1.1,{#MEMBER_PORT},1.3.6.1.4.1.3375.2.2.5.4.3.1.4,{#MEMBER_NODE},1.3.6.1.4.1.3375.2.2.5.4.3.1.28]

O script deve ser adicionado no diretório ExternalScripts definido no arquivo de configuração do Server ou Proxy.

sysCmTrafficGroupStatusFailoverStatus.sh

Script utilizado no Template F5 para coletar as informações de Failover.

Type: External check

Key:sysCmTrafficGroupStatusFailoverStatus.sh[{HOST.HOST},{HOST.CONN},{SNMP_COMMUNITY}]

Name:

Type:

Key: [Select](#)

Type of information:

Data type:

Units:

Use custom multiplier: ☐

Update interval (in sec):

Custom intervals

TYPE	INTERVAL	PERIOD	ACTION
<input checked="" type="checkbox"/> Flexible	<input type="text" value="Scheduling"/>	<input type="text" value="50"/>	<input type="text" value="1-7,00:00-24:00"/>
			Remove
Add			

History storage period (in days):

Trend storage period (in days):

Store value:

Show value: [show value mappings](#)

New application:

Applications

- None-
- Chaveamento
- Connections
- CPU
- Filesystem
- Memory
- Networks
- OS

```
#!/bin/bash

#sysCmTrafficGroupStatusFailoverStatus.sh[{HOST.CONN},{SNMP_COMM
UNITY},{#GROUP_NAME},{#DEVICE_NAME}]
HOST=$1;
CONN=$2;
COMM=$3
ZABBIX_PROXY="zabbix-snmp";

#discovery[{#GROUP_NAME},1.3.6.1.4.1.3375.2.1.14.5.2.1.1,{#DEVICE_NA
ME},1.3.6.1.4.1.3375.2.1.14.5.2.1.2]
#GROUP_NAME=$3;
#DEVICE_NAME=$4;

sysCmTrafficGroupStatusTrafficGroup="1.3.6.1.4.1.3375.2.1.14.5.2.1.1";
sysCmTrafficGroupStatusDeviceName="1.3.6.1.4.1.3375.2.1.14.5.2.1.2";
sysCmTrafficGroupStatusFailoverStatus="1.3.6.1.4.1.3375.2.1.14.5.2.1.3";

SNMPWALK="snmpwalk -v 2c -c $COMM $CONN";
ZABBIX_SENDER="zabbix_sender -z $ZABBIX_PROXY -s $HOST -p
10051";

err=0;

d1=$(date +"%s.%N");
mapfile -t groups < <($SNMPWALK -Onq
$sysCmTrafficGroupStatusTrafficGroup | tr -d "");
if [ $? -ne 0 ]
```

```
then
    echo "0";
    exit 1;
fi
d2=$(date +"%s.%N");
mapfile -t device < <($SNMPWALK -Ovq
$sysCmTrafficGroupStatusDeviceName | tr -d "");
if [ $? -ne 0 ]
then
    echo "0";
    exit 1;
fi
d3=$(date +"%s.%N");
mapfile -t status < <($SNMPWALK -Ovq
$sysCmTrafficGroupStatusFailoverStatus);
if [ $? -ne 0 ]
then
    echo "0";
    exit 1;
fi
d4=$(date +"%s.%N");

for ((i=0; i<${#groups[@]}; i++))
do
    index=$(echo ${groups[$i]} | cut -d ' ' -f1 | cut -d '.' -f2-);
    group=$(echo ${groups[$i]} | cut -d ' ' -f2);
    key="sysCmTrafficGroupStatusFailoverStatus[${group},${device[$i]}";

    send="$ZABBIX_SENDER -k $key -o ${status[$i]}";
```

```
#      echo $send;
      $send >> /dev/null 2>&1;

      if [ $? -ne 0 ]
      then
          err=$((err+1));
      fi

done

d5=$(date +"%s.%N");

script_status="$ZABBIX_SENDER -k
trap_sysCmTrafficGroupStatusFailoverStatus_status -o $err";
#echo $script_status;
$script_status >> /dev/null 2>&1;

:<<'LOG_OUT'
echo "";
echo "EXEC: $(date +"%Y-%m-%d %H:%M:%S")";
echo "walk1: $(echo "scale=9; $d2 - $d1" | bc)";
echo "walk2: $(echo "scale=9; $d3 - $d2" | bc)";
echo "walk3: $(echo "scale=9; $d4 - $d3" | bc)";
echo " calc: $(echo "scale=9; $d5 - $d4" | bc)";
echo "";
echo "total lines: ${#groups[@]}"
echo "total error: $err";
echo "total delta: $(echo "scale=9; $d5 - $d1" | bc)";
echo "";
```

```
LOG_OUT
```

```
echo "1";
```

Esse script alimentará os itens criados pelo LLD na regra de discovery "**Traffic Group Failover Status discovery**".

Type: SNMPv2 agent

Key: TrafficGroupStatusFailoverStatus_discovery

OID:discovery[#{GROUP_NAME},1.3.6.1.4.1.3375.2.1.14.5.2.1.1,#{DEVICE_NAME},1.3.6.1.4.1.3375.2.1.14.5.2.1.2]

O script deve ser adicionado no diretório ExternalScripts definido no arquivo de configuração do Server ou Proxy.

snmp_HR_proc_num.sh

Script utilizado no Template McAfee para coletar as informações do processo {\$PROCESSO}.

Type: External check

Key:

snmp_HR_proc_num.sh[#{HOST.CONN},{\${SNMP_COMMUNITY}},{\${PROCESSO}]

Name

Type

Key

Host interface

Type of information

Data type

Units

Use custom multiplier ☐

Update interval (in sec)

Custom intervals

TYPE	INTERVAL	PERIOD	ACTION
<input checked="" type="checkbox"/> Flexible	<input type="text" value="Scheduling"/>	<input type="text" value="50"/>	<input type="text" value="1-7,00:00-24:00"/>
			Remove
Add			

History storage period (in days)

Trend storage period (in days)

Store value

Show value [show value mappings](#)

New application

Applications

- None-
- CPU
- Filesystem
- Memory
- Networks
- OS
- Processos**

```
#!/bin/bash
```

```
HOST=$1;
```

```
COMM=$2;
```

```
PROC=$3;
```

```
SNMPWALK="snmpwalk -v 2c -c $COMM $HOST";
```

```
#TYPE=$2;
```

```
#AUTH_USER=$3;
```

```
#AUTH_PASS=$4;
#PRIV_USER=$5;
#PRIV_PASS=$6;

hrSWRunName="1.3.6.1.2.1.25.4.2.1.2";
hrSWRunPath="1.3.6.1.2.1.25.4.2.1.4";
hrSWRunParameters="1.3.6.1.2.1.25.4.2.1.5"
hrSWRunStatus="1.3.6.1.2.1.25.4.2.1.7";

$SNMPWALK -Ovq $hrSWRunName 2>/dev/null | grep $PROC | wc -l
```

O script deve ser adicionado no diretório ExternalScripts definido no arquivo de configuração do Server ou Proxy.

sdm-check.sh

Item para checagem da disponibilidade da API do CA SDM. Esse script aceita um parâmetro, que deve ser o IP do servidor SDM que será checado.

Type: External check

Key: sdm-check.sh[172.26.26.12]

The screenshot shows the Zabbix web interface with the 'Configuration' menu selected. The 'Items' section is active, displaying the configuration for a new item named 'sdm check login \$1'. The configuration includes:

- Name:** sdm check login \$1
- Type:** External check
- Key:** sdm-check.sh[172.26.26.12]
- Host interface:** 172.27.47.88 : 10050
- Type of information:** Numeric (float)
- Units:** s
- Use custom multiplier:** 1
- Update interval (in sec):** 60
- Custom intervals:** A table with columns Type, Interval, Period, and Action. The first row is 'Flexible' with an interval of 50 and a period of 1-7:00:00-24:00.
- History storage period (in days):** 90
- Trend storage period (in days):** 365
- Store value:** As is
- Show value:** As is
- New application:** A dropdown menu showing options: -None-, CPU, Disk Operations/s, Disk Sector/s, DNS, and Filesystems.

```
#!/bin/bash
STARTTIME=`date +%s.%N`

ENDPOINT="http://$1:8080/axis/services/USD_R11_WebService"
SDM_USER="usrcaspc"
SDM_PASS="caspectrum01"

# AUTENTICACAO
porto_sdm_auth(){
timeout 15s curl --silent \
    --data \
    @- \
    --header 'Content-Type: application/soap+xml; charset=utf-8' \
```



```

--header 'SOAPAction: "" \
--user-agent "" \
${ENDPOINT} <<EOF | xmllint --format -
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://www.ca.com/UnicenterServicePlus/ServiceDesk">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:login>
      <username>${SDM_USER}</username>
      <password>${SDM_PASS}</password>
    </ser:login>
  </soapenv:Body>
</soapenv:Envelope>
EOF
}

# LOGOUT
porto_sdm_logout(){
timeout 15s curl --silent \
  --data \
  @- \
  --header 'Content-Type: application/soap+xml; charset=utf-8' \
  --header 'SOAPAction: "" \
  --user-agent "" \
  ${ENDPOINT} <<EOF | xmllint --format -
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope

```

```

xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://www.ca.com/UnicenterServicePlus/ServiceDesk">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:logout>
      <sid>$SDM_LOGIN_RETURN</sid>
    </ser:logout>
  </soapenv:Body>
</soapenv:Envelope>
EOF
}

SDM_LOGIN_RETURN=$(porto_sdm_auth 2> /dev/null| grep
loginReturn | cut -d\> -f2 | cut -d\< -f1 )

if [ -z ${SDM_LOGIN_RETURN} ] || [ ${SDM_LOGIN_RETURN} -eq
"1005" ] ; then
  echo 0
else
  porto_sdm_logout &> /dev/null

  ENDTIME=$(date +%s.%N)
  TIMEDIFF=$(echo "$ENDTIME - $STARTTIME" | bc | awk -F"."
'{print $1"."substr($2,1,3)}')
  echo ${TIMEDIFF}
fi

```

O script deve ser adicionado no diretório ExternalScripts definido no arquivo de configuração do Server ou Proxy.

mountpoint.sh

Item para realizar o mapeamento dos pontos de montagem nos Windows utilizando o PowerShell.

Foi declarado a hostmacro {\$HOST.PORT} nos templates pois a macro {HOST.PORT} não é suportado na key do item.

The screenshot shows the Zabbix web interface for configuring a discovery rule. The 'Discovery rule' tab is active. The configuration details are as follows:

- Name:** Mounted filesystem discovery (sub)
- Type:** External check
- Key:** mountpoint.sh[{\$HOST.CONN},{\$HOST.PORT}]
- Update interval (in sec):** 3600
- Custom intervals:**

Type	Interval	Period	Action
Flexible	50	1-7,00:00-24:00	Remove
- Keep lost resources period (in days):** 30
- Description:** (Empty text area)
- Enabled:** ☒
- Buttons:** Update, Clone, Delete, Cancel

LLD configurado com filtro para descartar algumas informações desnecessárias.

Discovery rule **Filters**

Type of calculation: And/Or

Label	Macro	Regular expression	Action
A	{#FS_NAME}	matches @discard_NT_mount	Remove

[Add](#)

[Update](#) [Clone](#) [Delete](#) [Cancel](#)

	1	>	[?](\Volume{	[Result is FALSE]
	2	>	NULL	[Result is FALSE]
	3	>	YourKit Profiler	[Result is FALSE]

☒ discard_NT_mount

```
#!/bin/bash
```

```
mapfile -t arr < <(zabbix_get -s $1 -p $2 -k system.run["PowerShell -
Command \"Get-WmiObject Win32_Volume -Filter 'DriveLetter=\"null\"\" |
select Name\""] | grep '\\ ' | tr -d '\\r' | sed 's/^ *//;s/ *$//' | sed -s 's/[\\$//g');
```

```
#echo "num: ${#arr[@]}";
```

```
json="{\"data\":[";
```

```
if [ ${#arr[@]} -gt 0 ]
```

```
then
```

```
    for ((i=0;i<${#arr[@]};i++))
```

```
    do
```

```
#        echo -e "${i\\t\"${arr[i]}\\\"";
```

```
        json+=$(echo -E \"{\"{#FS_NAME}\"\":\"${arr[i]}\\\", \" | sed -s
's/^\\\\\\\\\\\\\\\\g');
```

```
    done
```

```
    json=$(echo $json | sed -s 's/,,$//g');
```

```
else
```

```
    json+="{\"{#FS_NAME}\"\":\"NULL\\\"}"
```

```
fi
```

```
json+="}";

#echo "";
echo -E $json
```

Clear TRAP

O Módulo de clear trap tem como objetivo limpar um alerta de trap manualmente pelo front-end.

O acesso dessa funcionalidade está liberado apenas para os usuário que estiverem configurados com o User type “Zabbix Admin”.

Para limpar uma trap basta acessar Configuration -> Clear_Traps e clicar em Clear_Trap

Severity	Status	Info	Last change	Age	Ack	Host	Name	Clear
Average	PROBLEM	10/15/2017 09:42:40 PM	12d 9h 30m	No events	VN00000_DB_CS1-NAS	CEL MASTER CTL FAULT	Clear	
Average	PROBLEM	10/15/2017 11:36:51 AM	12d 9h 37m	No events	phoenix19labrac	CPD SMD INTERFACE ERROR	Clear	
High	PROBLEM	10/14/2017 09:14:07 PM	12d 22h 58m	No events	u003	[Oracle] Alarma de Integracao OEM - TargetName: u003.portseguro.brazil.1830.TargetType: Agent	Clear	
High	PROBLEM	10/14/2017 08:53:00 PM	13d 10m	No events	u003	[Oracle] Alarma de Integracao OEM - TargetName: u003.portseguro.brazil.1830.TargetType: Agent	Clear	
High	PROBLEM	10/14/2017 01:24:46 PM	13d 6h 48m	No events	VN00000-CS0-DC-NAS	[Storage NAS] Filesystem size threshold crossed: u003	Clear	
High	PROBLEM	10/12/2017 08:01:07 AM	15d 12h 12m	No events	02027.portseguro.brazil	[Oracle] Alarma de Integracao OEM - TargetName: LISTENER_SCAN1_listener.TargetType: Listener	Clear	
High	PROBLEM	10/11/2017 07:58:20 PM	16d 18m	No events	VN00000-CS0-DC-NAS	[Storage NAS] Block soft quota systemssu003	Clear	
High	PROBLEM	10/11/2017 05:52:11 PM	16d 3h 11m	No events	02027.portseguro.brazil	[Oracle] Alarma de Integracao OEM - TargetName: dbp_gst_001.TargetType: Database Instance	Clear	
High	PROBLEM	10/11/2017 04:25:10 PM	16d 3h 48m	No events	11288	CISCO WIRELESS M.O STATUS NOTIFICATION - AP_LAB_PORTO	Clear	
High	PROBLEM	10/11/2017 03:38:15 PM	16d 4h 35m	No events	VN00000-CS0-DC-NAS	[Storage NAS] Block soft quota ADM02	Clear	
High	PROBLEM	10/11/2017 03:38:46 PM	16d 5h 4m	No events	clever001_uCenter	[VMware] Caminho de multipath degradado	Clear	
Warning	PROBLEM	10/11/2017 03:05:15 PM	16d 5h 4m	No events	000328	[Rioses SAN] SWITCH FIBRE CHANNEL PORT SCAN - VMAX_003_PA001 index 187	Clear	
Warning	PROBLEM	10/11/2017 03:05:15 PM	16d 5h 4m	No events	000328	[Rioses SAN] SWITCH FIBRE CHANNEL PORT SCAN - VMAX_003_PA001 index 130	Clear	
Warning	PROBLEM	10/11/2017 03:05:15 PM	16d 5h 4m	No events	000327	[Rioses SAN] SWITCH FIBRE CHANNEL PORT SCAN - VMAX_003_PA001 index 210	Clear	
Warning	PROBLEM	10/11/2017 03:05:15 PM	16d 5h 4m	No events	000328	[Rioses SAN] SWITCH FIBRE CHANNEL PORT SCAN - VMAX_003_PA001 index 300	Clear	
Warning	PROBLEM	10/11/2017 03:05:15 PM	16d 5h 4m	No events	000328	[Rioses SAN] SWITCH FIBRE CHANNEL PORT SCAN - VMAX_003_PA001 index 188	Clear	
Warning	PROBLEM	10/11/2017 03:05:15 PM	16d 5h 4m	No events	000328	[Rioses SAN] SWITCH FIBRE CHANNEL PORT SCAN - VMAX_003_PA001 index 181	Clear	
Warning	PROBLEM	10/11/2017 03:05:15 PM	16d 5h 4m	No events	000327	[Rioses SAN] SWITCH FIBRE CHANNEL PORT SCAN - VMAX_003_PA001 index 187	Clear	

É possível filtrar as traps por grupo e/ou hosts.

Severity	Status	Info	Last change	Age	Ack	Host	Name	Clear
High	PROBLEM	10/14/2017 09:14:07 PM	12d 22h 2m	No events	u003	[Oracle] Alarma de Integracao OEM - TargetName: u003.portseguro.brazil.1830.TargetType: Agent	Clear	
High	PROBLEM	10/14/2017 08:53:00 PM	13d 13m	No events	u003	[Oracle] Alarma de Integracao OEM - TargetName: u003.portseguro.brazil.1830.TargetType: Agent	Clear	

Após clicar em `Clear_Trap`, deve aparecer uma mensagem de trap com o texto “Clear Manual” e o alias do usuário que executou a limpeza.



Todos os clears executados pelo módulo serão logados no arquivo `/var/log/zabbix/zabbix_agentd_clear.log` que está dentro do servidor `zabbix-web.portoseguro.brasil`.

```
zabbix-web (LI2455)[~]# tail /var/log/zabbix/zabbix_agentd_clear.log
zabbix-snmp.portoseguro.brasil 07:40:51 2017/10/17 - ZBXTRAP
172.26.46.57 cpqSm2InterfaceError Clear Manual Usuario: f0116305
zabbix-nix.portoseguro.brasil 07:40:51 2017/10/17 - ZBXTRAP
u503.portoseguro.brasil OracleEM Hostname= u503.portoseguro.brasil .
Target Name= u503.portoseguro.brasil:1830 . Severity= Fatal Clear Manual
Usuario: f0116305
zabbix-nix.portoseguro.brasil 07:40:52 2017/10/17 - ZBXTRAP
u503.portoseguro.brasil OracleEM Hostname= u503.portoseguro.brasil .
Target Name= u503.portoseguro.brasil:1830 . Severity= Critical Clear Manual
Usuario: f0116305
zabbix-snmp.portoseguro.brasil 07:40:52 2017/10/17 - ZBXTRAP
172.26.47.58 filesystem size threshold . crossed ./vida) Clear Manual
Usuario: f0116305
zabbix-nix.portoseguro.brasil 07:40:53 2017/10/17 - ZBXTRAP
li2327.portoseguro.brasil OracleEM Hostname= li2327.portoseguro.brasil .
Target Name= LISTENER_SCAN1_clsrepr . Severity= Fatal Clear Manual
Usuario: f0116305
zabbix-snmp.portoseguro.brasil 07:40:54 2017/10/17 - ZBXTRAP
172.26.47.58 Block soft quota crossed (fs ./sistemasauade, Clear Manual
Usuario: f0116305
zabbix-nix.portoseguro.brasil 07:40:54 2017/10/17 - ZBXTRAP
```

li252.portoseguro.brasil OracleEM Hostname= li252.portoseguro.brasil .
 Target Name= dfpr_pri_dfpr1 . Severity= Critical Clear Manual Usuario:
 f0116305

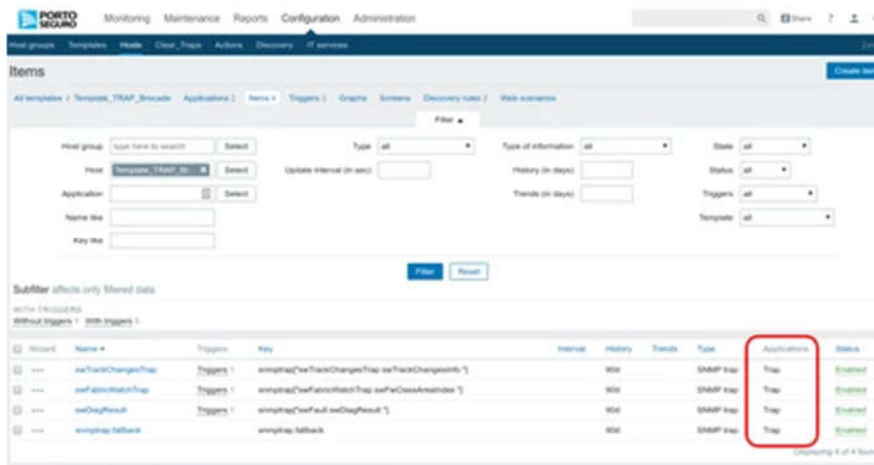
zabbix-snmp.portoseguro.brasil 07:40:55 2017/10/17 - ZBXTRAP
 172.26.47.58 Block soft quota crossed (fs ./ADM02, Clear Manual Usuario:
 f0116305

zabbix-vmware.portoseguro.brasil 07:40:55 2017/10/17 - ZBXTRAP
 172.26.18.253 vpxdAlarmInfo - vmwVpxdTargetObjType .
 vmwVpxdNewStatus.Red.vmwVpxdObjValue .Caminho de multipath
 degradado. Clear Manual Usuario: f0116305

zabbix-vmware.portoseguro.brasil 07:40:55 2017/10/17 - ZBXTRAP
 172.26.18.253 vpxdAlarmInfo - vmwVpxdTargetObjType .
 vmwVpxdNewStatus.Red.vmwVpxdObjValue .Caminho de multipath
 degradado. Clear Manual Usuario: f0116305

Pré-requisitos

- Para que este módulo funcione corretamente o item de trap deve estar configurado com o application "Trap"



- O parâmetro "Server" no arquivo /etc/zabbix/zabbix_agentd.conf de todos os proxys devem conter o servidor zabbix-web.portoseguro.brasil para que esta solução funcione.

Ex: Server=zabbix-nix.portoseguro.brasil,zabbix-server.portoseguro.brasil,zabbix-server-homologacao.portoseguro.brasil,**zabbix-web.portoseguro.brasil**

- As triggers das traps precisam conter a condição da string "Clear Manual" para funcionar:

Ex:

The screenshot shows the Zabbix Triggers page. The trigger is named 'Caminho de multipath degradado' with a severity of 'High'. The expression is '[Template_TRAP_VMware SNMPtrap]<=0&nameinfo - vm/vpadTargetObjType - vm/vpadNewStatus - Red - vm/vpadObjValue - "Caminho de multipath degradado" - str(Clear Manual)=0'. The 'str(Clear Manual)=0' part is highlighted with a red box.

The screenshot shows the Zabbix Trigger prototypes page. There are four trigger prototypes listed, all with a severity of 'High' or 'Warning'. Each prototype has an expression that includes the condition 'str(Clear Manual)=0', which is highlighted with a red box in each row.

Instalação

1- Criar o arquivo /usr/share/zabbix/tr_trap_clear.php com o conteúdo abaixo:

```
<?php
/*
** Zabbix
** Copyright (C) 2001-2016 Zabbix SIA
**
** This program is free software; you can redistribute it and/or modify
** it under the terms of the GNU General Public License as published by
** the Free Software Foundation; either version 2 of the License, or
** (at your option) any later version.
**
** This program is distributed in the hope that it will be useful,
** but WITHOUT ANY WARRANTY; without even the implied warranty of
** MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See
the
** GNU General Public License for more details.
**
** You should have received a copy of the GNU General Public License
** along with this program; if not, write to the Free Software
** Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301,
USA.
**/

require_once dirname(__FILE__).'/include/config.inc.php';

$page['file'] = 'tr_trap_clear.php';
$page['title'] = _('Clear Trap');
```

```

$page['scripts'] = ['class.cswitcher.js'];
$page['type'] = detect_page_type(PAGE_TYPE_HTML);

if ($page['type'] == PAGE_TYPE_HTML) {
    define('ZBX_PAGE_DO_REFRESH', 1);
}

require_once dirname(__FILE__).'include/page_header.php';

// VARTYPE OPTIONAL  FLAGS      VALIDATIONEXCEPTION
$fields = [
    'groupid' =>          [T_ZBX_INT, O_OPT, P_SYS,   DB_ID,
                          null],
    'hostid' =>          [T_ZBX_INT, O_OPT, P_SYS,
    DB_ID,                null],
    'fullscreen' =>      [T_ZBX_INT, O_OPT, P_SYS,
    IN('0,1'),           null],
    'btnSelect' =>       [T_ZBX_STR, O_OPT, null,      null,
    null],
    // filter
    'filter_rst' =>      [T_ZBX_STR, O_OPT, P_SYS,   null,
    null],
    'filter_set' =>      [T_ZBX_STR, O_OPT, P_SYS,   null,
    null],
    'show_triggers' =>   [T_ZBX_INT, O_OPT, null, null,
    null],
    'show_events' =>     [T_ZBX_INT, O_OPT, P_SYS,   null,
    null],
    'ack_status' =>      [T_ZBX_INT, O_OPT, P_SYS,

```

```

        null,          null],
        'show_severity' =>      [T_ZBX_INT, O_OPT, P_SYS,    null,
        null],
        'show_details' =>      [T_ZBX_INT, O_OPT, null,null,
        null],
        'show_maintenance' =>  [T_ZBX_INT, O_OPT, null,null,
        null],
        'status_change_days' => [T_ZBX_INT, O_OPT, null,BETWEEN(1,
DAY_IN_YEAR * 2), null],
        'status_change' =>      [T_ZBX_INT, O_OPT, null,null,
        null],
        'txt_select' =>          [T_ZBX_STR, O_OPT, null,
        null,          null],
        'application' =>        [T_ZBX_STR, O_OPT, null,      null,
        null],
        'inventory' =>          [T_ZBX_STR, O_OPT, null,      null,
        null],
        // sort and sortorder
        'sort' =>                [T_ZBX_STR, O_OPT, P_SYS,
IN("description","lastchange","priority"), null],
        'sortorder' =>          [T_ZBX_STR, O_OPT, P_SYS,
IN("".ZBX_SORT_DOWN."",".ZBX_SORT_UP.""),    null]
    ];
    check_fields($fields);

/*
 * Permissions
 */
if (getRequest('groupid') && !API::HostGroup()-

```

```

>isReadable([getRequest('groupid')])) {
    access_deny();
}
if (getRequest('hostid') && !API::Host()->isReadable([getRequest('hostid')])) {
    access_deny();
}

/*
 * Filter
 */
$config = select_config();

$pageFilter = new CPageFilter([
    'groups' => [
        'monitored_hosts' => true,
        'with_monitored_triggers' => true
    ],
    'hosts' => [
        'monitored_hosts' => true,
        'with_monitored_triggers' => true
    ],
    'hostid' => getRequest('hostid'),
    'groupid' => getRequest('groupid')
]);
$_REQUEST['groupid'] = $pageFilter->groupid;
$_REQUEST['hostid'] = $pageFilter->hostid;

// filter set
if (hasRequest('filter_set')) {

```

```

        CProfile::update('web.tr_trap_clear.filter.show_details',
getRequest('show_details', 0), PROFILE_TYPE_INT);
        CProfile::update('web.tr_trap_clear.filter.show_maintenance',
getRequest('show_maintenance', 0), PROFILE_TYPE_INT);
        CProfile::update('web.tr_trap_clear.filter.show_severity',
            getRequest('show_severity',
TRIGGER_SEVERITY_NOT_CLASSIFIED), PROFILE_TYPE_INT
        );
        CProfile::update('web.tr_trap_clear.filter.txt_select',
getRequest('txt_select', ""), PROFILE_TYPE_STR);
        CProfile::update('web.tr_trap_clear.filter.status_change',
getRequest('status_change', 0), PROFILE_TYPE_INT);
        CProfile::update('web.tr_trap_clear.filter.status_change_days',
getRequest('status_change_days', 14),
            PROFILE_TYPE_INT
        );
        #CProfile::update('web.tr_trap_clear.filter.application',
getRequest('application'), PROFILE_TYPE_STR);
        CProfile::update('web.tr_trap_clear.filter.application', 'Trap',
PROFILE_TYPE_STR);

// show triggers
// when this filter is set to "All" it must not be remembered in the profiles
because it may render the
// whole page inaccessible on large installations.
#if (getRequest('show_triggers') != TRIGGERS_OPTION_ALL) {
#    CProfile::update('web.tr_trap_clear.filter.show_triggers',
getRequest('show_triggers'), PROFILE_TYPE_INT);
#}

```

```
CProfile::update('web.tr_trap_clear.filter.show_triggers', 3,
PROFILE_TYPE_INT);
// show events
$showEvents = getRequest('show_events',
EVENTS_OPTION_NOEVENT);
if ($config['event_ack_enable'] == EVENT_ACK_ENABLED ||
$showEvents != EVENTS_OPTION_NOT_ACK) {
    CProfile::update('web.tr_trap_clear.filter.show_events',
$showEvents, PROFILE_TYPE_INT);
}

// ack status
if ($config['event_ack_enable'] == EVENT_ACK_ENABLED) {
    CProfile::update('web.tr_trap_clear.filter.ack_status',
getRequest('ack_status', ZBX_ACK_STS_ANY), PROFILE_TYPE_INT);
}

// update host inventory filter
$inventoryFields = [];
$inventoryValues = [];
foreach (getRequest('inventory', []) as $field) {
    if ($field['value'] === '') {
        continue;
    }

    $inventoryFields[] = $field['field'];
    $inventoryValues[] = $field['value'];
}
CProfile::updateArray('web.tr_trap_clear.filter.inventory.field',
```

```
$inventoryFields, PROFILE_TYPE_STR);
    CProfile::updateArray('web.tr_trap_clear.filter.inventory.value',
$inventoryValues, PROFILE_TYPE_STR);
}
elseif (hasRequest('filter_rst')) {
    DBStart();
    CProfile::delete('web.tr_trap_clear.filter.show_triggers');
    CProfile::delete('web.tr_trap_clear.filter.show_details');
    CProfile::delete('web.tr_trap_clear.filter.show_maintenance');
    CProfile::delete('web.tr_trap_clear.filter.show_events');
    CProfile::delete('web.tr_trap_clear.filter.ack_status');
    CProfile::delete('web.tr_trap_clear.filter.show_severity');
    CProfile::delete('web.tr_trap_clear.filter.txt_select');
    CProfile::delete('web.tr_trap_clear.filter.status_change');
    CProfile::delete('web.tr_trap_clear.filter.status_change_days');
    CProfile::delete('web.tr_trap_clear.filter.application');
    CProfile::deleteIdx('web.tr_trap_clear.filter.inventory.field');
    CProfile::deleteIdx('web.tr_trap_clear.filter.inventory.value');
    DBend();
}

#if (hasRequest('filter_set') && getRequest('show_triggers') ==
TRIGGERS_OPTION_ALL) {
#    $showTriggers = TRIGGERS_OPTION_ALL;
#}
#else {
#    $showTriggers = CProfile::get('web.tr_trap_clear.filter.show_triggers',
TRIGGERS_OPTION_RECENT_PROBLEM);
#}
```

```
#$showTriggers = TRIGGERS_OPTION_IN_PROBLEM;
$showTriggers =
CProfile::get('web.tr_trap_clear.filter.show_triggers',TRIGGERS_OPTION_IN
_PROBLEM);
$showDetails = CProfile::get('web.tr_trap_clear.filter.show_details', 0);
$showMaintenance =
CProfile::get('web.tr_trap_clear.filter.show_maintenance', 1);
$showSeverity = CProfile::get('web.tr_trap_clear.filter.show_severity',
TRIGGER_SEVERITY_NOT_CLASSIFIED);
$txtSelect = CProfile::get('web.tr_trap_clear.filter.txt_select', "");
$showChange = CProfile::get('web.tr_trap_clear.filter.status_change', 0);
$statusChangeBydays =
CProfile::get('web.tr_trap_clear.filter.status_change_days', 14);
$ackStatus = ($config['event_ack_enable'] == EVENT_ACK_DISABLED)
    ? ZBX_ACK_STS_ANY :
CProfile::get('web.tr_trap_clear.filter.ack_status', ZBX_ACK_STS_ANY);
$showEvents = CProfile::get('web.tr_trap_clear.filter.show_events',
EVENTS_OPTION_NOEVENT);

// check event acknowledges
if ($config['event_ack_enable'] == EVENT_ACK_DISABLED && $showEvents
== EVENTS_OPTION_NOT_ACK) {
    $showEvents = EVENTS_OPTION_NOEVENT;
}

// fetch filter from profiles
$filter = [
    'application' => CProfile::get('web.tr_trap_clear.filter.application',
'Trap'),
```



```
'inventory' => []  
];  
  
foreach (CProfile::getArray('web.tr_trap_clear.filter.inventory.field', []) as $i =>  
$field) {  
    $filter['inventory'][] = [  
        'field' => $field,  
        'value' => CProfile::get('web.tr_trap_clear.filter.inventory.value',  
null, $i)  
    ];  
}  
  
/*  
 * Page sorting  
 */  
$sortField = getRequest('sort', CProfile::get('web.'.$page['file'].'.sort',  
'lastchange'));  
$sortOrder = getRequest('sortorder',  
CProfile::get('web.'.$page['file'].'.sortorder', ZBX_SORT_DOWN));  
  
CProfile::update('web.'.$page['file'].'.sort', $sortField, PROFILE_TYPE_STR);  
CProfile::update('web.'.$page['file'].'.sortorder', $sortOrder,  
PROFILE_TYPE_STR);  
  
/*  
 * Display  
 */  
$triggerWidget = (new CWidget())->setTitle(_('List Trap with PROBLEM'));
```

```
$rightForm = (new CForm('get'))
    ->addVar('fullscreen', $_REQUEST['fullscreen']);

$controls = new CList();
$controls->addItem([_('Group').SPACE, $pageFilter->getGroupsCB()]);
$controls->addItem([_('Host').SPACE, $pageFilter->getHostsCB()]);
$controls->addItem(get_icon('fullscreen', ['fullscreen' =>
$_REQUEST['fullscreen']]));

$rightForm->addItem($controls);

$triggerWidget->setControls($rightForm);

// filter
#application' => $filter['application'],
$filterFormView = new CView('common.filter.trigger', [
    'overview' => false,
    'filter' => [
        'filterid' => 'web.tr_trap_clear.filter.state',
        'showTriggers' => $showTriggers,
        'ackStatus' => $ackStatus,
        'showEvents' => $showEvents,
        'showSeverity' => $showSeverity,
        'statusChange' => $showChange,
        'statusChangeDays' => $statusChangeBydays,
        'application' => 'Trap',
        'showDetails' => $showDetails,
        'txtSelect' => $txtSelect,
```

```
        'inventory' => $filter['inventory'],
        'showMaintenance' => $showMaintenance,
        'hostId' => getRequest('hostid'),
        'groupId' => getRequest('groupid'),
        'fullScreen' => getRequest('fullscreen')
    ],
    'config' => $config
]);

#$filterForm = $filterFormView->render();
#$triggerWidget->addItem($filterForm);

/*
 * Form
 */
$triggerForm = (new CForm('get', 'zabbix.php'))
    ->setName('tr_trap_clear')
    ->addVar('backurl', $page['file'])
    ->addVar('acknowledge_type', ZBX_ACKNOWLEDGE_PROBLEM);

/*
 * Table
 */
$switcherName = 'trigger_switchers';

if ($showEvents == EVENTS_OPTION_ALL || $showEvents ==
EVENTS_OPTION_NOT_ACK) {
    $showHideAllDiv = (new CColHeader(
        (new CDiv())
```

```

        ->addClass(ZBX_STYLE_TREEVIEW)
        ->setId($switcherName)
        ->addItem((new CSpan())-
>addClass(ZBX_STYLE_ARROW_RIGHT))
        ))->addClass(ZBX_STYLE_CELL_WIDTH);
    }
    else {
        $showHideAllDiv = null;
    }

    if ($config['event_ack_enable']) {
        $headerCheckBox = (new CColHeader(
            (new CCheckBox('all_eventids'))
                ->onClick("checkAll('".$triggerForm->GetName()."',
'all_eventids', 'eventids');")
        ))->addClass(ZBX_STYLE_CELL_WIDTH);
    }
    else {
        $headerCheckBox = null;
    }

    $triggerTable = (new CTableInfo())
        ->setHeader([
            $showHideAllDiv,
            make_sorting_header(_('Severity'), 'priority', $sortField,
$sortOrder),
            _('Status'),
            _('Info'),
            make_sorting_header(_('Last change'), 'lastchange', $sortField,

```

```
$sortOrder),
    _('Age'),
    ($showEvents == EVENTS_OPTION_ALL || $showEvents ==
EVENTS_OPTION_NOT_ACK) ? _('Duration') : null,
    $config['event_ack_enable'] ? _('Ack') : null,
    _('Host'),
    _('Name'),
    _('Clear')
]);

// get triggers
$options = [
    'output' => ['triggerid', $sortField],
    'monitored' => true,
    'skipDependent' => true,
    'sortfield' => $sortField,
    'limit' => $config['search_limit'] + 1
];

if ($pageFilter->hostsSelected) {
    if ($pageFilter->hostid > 0) {
        $options['hostids'] = $pageFilter->hostid;
    }
    elseif ($pageFilter->groupid > 0) {
        $options['groupids'] = $pageFilter->groupid;
    }
}
else {
    $options['hostids'] = [];
```

```
}

// inventory filter
if ($filter['inventory']) {
    $inventoryFilter = [];
    foreach ($filter['inventory'] as $field) {
        $inventoryFilter[$field['field']] = $field['value'];
    }

    $hosts = API::Host()->get([
        'output' => ['hostid'],
        'hostids' => isset($options['hostids']) ? $options['hostids'] : null,
        'searchInventory' => $inventoryFilter
    ]);
    $options['hostids'] = zbx_objectValues($hosts, 'hostid');
}

// application filter
if ($filter['application'] !== "") {
    $applications = API::Application()->get([
        'output' => ['applicationid'],
        'hostids' => isset($options['hostids']) ? $options['hostids'] : null,
        'search' => ['name' => 'Trap']
    ]);
    $options['applicationids'] = zbx_objectValues($applications,
'applicationid');
}
```

```
if (!zbx_empty($txtSelect)) {
    $options['search'] = ['description' => $txtSelect];
}

if ($showTriggers == TRIGGERS_OPTION_RECENT_PROBLEM) {
    $options['only_true'] = 1;
}
elseif ($showTriggers == TRIGGERS_OPTION_IN_PROBLEM) {
    $options['filter'] = ['value' => TRIGGER_VALUE_TRUE];
}
if ($ackStatus == ZBX_ACK_STS_WITH_UNACK) {
    $options['withUnacknowledgedEvents'] = 1;
}
if ($ackStatus == ZBX_ACK_STS_WITH_LAST_UNACK) {
    $options['withLastEventUnacknowledged'] = 1;
}
if ($showSeverity > TRIGGER_SEVERITY_NOT_CLASSIFIED) {
    $options['min_severity'] = $showSeverity;
}
if ($showChange) {
    $options['lastChangeSince'] = time() - $statusChangeBydays *
SEC_PER_DAY;
}
if (!$showMaintenance) {
    $options['maintenance'] = false;
}
$triggers = API::Trigger()->get($options);

order_result($triggers, $sortField, $sortOrder);
```

```
#miyagi ->
$url = new CUrl();
$paging = getPagingLine($triggers, $sortOrder, $url);

$triggers = API::Trigger()->get([
    'triggerids' => zbx_objectValues($triggers, 'triggerid'),
    'output' => API_OUTPUT_EXTEND,
    'selectHosts' => [
        'hostid', 'name', 'description', 'status', 'maintenanceid',
'maintenance_status', 'maintenance_type'
    ],
    'selectItems' => ['itemid', 'hostid', 'name', 'key_', 'value_type'],
    'selectDependencies' => API_OUTPUT_EXTEND,
    'selectLastEvent' => true,
    'expandDescription' => true,
    'preservekeys' => true
]);

$triggers = CMacrosResolverHelper::resolveTriggerUrls($triggers);
if ($showDetails) {
    $triggers =
CMacrosResolverHelper::resolveTriggerExpressions($triggers,
    ['html' => true, 'resolve_usermacros' => true, 'resolve_macros'
=> true]
    );
}

order_result($triggers, $sortField, $sortOrder);
```



```
// sort trigger hosts by name
foreach ($triggers as &$trigger) {
    if (count($trigger['hosts']) > 1) {
        order_result($trigger['hosts'], 'name', ZBX_SORT_UP);
    }
}
unset($trigger);

$triggerIds = zbx_objectValues($triggers, 'triggerid');

// get editable triggers
$triggerEditable = API::Trigger()->get([
    'triggerids' => $triggerIds,
    'output' => ['triggerid'],
    'editable' => true,
    'preservekeys' => true
]);

// get events
if ($config['event_ack_enable']) {
    // get all unacknowledged events, if trigger has unacknowledged event
    => it has events
    $eventCounts = API::Event()->get([
        'source' => EVENT_SOURCE_TRIGGERS,
        'object' => EVENT_OBJECT_TRIGGER,
        'countOutput' => true,
        'groupCount' => true,
        'objectids' => $triggerIds,
        'filter' => [
```

```
        'acknowledged' => 0,
        'value' => TRIGGER_VALUE_TRUE
    ]
});
foreach ($eventCounts as $eventCount) {
    $triggers[$eventCount['objectid']]['hasEvents'] = true;
    $triggers[$eventCount['objectid']]['event_count'] =
$eventCount['rowcount'];
}

// gather ids of triggers which don't have unack. events
$triggerIdsWithoutUnackEvents = [];
foreach ($triggers as $tnum => $trigger) {
    if (!isset($trigger['hasEvents'])) {
        $triggerIdsWithoutUnackEvents[] = $trigger['triggerid'];
    }
    if (!isset($trigger['event_count'])) {
        $triggers[$tnum]['event_count'] = 0;
    }
}
if (!empty($triggerIdsWithoutUnackEvents)) {
    // for triggers without unack. events we try to select any event
    $allEventCounts = API::Event()->get([
        'source' => EVENT_SOURCE_TRIGGERS,
        'object' => EVENT_OBJECT_TRIGGER,
        'countOutput' => true,
        'groupCount' => true,
        'objectids' => $triggerIdsWithoutUnackEvents
    ]);
}
```

```
$allEventCounts = zbx_toHash($allEventCounts, 'objectid');

foreach ($triggers as $tnum => $trigger) {
    if (!isset($trigger['hasEvents'])) {
        $triggers[$tnum]['hasEvents'] =
isset($allEventCounts[$trigger['triggerid']]);
    }
}

}

}

if ($showEvents == EVENTS_OPTION_ALL || $showEvents ==
EVENTS_OPTION_NOT_ACK) {
    foreach ($triggers as &$amp;trigger) {
        $trigger['events'] = [];
    }
    unset($trigger);

    $options = [
        'output' => ['eventid', 'objectid', 'clock', 'value'],
        'source' => EVENT_SOURCE_TRIGGERS,
        'object' => EVENT_OBJECT_TRIGGER,
        'objectids' => zbx_objectValues($triggers, 'triggerid'),
        'time_from' => time() - $config['event_expire'] * SEC_PER_DAY,
        'time_till' => time(),
        'sortfield' => ['clock', 'eventid'],
        'sortorder' => ZBX_SORT_DOWN
    ];
```

```
if ($config['event_ack_enable']) {
    $options['select_acknowledges'] = API_OUTPUT_COUNT;
    $options['output'][] = 'acknowledged';
}
$events = API::Event()->get($options);

foreach ($events as $event) {
    $triggers[$event['objectid']]['events'][] = $event;
}
}

// get host ids
$hostIds = [];
foreach ($triggers as $tnum => $trigger) {
    foreach ($trigger['hosts'] as $host) {
        $hostIds[$host['hostid']] = $host['hostid'];
    }
}

// get hosts
$hosts = API::Host()->get([
    'output' => ['hostid', 'status'],
    'hostids' => $hostIds,
    'preservekeys' => true,
    'selectGraphs' => API_OUTPUT_COUNT,
    'selectScreens' => API_OUTPUT_COUNT
]);

// get host scripts
```

```
$scriptsByHosts = API::Script()->getScriptsByHosts($hostIds);

// get trigger dependencies
$dbTriggerDependencies = DBselect(
    'SELECT triggerid_down,triggerid_up'.
    ' FROM trigger_depends'.
    ' WHERE '.dbConditionInt('triggerid_up', $triggerIds)
);
$triggerIdsDown = [];
while ($row = DBfetch($dbTriggerDependencies)) {
    $triggerIdsDown[$row['triggerid_up']] = intval($row['triggerid_down']);
}

$maintenanceids = [];

foreach ($triggers as $trigger) {
    foreach ($trigger['hosts'] as $host) {
        if ($host['maintenance_status'] ==
HOST_MAINTENANCE_STATUS_ON) {
            $maintenanceids[$host['maintenanceid']] = true;
        }
    }
}

if ($maintenanceids) {
    $maintenances = API::Maintenance()->get([
        'maintenanceids' => array_keys($maintenanceids),
        'output' => ['name', 'description'],
        'preservekeys' => true
    ]
);
}
```

```

    });
}

foreach ($triggers as $trigger) {
    /*
        * At this point "all" or one group is selected. And same goes for hosts.
        It is safe to pass 'groupid' and 'hostid'
        * to trigger menu pop-up, so it properly redirects to Events page. Mind
        that 'DDRemember' option will be ignored.
    */
    $trigger['groupid'] = $pageFilter->groupid;
    $trigger['hostid'] = $pageFilter->hostid;

    $description = [];

    if (!empty($trigger['dependencies'])) {
        $dependenciesTable = (new CTable())
            ->setAttribute('style', 'min-width:
'.ZBX_TEXTAREA_STANDARD_WIDTH.'px;')
            ->addRow(_('Depends on').':');

        foreach ($trigger['dependencies'] as $dependency) {
            $dependenciesTable->addRow(" -
'.CMacrosResolverHelper::resolveTriggerNameById($dependency['triggerid']))
;
        }

        $description[] = (new CSpan())
            ->addClass(ZBX_STYLE_ICON_DEPEND_DOWN)

```

```

        ->addClass(ZBX_STYLE_CURSOR_POINTER)
        ->setHint($dependenciesTable);
    }

    $dependency = false;
    $dependenciesTable = (new CTable())
        ->setAttribute('style', 'min-width:
'.ZBX_TEXTAREA_STANDARD_WIDTH.'px;')
        ->addRow(_('Dependent').':');
    if (array_key_exists($trigger['triggerid'], $triggerIdsDown) &&
$triggerIdsDown[$trigger['triggerid']]) {
        $depTriggers =
CMacrosResolverHelper::resolveTriggerNameByIds($triggerIdsDown[$trigger[
'triggerid']]);

        foreach ($depTriggers as $depTrigger) {
            $dependenciesTable->addRow(SPACE.'-
'.SPACE.$depTrigger['description']);
            $dependency = true;
        }
    }

    if ($dependency) {
        $description[] = (new CSpan())
            ->addClass(ZBX_STYLE_ICON_DEPEND_UP)
            ->addClass(ZBX_STYLE_CURSOR_POINTER)
            ->setHint($dependenciesTable);
    }
    unset($img, $dependenciesTable, $dependency);

```

```
$description[] = (new CSpan($trigger['description']))
    ->addClass(ZBX_STYLE_LINK_ACTION)
    ->setMenuPopup(CMenuPopupHelper::getTrigger($trigger));

if ($showDetails) {
    $description[] = BR();
    $description[] = $trigger['expression'];
}

// host js menu
$hostList = [];
foreach ($trigger['hosts'] as $host) {
    // fetch scripts for the host js menu
    $scripts = [];
    if (isset($scriptsByHosts[$host['hostid']])) {
        foreach ($scriptsByHosts[$host['hostid']] as $script) {
            $scripts[] = $script;
        }
    }

    $host_name = (new CSpan($host['name']))
        ->addClass(ZBX_STYLE_LINK_ACTION)
        -
    >setMenuPopup(CMenuPopupHelper::getHost($hosts[$host['hostid']],
    $scripts));

    // add maintenance icon with hint if host is in maintenance
    if ($host['maintenance_status'] ==
```



```
HOST_MAINTENANCE_STATUS_ON) {
    $maintenance_icon = (new CSpan())
        ->addClass(ZBX_STYLE_ICON_MAINT)
        ->addClass(ZBX_STYLE_CURSOR_POINTER);

    if (array_key_exists($host['maintenanceid'],
$maintenances)) {
        $maintenance =
$maintenances[$host['maintenanceid']];

        $hint = $maintenance['name'].
['.($host['maintenance_type']
        ? _('Maintenance without data collection')
        : _('Maintenance with data collection')).'];

        if ($maintenance['description']) {
            $hint .= "\n".$maintenance['description'];
        }

        $maintenance_icon->setHint($hint);
    }

    $host_name = (new CSpan([$host_name,
$maintenance_icon]))->addClass(ZBX_STYLE_REL_CONTAINER);
}

$hostList[] = $host_name;
$hostList[] = ', ';
}
```

```
array_pop($hostList);

// status
$statusSpan = new CSpan(trigger_value2str($trigger['value']));

// add colors and blinking to span depending on configuration and
trigger parameters
addTriggerValueStyle(
    $statusSpan,
    $trigger['value'],
    $trigger['lastchange'],
    $config['event_ack_enable'] ? ($trigger['event_count'] == 0) :
false
);

// open or close
// acknowledge
if ($config['event_ack_enable']) {
    if ($trigger['hasEvents']) {
        $ack_checkbox = new
CCheckBox('eventids['.$trigger['lastEvent']['eventid'].']',
            $trigger['lastEvent']['eventid']
        );
        if ($trigger['event_count']) {
            $ackColumn = [
                (new CLink_('No'),

'zabbix.php?action=acknowledge.edit'.
```

```

'&acknowledge_type='.ZBX_ACKNOWLEDGE_PROBLEM.

'&eventids[]='.$trigger['lastEvent']['eventid'].
                                '&backurl='.$page['file']
                                ))
                                ->addClass(ZBX_STYLE_LINK_ALT)
                                ->addClass(ZBX_STYLE_RED),

CViewHelper::showNum($trigger['event_count'])
                                ];
                                }
                                else {
                                    $ackColumn = (new CLink(_('Yes'),
                                        'zabbix.php?action=acknowledge.edit'.

'&acknowledge_type='.ZBX_ACKNOWLEDGE_PROBLEM.

'&eventids[]='.$trigger['lastEvent']['eventid'].
                                '&backurl='.$page['file']
                                ))
                                ->addClass(ZBX_STYLE_LINK_ALT)
                                ->addClass(ZBX_STYLE_GREEN);
                                }
                                }
                                else {
                                    $ack_checkbox = "";
                                    $ackColumn = (new CCol(_('No events')))-
>addClass(ZBX_STYLE_GREY);
                                }

```

```

    }
    else {
        $ack_checkbox = null;
        $ackColumn = null;
    }

    if ($showEvents == EVENTS_OPTION_ALL || $showEvents ==
EVENTS_OPTION_NOT_ACK) {
        $openOrCloseDiv = $trigger['events']
            ? (new CDiv())
                ->addClass(ZBX_STYLE_TREEVIEW)
                ->setAttribute('data-switcherid', $trigger['triggerid'])
                ->addItem((new CSpan())-
>addClass(ZBX_STYLE_ARROW_RIGHT))
                : "";
    }
    else {
        $openOrCloseDiv = null;
    }

    // comments
    if (isset($triggerEditable[$trigger['triggerid']])) {
        #$_comments = new CLink(zbx_empty($trigger['comments']) ?
_('Add') : _('Show'), 'tr_comments.php?triggerid='.$trigger['triggerid']);
        $_comments = new CLink(_('Clear
Trap'),'trap_send_clear.php?triggerid='.$trigger['triggerid'].'&backurl='.$page['fi
le']);
    }
    #$_ack_checkbox,

```

```

$triggerTable->addRow([
    $openOrCloseDiv,
    getSeverityCell($trigger['priority'], $config, null, !$trigger['value']),
    $statusSpan,
    ($trigger['state'] == TRIGGER_STATE_UNKNOWN) ?
makeUnknownIcon($trigger['error']) : "",
    ($trigger['lastchange'] == 0)
        ? _('Never')
        : new
CLink(zbx_date2str(DATE_TIME_FORMAT_SECONDS,
$trigger['lastchange']),

    'events.php?filter_set=1&triggerid='.$trigger['triggerid'].'&source='.EVENT_SOURCE_TRIGGERS.

        '&stime='.date(TIMESTAMP_FORMAT,
$trigger['lastchange']).&period='.ZBX_PERIOD_DEFAULT
    ),
    ($trigger['lastchange'] == 0) ? " :
zbx_date2age($trigger['lastchange']),
    ($showEvents == EVENTS_OPTION_ALL || $showEvents ==
EVENTS_OPTION_NOT_ACK) ? " : null,
    $ackColumn,
    $hostList,
    $description,
    $comments
]);

if ($showEvents == EVENTS_OPTION_ALL || $showEvents ==
EVENTS_OPTION_NOT_ACK) {

```

```
$next_event_clock = time();

foreach (array_slice($trigger['events'], 0,
$config['event_show_max']) as $enum => $event) {
#           if ($showEvents == EVENTS_OPTION_NOT_ACK) {
#           if ($event['acknowledged'] || $event['value'] !=
TRIGGER_VALUE_TRUE) {
#               continue;
#           }
#       }

        $eventStatusSpan = new
Cspan(trigger_value2str($event['value']));

        // add colors and blinking to span depending on
configuration and trigger parameters
        addTriggerValueStyle($eventStatusSpan, $event['value'],
$event['clock'],
            $config['event_ack_enable'] &&
$event['acknowledged']);

        $clock = new
CLink(zbx_date2str(DATE_TIME_FORMAT_SECONDS, $event['clock']),

'tr_events.php?triggerid='.$trigger['triggerid'].'&eventid='.$event['eventid
']);

        if ($enum != 0) {
            $next_event_clock = $trigger['events'][$enum -
```

```

1]['clock'];

        }

        $triggerTable->addRow(
            (new CRow([
                (new CCol())-
>setColSpan($config['event_ack_enable'] ? 3 : 2),
                (new CCol($eventStatusSpan))-
>setColSpan(2),

                $clock,
                zbx_date2age($event['clock']),
                zbx_date2age($next_event_clock,
$event['clock']),

                $config['event_ack_enable'] ?
getEventAckState($event, $page['file']) : null,
                (new CCol())->setColSpan(3)
            ]))
            ->setAttribute('data-parentid',
$trigger['triggerid'])
            ->addStyle('display: none;')
        );
    }
}

/*
 * Go buttons
 */
$footer = null;

```

```

$triggerForm->addItem([$triggerTable, $paging, $footer]);
$triggerWidget->addItem($triggerForm)->show();

zbx_add_post_js('jqBlink.blink();');
zbx_add_post_js('var switcher = new CSwitcher(\".$switcherName.\");');

require_once dirname(__FILE__).'/include/page_footer.php';
?>

```

2- Criar o arquivo /usr/share/zabbix/trap_send_clear.php com o conteúdo abaixo:

```

<?php

/*
** Zabbix
** Copyright (C) 2001-2017 Zabbix SIA
**
** This program is free software; you can redistribute it and/or modify
** it under the terms of the GNU General Public License as published by
** the Free Software Foundation; either version 2 of the License, or
** (at your option) any later version.
**
** This program is distributed in the hope that it will be useful,
** but WITHOUT ANY WARRANTY; without even the implied warranty of
** MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See
the
** GNU General Public License for more details.

```



```

**
** You should have received a copy of the GNU General Public License
** along with this program; if not, write to the Free Software
** Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301,
USA.
**/

////////////////////////////////////
// Script Name: trap_send_clear.php
// Project: NOC Porto Seguro
// Porto Seguro - Ageri
// Resp: Paulo Tetsuo Hoashi      Date: 18/05/17  Desc: Emission
// Function: Envia a mensagem de clear manual para o host.
////////////////////////////////////

require_once dirname(__FILE__).'/include/config.inc.php';
require_once dirname(__FILE__).'/include/triggers.inc.php';
require_once dirname(__FILE__).'/include/forms.inc.php';

$page['title'] = _('Trap send clear');
$page['file'] = 'trap_send_clear.php';
$page['scripts'] = ['class.cswitcher.js'];
$page['type'] = detect_page_type(PAGE_TYPE_HTML);

require_once dirname(__FILE__).'/include/page_header.php';
require_once dirname(__FILE__).'/conf/config_db.php';
    $user_alias=CWebUser::$data['alias'];
    $conn = @mysqli_connect($hostLocal, $userLocal, $passLocal,
$dbLocal);
    $triggerid=$_GET['triggerid'];

```

```
$backurl=$_GET['backurl'];  
#    echo "triggerid: ".$_GET['triggerid']." alias ".$user_alias."<br>";  
  
#    $sql_1 = "SELECT hosts.host, hosts.proxy_hostid,proxyname,  
items.key_ FROM hosts, items, functions,( SELECT descriptions as  
proxymane FROM hostsas proxy WHERE proxy.hostid=hosts.proxy ) as  
proxy WHERE hosts.hostid=items.hostid AND items.itemid=functions.itemid  
AND functions.triggerid=".$triggerid."";  
#    $sql_1="SELECT hosts.host, hosts.proxy_hostid,proxy.description,  
items.key_ FROM hosts, items, functions,( SELECT hostid,description FROM  
hosts GROUP BY hostid ) as proxy WHERE proxy.hostid =hosts.proxy_hostid  
AND hosts.hostid=items.hostid AND items.itemid=functions.itemid AND  
functions.triggerid=".$triggerid."";  
    $sql_1="SELECT distinct hosts.host,  
hosts.proxy_hostid,proxy.description, items.key_, interface.ip, interface.dns,  
interface.useip FROM hosts, items, functions,( SELECT hostid,description  
FROM hosts GROUP BY hostid ) as proxy, interface WHERE proxy.hostid  
=hosts.proxy_hostid AND hosts.hostid=items.hostid AND  
items.itemid=functions.itemid AND hosts.hostid=interface.hostid AND  
interface.type =2 AND functions.triggerid=".$triggerid."";  
#    echo $sql_1."<br>";  
    $query_1 = mysqli_query($conn, $sql_1);  
    $num_lines_1 = mysqli_num_rows($query_1);  
#    echo "Lines: ".$num_lines_1."<br>";  
    if ($num_lines_1>0){  
        while ($list_1 = mysqli_fetch_array($query_1)) {  
            $host=$list_1['host'];  
            $proxyid=$list_1['proxy_hostid'];  
            $proxy=$list_1['description'];
```

```

        $key=$list_1['key_'];
        $ip=$list_1['ip'];
        $useip=$list_1['useip'];
        $dns=$list_1['dns'];
        #echo "HOST ".$host." proxy: ".$proxy." key: ".$key." ip:
        ".$ip."<br>";
    }
    $chave= str_replace('snmptrap["',"$key);
    $chave= substr_replace ($chave,'"",-2);
    $pos1 = stripos($chave,['');
    $pos2 = stripos($chave,']');
    #echo "chave: $chave <br>";
    if (($pos1>0) && ($pos2>0)){
        $prefix_key = substr ($chave,0,$pos1);
        $sufix_key = substr ($chave,$pos2+1);
        $array=substr ($chave,$pos1,2*strlen($chave) - $pos1 -
        $pos2);

        $array=trim($array,['');
        $array=trim($array,']');
        #echo "pos1: $pos1 | pos2: $pos2 <br>";
        #echo "array: $array<br>";
        $value=explode('|',$array);
        #echo "prefix: $prefix_key<br>";
        #echo "sufix: $sufix_key<br>";
        $chave = $prefix_key.$value[0].$sufix_key;
    }else {
        $pos1 = stripos($chave,['');
        $pos2 = stripos($chave,']');
        #echo "chave: $chave <br>";
    }

```

```

        if (($pos1>0) && ($pos2>0)){
            $prefix_key = substr ($chave,0,$pos1);
            $sufix_key = substr ($chave,$pos2+1);
            $array=substr ($chave,$pos1, 2*strlen($chave) - $pos1 -
$pos2);

            $array=trim($array,'(');
            $array=trim($array,')');
            #echo "pos1: $pos1 | pos2: $pos2 <br>";
            #echo "array: $array<br>";
            $value=explode('|',$array);
            #echo "prefix: $prefix_key<br>";
            #echo "sufix: $sufix_key<br>";
            $chave = $prefix_key.$value[0].$sufix_key;
            #echo "Message de CLEAR MANUAL : $chave
<br>";

        }
    }

    $chave= str_replace("*","", $chave);
    $chave= str_replace("\\","", $chave);
    $data=date("H:i:s Y/m/d");
    # $chave= $data." - ZBXTRAP $ip ".$chave." Clear Manual
".$user_alias;
    if ( $useip == 1 ){
        $chave= $data." - ZBXTRAP $ip ".$chave." Clear Manual
Usuario: ".$user_alias;
    }else{
        $chave= $data." - ZBXTRAP $dns ".$chave." Clear
Manual Usuario: ".$user_alias;
    }

```

```

        echo "Message de CLEAR MANUAL : $chave <br>";
        #$resultado=shell_exec("/usr/bin/zabbix_get -s $proxy -k
system.run['echo \"\$chave\" >> /var/log/zabbix/teste.log'] ");
        #echo "/usr/bin/zabbix_get -s $proxy -k 'system.run[\"echo
$chave >> /var/log/zabbix/snmptrap.log\\\"\\\"<br>";
        ##$resultado=shell_exec("/usr/bin/zabbix_get -s $proxy -k
'system.run[\"echo $chave >> /var/log/zabbix/snmptrap.log\\\"'] ");
        $resultado=shell_exec("/usr/bin/zabbix_get -s $proxy -k
\\\"system.run[\\\"echo \\\"\\\"\\\".\\$chave.\\\"\\\"\\\" >>
/var/log/zabbix/snmptrap.log\\\"\\\"\\\"'] \" ");
        #echo $resultado >> "/var/log/zabbix/zabbix_agentd.log";

        $myfile = fopen("/var/log/zabbix/zabbix_agentd_clear.log", "a")
or die("Unable to open file!");
        fwrite($myfile, $proxy." ".$chave."\\n");
        fclose($myfile);
        #echo "/usr/bin/zabbix_get -s $proxy -k \\\"system.run[\\\"echo
\\\"\\\"\\\".str_replace(\"\\\", \"\", $chave).\\\"\\\"\\\" >> /var/log/zabbix/snmptrap.log\\\"\\\"\\\"] \" ";
        #$resultado=shell_exec("/usr/bin/zabbix_get -s $proxy -k
'system.run[\"echo $chave \\\"'] ");
        #echo $resultado."<br>\\n";
        #$resultado=`/usr/bin/zabbix_get -s $proxy -k 'system.run[\"echo
$chave >> /var/log/zabbix/snmptrap.log\\\"']";
        #$resultado=shell_exec("ls -al /tmp ");
        #echo "proxy: ".$proxy."<br>";
        #echo "Resultado ".$resultado."<br>";
        echo "SEND CLEAR TO HOST: $host";
    }
#    close($conn);

```

```
#      echo "<a href=\".$backurl.\">\".$backurl.</a><BR>";
      require_once dirname(__FILE__).'/include/page_footer.php';
?>
```

3- modificar a barra de menu para aparecer a opção “Clear_traps”.

Editar o arquivo /usr/share/zabbix/include/menu.inc.php, e adicionar o bloco em destaque

```
...
    'config' => [
        'label' => _('Configuration'),
        'user_type' => USER_TYPE_ZABBIX_ADMIN,
        'default_page_id' => 0,
        'pages' => [
...
            'url' => 'hosts.php',
            'label' => _('Hosts'),
...
            'popup_httpstep.php'
        ]
    ],
    [
        'url' => 'tr_trap_clear.php',
        'label' => _('Clear_Traps'),
        'sub_pages' => ['trap_send_clear.php']
    ],
/*
    [
        'url' => 'maintenance.php',
        'label' => _('Maintenance')
    ],
*/
    [
```

```
        'url' => 'actionconf.php',  
        'label' => _('Actions')  
    ],  
    ...
```

Maintenance

Para que os usuários sem o perfil de administrador possam criar janelas de manutenção, foi necessário customizar o front-end do Zabbix.

1- editar o arquivo `/usr/share/zabbix/include/menu.inc.php` e comentar o bloco `cm Inventory` e adicionar `maintenance`.

```

        ],
        [
            'url' => 'jsrpc.php'
        ]
    ]
],
/*      'cm' => [
            'label' => _('Inventory'),
            'user_type' => USER_TYPE_ZABBIX_USER,
            'default_page_id' => 0,
            'pages' => [
                [
                    'url' => 'hostinventoriesoverview.php',
                    'label' => _('Overview')
                ],
                [
                    'url' => 'hostinventories.php',
                    'label' => _('Hosts')
                ]
            ]
        ],
    ],
*/

```



```

'maintenance' => [
    'label' => _('Maintenance'),
    'user_type' => USER_TYPE_ZABBIX_USER,
    'default_page_id' => 0,
    'pages' => [
        [
            'url' => 'maintenance.php',
            'label' => _('Maintenance')
        ]
    ]
],
'reports' => [
    'label' => _('Reports'),
    'user_type' => USER_TYPE_ZABBIX_USER,
    'default_page_id' => 0,
    'pages' => [
        [

```

2- comentar o Maintenance do menu de configuração também no arquivo
/usr/share/zabbix/include/menu.inc.php.

```

[
    'url' => 'hosts.php',
    'label' => _('Hosts'),
    'sub_pages' => [
        'items.php',
        'triggers.php',

```

```
        'graphs.php',
        'applications.php',
        'tr_logform.php',
        'tr_testexpr.php',
        'popup_trexp.php',
        'host_discovery.php',
        'disc_prototypes.php',
        'trigger_prototypes.php',
        'host_prototypes.php',
        'httpconf.php',
        'popup_httpstep.php'
    ]
],

[
    'url' => 'tr_trap_clear.php',
    'label' => _('Clear_Traps'),
    'sub_pages' => ['trap_send_clear.php']
],
/*      [
        'url' => 'maintenance.php',
        'label' => _('Maintenance')
    ],
*/

[
    'url' => 'actionconf.php',
    'label' => _('Actions')
],
[
```

```

        'url' => 'discoveryconf.php',
        'label' => _('Discovery')
    ],

```

3- alterar o arquivo /usr/share/zabbix/js/main.js conforme a linha em destaque.

```

/*
 * Main menu
 */
var MMenu = {
    menus:          {'view': 0, 'maintenance': 0, 'reports': 0, 'config': 0,
'admin': 0},
    def_label:      null,
    sub_active:     false,
    timeout_reset:  null,
    timeout_change: null,

    mouseOver: function(show_label) {
        clearTimeout(this.timeout_reset);
        this.timeout_change = setTimeout('MMenu.showSubMenu("'" +
show_label + "'", 10);
        PageRefresh.restart();
    },

```

4- alterar o function create no arquivo

include/classes/api/services/CMaintenance.php da seguinte forma:

```

public function create(array $maintenances) {
    $maintenances = zbx_toArray($maintenances);

```

```

if (self::$userData['type'] == 0) {
    self::exception(ZBX_API_ERROR_PERMISSIONS, _('No
permissions to referred object or it does not exist!'));
}

$hostids = [];
$groupids = [];

```

5- alterar o function update no arquivo

include/classes/api/services/CMaintenance.php da seguinte forma:

```

public function update(array $maintenances) {
    $maintenances = zbx_toArray($maintenances);
    $maintenanceids = zbx_objectValues($maintenances,
'maintenanceid');

    // validate maintenance permissions
if (self::$userData['type'] == 0) {
        self::exception(ZBX_API_ERROR_PERMISSIONS, _('No
permissions to referred object or it does not exist!'));
    }
}

```

6- alterar o function delete no arquivo

include/classes/api/services/CMaintenance.php da seguinte forma:

```

public function delete(array $maintenanceids) {
if (self::$userData['type'] == 0) {
    self::exception(ZBX_API_ERROR_PERMISSIONS, _('You do
not have permission to perform this operation.'));
}
}

```

```

}

$options = [
    'output' => ['maintenanceid', 'name'],
    'maintenanceids' => $maintenanceids,
    'editable' => true,
    'preservekeys' => true
];

```

Após esses ajustes o menu maintenance aparecerá no menu principal do zabbix para todos os usuários.

*



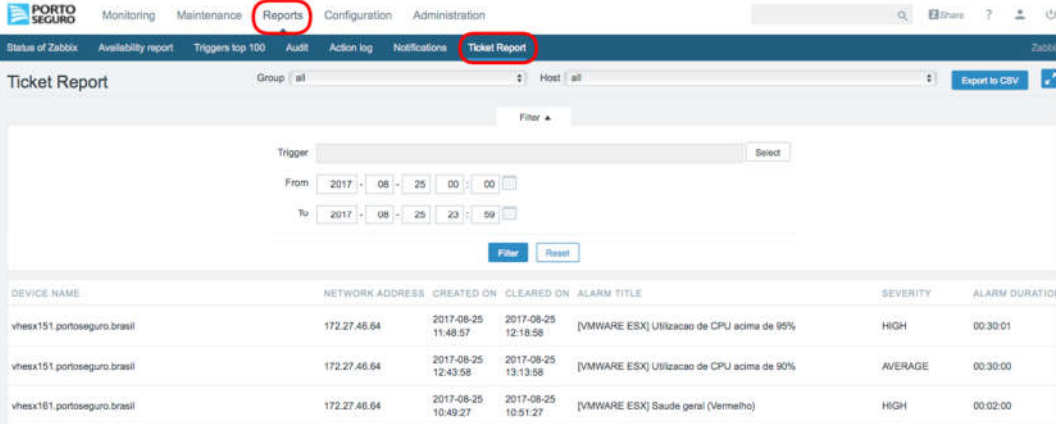
* Para colocar os hosts em manutenção o usuário precisa de permissão de leitura e escrita nos hosts.

Ticket Report

Este módulo refere-se ao acompanhamento da abertura de ticket, bem como horário do incidente e o retorno a normalizada, até a data da geração do relatório. Este relatório estará acessível apenas para os usuários que estejam configurados como User Type Zabbix Admin ou Zabbix Super Admin.

Acesso do usuário

O acesso do usuário é feita a partir do caminho: Reports -> Ticket Report.



DEVICE NAME	NETWORK ADDRESS	CREATED ON	CLEARED ON	ALARM TITLE	SEVERITY	ALARM DURATION
vhess151.portoseguro.brasil	172.27.46.64	2017-08-25 11:48:57	2017-08-25 12:18:58	[VMWARE ESX] Utilizacao de CPU acima de 95%	HIGH	00:30:01
vhess151.portoseguro.brasil	172.27.46.64	2017-08-25 12:43:58	2017-08-25 13:13:58	[VMWARE ESX] Utilizacao de CPU acima de 90%	AVERAGE	00:30:00
vhess161.portoseguro.brasil	172.27.46.64	2017-08-25 10:49:27	2017-08-25 10:51:27	[VMWARE ESX] Saude geral (Vermelho)	HIGH	00:02:00

Filtros

O relatório de tickets possuem 4 tipos de de filtros:

Grupos

Para filtrar por grupo, basta selecionar o grupo em “Group”

The top screenshot shows the Zabbix Ticket Report interface. The 'Group' dropdown menu is set to 'all'. The table below shows the following data:

DEVICE NAME	NETWORK ADDRESS	CREATED ON	CLEARED ON	ALARM TITLE	SEVERITY	ALARM DURATION
vhess151.portoseguro.brasil	172.27.46.64	2017-08-25 11:48:57	2017-08-25 12:18:58	[VMWARE ESX] Utilizacao de CPU acima de 95%	HIGH	00:30:01
vhess151.portoseguro.brasil	172.27.46.64	2017-08-25 12:43:58	2017-08-25 13:13:58	[VMWARE ESX] Utilizacao de CPU acima de 90%	AVERAGE	00:30:00
vhess161.portoseguro.brasil	172.27.46.64	2017-08-25 10:49:27	2017-08-25 10:51:27	[VMWARE ESX] Saude geral (Vermelho)	HIGH	00:02:00

The bottom screenshot shows the same interface, but with the 'Host' dropdown menu open, displaying a list of hosts including: all, 3Com, 3Com-Paralelos, ALL_Servers, Aruba, Avaya, AWAYA_VOZ, Brocade, Capacity_Linux, Capacity_Unix, Capacity_Windows, CheckPoint, Cisco, Cisco-8xx-18xx-19xx-26xx-29xx-72xx-Paralelos, Cisco_CT-SW-AG-NCL-CORE-SPN-DCI_Nexus, Cisco_DSXX_4500, Cisco_DSXX_Nexus, Cisco_RouterRIPS, Cisco_RouterSUC-PTO, Cisco_SV-SF-PR_2960, Cisco_SV-SF-PR_Nexus, and Cisco_SW0X-2960-3k.

Assim que o grupo for selecionado o filtro será aplicado automaticamente.

Hosts

Para filtrar por hosts, basta selecionar o host em “Host”

Ticket Report

Group: all Host: all

Filter

Trigger: Select

From: 2017-08-25 00:00 To: 2017-08-25 23:59

Filter Reset

DEVICE NAME	NETWORK ADDRESS	CREATED ON	CLEARED ON	ALARM TITLE	SEVERITY	ALARM DURATION
vhess151.portoseguro.brasil	172.27.46.64	2017-08-25 11:48:57	2017-08-25 12:18:58	[VMWARE ESX] Utilizacao de CPU acima de 95%	HIGH	00:30:01
vhess151.portoseguro.brasil	172.27.46.64	2017-08-25 12:43:58	2017-08-25 13:13:58	[VMWARE ESX] Utilizacao de CPU acima de 90%	AVERAGE	00:30:00
vhess161.portoseguro.brasil	172.27.46.64	2017-08-25 10:49:27	2017-08-25 10:51:27	[VMWARE ESX] Saude geral (Vermelho)	HIGH	00:02:00

Ticket Report

Group: all Host: all

Filter

Trigger: Select

From: 2017-09-03 00:00 To: 2017-09-04 00:00

Filter Reset

DEVICE NAME	NETWORK ADDRESS	CREATED ON	CLEARED ON	ALARM TITLE	SEVERITY	ALARM DURATION
vhess151.portoseguro.brasil	172.27.46.64	2017-08-25 11:48:57	2017-08-25 12:18:58	[VMWARE ESX] Utilizacao de CPU acima de 95%	HIGH	00:30:01
vhess151.portoseguro.brasil	172.27.46.64	2017-08-25 12:43:58	2017-08-25 13:13:58	[VMWARE ESX] Utilizacao de CPU acima de 90%	AVERAGE	00:30:00
vhess161.portoseguro.brasil	172.27.46.64	2017-08-25 10:49:27	2017-08-25 10:51:27	[VMWARE ESX] Saude geral (Vermelho)	HIGH	00:02:00

Assim que o host for selecionado o filtro será aplicado automaticamente.

Triggers

Para filtrar por Triggers, basta selecionar a trigger em “Trigger”

Ticket Report

Group: all Host: all

Filter

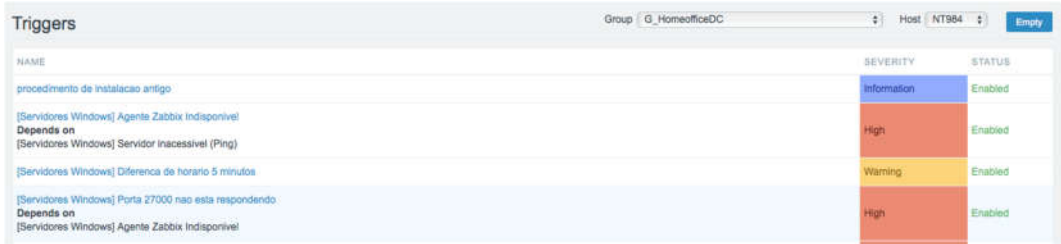
Trigger: Select

From: 2017-08-25 00:00 To: 2017-08-25 23:59

Filter Reset

DEVICE NAME	NETWORK ADDRESS	CREATED ON	CLEARED ON	ALARM TITLE	SEVERITY	ALARM DURATION
vhess151.portoseguro.brasil	172.27.46.64	2017-08-25 11:48:57	2017-08-25 12:18:58	[VMWARE ESX] Utilizacao de CPU acima de 95%	HIGH	00:30:01
vhess151.portoseguro.brasil	172.27.46.64	2017-08-25 12:43:58	2017-08-25 13:13:58	[VMWARE ESX] Utilizacao de CPU acima de 90%	AVERAGE	00:30:00
vhess161.portoseguro.brasil	172.27.46.64	2017-08-25 10:49:27	2017-08-25 10:51:27	[VMWARE ESX] Saude geral (Vermelho)	HIGH	00:02:00

Abrirá uma janela com as triggers



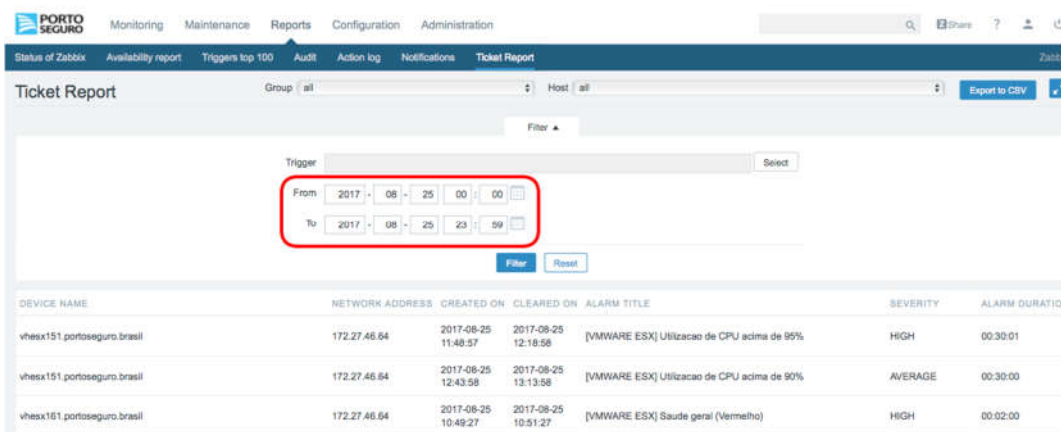
NAME	SEVERITY	STATUS
procedimento de instalacao antigo	Information	Enabled
[Servidores Windows] Agente Zabbix indisponivel Depends on [Servidores Windows] Servidor inacessivel (Ping)	High	Enabled
[Servidores Windows] Diferença de horario 5 minutos	Warning	Enabled
[Servidores Windows] Porta 27000 nao esta respondendo Depends on [Servidores Windows] Agente Zabbix indisponivel	High	Enabled

Clicar em Filter para realizar a busca.



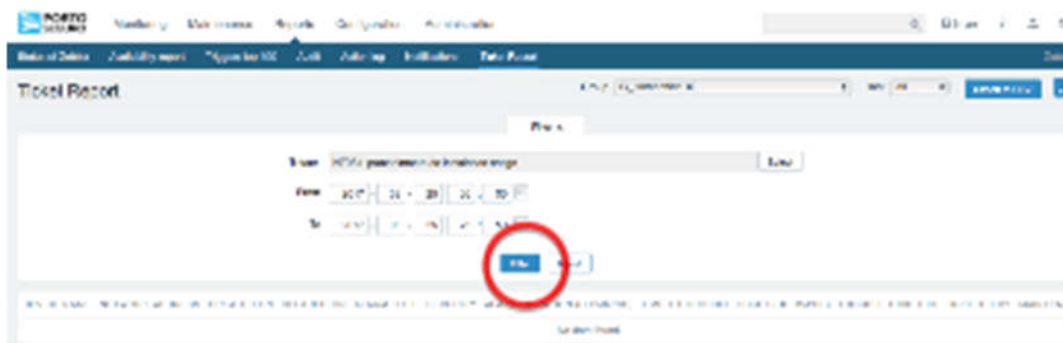
Período

Para filtrar por período, basta selecionar a data de inicio (From) e fim (To) da pesquisa.



DEVICE NAME	NETWORK ADDRESS	CREATED ON	CLEARED ON	ALARM TITLE	SEVERITY	ALARM DURATION
vhess151.portoseguro.brasil	172.27.46.64	2017-08-25 11:48:57	2017-08-25 12:18:58	[VMWARE ESX] Utilizacao de CPU acima de 95%	HIGH	00:30:01
vhess151.portoseguro.brasil	172.27.46.64	2017-08-25 12:43:58	2017-08-25 13:13:58	[VMWARE ESX] Utilizacao de CPU acima de 90%	AVERAGE	00:30:00
vhess161.portoseguro.brasil	172.27.46.64	2017-08-25 10:49:27	2017-08-25 10:51:27	[VMWARE ESX] Saude geral (Vermetho)	HIGH	00:02:00

Clicar em Filter para realizar a busca.



Funcionamento

O front-end report_Ticket.php, coleta e consolida os dados da tabela events_tickets, da base de dados Zabbix_report.

Essa tabela é populada através dos scripts, executados na cron, que realiza a inserção e atualização dos eventos Zabbix e tickets do SDM.

Configuração dos agendamentos:

```
#Relatorio zabbix
*/1 * * * * root /usr/share/zabbix/report/reportPortoAcompanhamentoTicketsDaily_Insert.sh
*/1 * * * * root /usr/share/zabbix/report/reportPortoAcompanhamentoTicketsDaily_UpdateAck.sh
*/1 * * * * root /usr/share/zabbix/report/reportPortoAcompanhamentoTicketsDaily_UpdateEvent.sh
00 1 * * * root /usr/share/zabbix/report/reportPortoAcompanhamentoTicketsDaily_DeleteTicket.sh
```

reportPortoAcompanhamentoTicketsDaily_Insert.sh

O script reportPortoAcompanhamentoTicketsDaily_Insert.php é responsável por coletar os novos incidentes, guarda na tabela events_tickets, na base de dados zabbix_report. Caso exista um processo deste script em execução o script da cron aborta a execução atual para não ocorrer a execução em paralelo do mesmo.

reportPortoAcompanhamentoTicketsDaily_UpdateAck.sh

O script reportPortoAcompanhamentoTicketsDaily_UpdateAck.php é responsável por preencher o ack, se os dados de ack não estiverem sido preenchidos. Caso exista um processo deste script em execução o script da cron aborta a execução atual para não ocorrer a execução em paralelo do mesmo.

reportPortoAcompanhamentoTicketsDaily_UpdateEvent.sh

O script reportPortoAcompanhamentoTicketsDaily_UpdateEvent.php é responsável por preencher o evento de normalização, se o evento de normalização não estiverem sido preenchidos. Caso exista um processo deste script em execução o script da cron aborta a execução atual para não ocorrer a execução em paralelo do mesmo.

reportPortoAcompanhamentoTicketsDaily_DeleteTicket.sh

O script reportPortoAcompanhamentoTicketsDaily_DeleteTicket.php é responsável por deletar os dados mais antigos. Default: 7meses. Caso exista um processo deste script em execução o script da cron aborta a execução atual para não ocorrer a execução em paralelo do mesmo.

Instalação do módulo

Os passos 1 e 2 devem ser executados no banco de dados, todos os demais serão executados no front-end.

1- Criar a base de dados zabbix_report

```
CREATE DATABASE `zabbix_report` /*!40100 DEFAULT CHARACTER SET  
utf8 */
```

2- Criar a tabela event_tickets

```
CREATE TABLE `events_tickets` (  
  `eventsTicketID` bigint(20) NOT NULL,  
  `eventid` bigint(20) NOT NULL,  
  `objectID` bigint(20) NOT NULL,  
  `trigger_description` varchar(255) DEFAULT NULL,  
  `clk_event_create` int(11) NOT NULL,  
  `clk_event_cleared` int(11) DEFAULT NULL,  
  `clk_ack` int(11) DEFAULT NULL,  
  `ticket` varchar(255) DEFAULT NULL,  
  `userid` int(11) DEFAULT NULL,  
  `clk_disable` int(11) DEFAULT NULL,  
  `maintenance` int(11) DEFAULT NULL,  
  `alias` varchar(100) DEFAULT NULL,  
  PRIMARY KEY (`eventsTicketID`,`eventid`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

3- Criar o arquivo /usr/share/zabbix/conf/config_db.php, com as configurações de acesso ao banco de dados.

```
<?php  
$hostLocal = "172.27.47.91";  
$userLocal = "zabbix_user";  
$passLocal = "mk7UPB70Burv";  
$portLocal = "3306";
```

```
$dbLocal = "zabbix_db";  
$dbLocalReport = "zabbix_report";  
?>
```

4- Criar a estrutura de diretórios do ticket report.

```
mkdir -p /usr/share/zabbix/report/includes
```

5- criar o arquivo /usr/share/zabbix/report/includes/phpfunctions.php com o conteúdo abaixo:

```
<?php  
  
function formataDataBD($strData) {  
  
    if ($strData != "" && $strData != "0000-00-00") {  
        $tokenData = split("/.", $strData);  
        $dataBD = $tokenData[2]."-".$tokenData[1]."-".$tokenData[0];  
    } else {  
        $dataBD = "";  
    }  
    return $dataBD;  
}  
  
function formataDataHTTP($strData){  
  
    if ($strData != "" && $strData != "0000-00-00") {
```

```

        $tokenData = split('[./-'],$strData);
        $dataHTTP = $tokenData[2]."/".$tokenData[1]."/".$tokenData[0];
    } else {
        $dataHTTP = "";
    }
    return $dataHTTP;
}

function GroupsSelect() {
    global $hostLocal, $userLocal, $passLocal, $dbLocal;

    $conn = @mysqli_connect($hostLocal, $userLocal, $passLocal,
$dbLocal);

    $sql_f = "SELECT * FROM groups WHERE internal='0' AND groupid !=
'1001000000000001' ORDER BY name";
    $query_f = mysqli_query($sql_f) or die("$langMsgConnectionFail(f2).
$langMsgTryAgain");
    while ($list_f = mysqli_fetch_array($query_f)) {
        $Dado .= "<option value=$list_f[groupid]>$list_f[name]</option> \n";
    }
    mysqli_close($conn);
    return $Dado;
}

function groupById($var_groupid) {
    global $host, $user, $pass, $db, $port;

```

```
$conn = @mysqli_connect($hostLocal, $userLocal, $passLocal,
$dbLocal);

$sql_f = "SELECT * FROM groups WHERE groupid='$var_groupid'";
$query_f = mysqli_query($sql_f) or die("$langMsgConnectionFail(f4).
$langMsgTryAgain");
$Dado = mysqli_fetch_array($query_f);
mysqli_close($conn);
return $Dado;
}

function hostByld($var_hostid) {
    global $host, $user, $pass, $db, $port;

    $conn = @mysqli_connect($hostLocal, $userLocal, $passLocal,
$dbLocal);

    $sql_f = "SELECT * FROM hosts WHERE hostid='$var_hostid'";
    $query_f = mysqli_query($sql_f) or die("$langMsgConnectionFail(f4).
$langMsgTryAgain");
    $Dado = mysqli_fetch_array($query_f);
    mysqli_close($conn);
    return $Dado;
}

function itemByld($var_itemid) {
    global $host, $user, $pass, $db, $port;
```

```

$conn = @mysqli_connect($hostLocal, $userLocal, $passLocal,
$dbLocal);

$sql_f = "SELECT * FROM items WHERE itemid='$var_itemid'";
$query_f = mysqli_query($sql_f) or die("$langMsgConnectionFail(f4).
$langMsgTryAgain");
$Dado = mysqli_fetch_array($query_f);
mysqli_close($conn);
return $Dado;
}

function Message($eventid) {
    global $host, $user, $pass, $db, $port;

    $sql_1 = "SELECT a.message, a.clock, u.name
        FROM acknowledges a, users u
        WHERE a.eventid = '$eventid'
        AND a.userid = u.userid ORDER BY clock ASC LIMIT 1";
    $query_1 = mysqli_query($sql_1);
    $list_1 = mysqli_fetch_array($query_1);
    return $list_1;
}

function getSeverityName1($severity){

    switch ($severity) {
        case 0: return "NOT_CLASSIFIED";
        case 1: return "INFORMATION";
        case 2: return "WARNING";
    }
}

```



```
        case 3: return "AVERAGE";
        case 4: return "HIGH";
        case 5: return "DISASTER";
        default: return 'Unknown';
    }
}

function age2date($original_time){
    $time = $original_time;
    $hours = ($time/3600)%3600;
    $time -= $hours*3600;
    $minutes = ($time/60)%60;
    $seconds = $time - $minutes*60;
    if ($seconds==0) $secondsStr = "00";
    else $secondsStr = ($seconds<10 ? "0" : "").$seconds;
    if ($minutes==0) $minutesStr = "00";
    else $minutesStr = ($minutes<10 ? "0" : "").$minutes;
    if ($hours==0) $hoursStr = "00";
    else $hoursStr = ($hours<10 ? "0" : "").$hours;
    $str = ($hoursStr.".".$minutesStr.".".$secondsStr);
    return $str;
}

function getColorAvailability($value){
    if ($value >= 99) $color_bg = "#32CD32"; // normal
    else if ($value > 97) $color_bg = "#FF3800"; // warn
    else $color_bg = "#E7471"; // critical
}
```

```
        return $color_bg;
    }

/* function: get_next_event
 * description: return next event by value
 * author: Aly Adapted by Lidia
 */
function NMX_get_next_event($row){
//      if((TRIGGER_MULT_EVENT_ENABLED == $row['type']) &&
(TRIGGER_VALUE_TRUE == $row['value'])){
    if(($row['type'] == 1) && ($row['value'] == 1)){
        $sql = "SELECT * FROM events
                WHERE objectid='$row[objectid]'
                AND clock > '$row[clock]'
                AND object = 0
                AND source = 0
                AND value = '$row[value]'
                ORDER BY object, objectid, clock limit 1" ;
    }
    else{
        $sql = "SELECT * FROM events
                WHERE objectid='$row[objectid]'
                AND clock > '$row[clock]'
                AND object = 0
                AND source = 0
                AND value != '$row[value]'
                ORDER BY object, objectid, clock limit 1";
    }
}
```

```
//      $query = mysqli_query($sql) or die("Fail");
//      $rez = mysqli_fetch_array($query);
//      return $rez;
//      return $sql;
}

/* function: get_next_event
 * description: return next event by value
 * author: Aly Adapted by Lidia
 */
function NMX_get_next_event2($row){
//      if((TRIGGER_MULT_EVENT_ENABLED == $row['type']) &&
(TRIGGER_VALUE_TRUE == $row['value'])){
    if(($row['type'] == 1) && ($row['value'] == 1)){
        $sql = "SELECT * FROM events
                WHERE objectid='$row[objectid]'
                AND eventid >= '$row[eventid]'
                AND object = 0
                AND value = '$row[value]'
                ORDER BY object, objectid, eventid limit 1" ;
    }
    else{
        $sql = "SELECT * FROM events
                WHERE objectid='$row[objectid]'
                AND eventid >= '$row[eventid]'
                AND object = 0
                AND value != '$row[value]'
                ORDER BY object, objectid, eventid limit 1";
    }
}
```

```
}

// $query = mysqli_query($sql) or die("Fail");
// $rez = mysqli_fetch_array($query);
// return $rez;
return $sql;
}

/* function: get_prev_event
 * description: return previous event by value
 * author: Lidia
 */
function NMX_get_prev_event($row){

    if((TRIGGER_MULT_EVENT_ENABLED == $row['type']) &&
(TRIGGER_VALUE_TRUE == $row['value'])){
        $sql = 'SELECT e.eventid, e.value, e.clock '.
            ' FROM events e'.
            ' WHERE e.objectid='.$row['objectid'].
            ' AND e.eventid < '.$row['eventid'].
            ' AND e.object='.EVENT_OBJECT_TRIGGER.
            ' AND e.value='.$row['value'].
            ' ORDER BY e.object, e.objectid, e.eventid DESC';
    }
    else{
        $sql = 'SELECT e.eventid, e.value, e.clock '.
            ' FROM events e'.
            ' WHERE e.objectid='.$row['objectid'].
            ' AND e.eventid < '.$row['eventid'].
    }
}
```

```
        ' AND e.object='.EVENT_OBJECT_TRIGGER.
        ' AND e.value<>'.'$row['value'].
    ' ORDER BY e.object, e.objectid, e.eventid DESC';

}
$rez = DBfetch(DBselect($sql,1));
return $rez;
}

function get_num_param($key, $num){
    $param="";
    $params=preg_split('/[\]\[,]/', $key);

    if(isset($params[$num])) {
        $param=$params[$num];
    }
    return $param;
}

function get_item_description($description, $key){
    $descr=$description;

    for($i=9;$i>0;$i--){
        $descr= str_replace("$${i}",get_num_param($key,$i),$descr);
    }
    return $descr;
}
```

```
?>
```

6- criar o arquivo

/usr/share/zabbix/report/reportPortoAcompanhamentoTicketsDaily_Insert.php

com o conteúdo abaixo:

```
<?php
////////////////////////////////////
// Script Name: reportPortoAvalil.php
// Project: NOC Porto Seguro
// Porto Seguro - Ageri
// Resp: Paulo Tetsuo Hoashi      Date: 27/02/17  Desc: Emission
// Function: Calculo de disponibilidade diaria - cron executado diariamente
////////////////////////////////////
require_once dirname(__FILE__).'/../conf/config_db.php';
require_once dirname(__FILE__).'/includes/phpfunctions.php';
ini_set("memory_limit","256M");

$ano = date('Y', strtotime('-1day'));
$mes = date('m', strtotime('-1day'));
$dia = date('d', strtotime('-1day'));

#$ano = 2017;
#$mes = 4;
#$dia = 3;
$var_ini = mktime("00","00","00",$mes,$dia,$ano);

$ano = date('Y');
$mes = date('m');
```

```
$dia = date('d');
$var_fim = mktime("23","59","59",$mes,$dia,$ano);
echo $dia."/".$mes."/".$ano."\n";

$data_geracao = date("Ymd_His");

$conn = @mysqli_connect($hostLocal, $userLocal, $passLocal, $dbLocal);
$connReport = @mysqli_connect($hostLocal, $userLocal, $passLocal,
$dbLocalReport);

$arrayEventsTickets_EventID = array();
$arrayEvents_EventID = array();
$arrayEvents_ObjectID = array();
$arrayEvents_Clock = array();

$sql_1 = "SELECT DISTINCT eventid FROM events_tickets
        WHERE clk_event_create >= '$var_ini' AND clk_event_create <=
'$var_fim'
        ORDER BY eventid ASC ";
#echo $sql_1."\n";
$query_1 = mysqli_query($connReport, $sql_1);
$num_lines_1 = mysqli_num_rows($query_1);
if ($num_lines_1>0){
    while ($list_1 = mysqli_fetch_array($query_1)) {
        array_push($arrayEventsTickets_EventID, $list_1['eventid']);
    }

    $list_EventsTickets_EventID =
implode("",$arrayEventsTickets_EventID);
```

```
$sql_1 = "SELECT DISTINCT eventid,objectID,clock FROM events
        WHERE object = '0' AND clock >= '$var_ini' AND clock <=
'$var_fim' AND source='0' AND value='1'
        AND not eventid in (".$list_EventsTickets_EventID.")
        ORDER BY eventid ASC ";
}else{
    $sql_1 = "SELECT DISTINCT eventid,objectID,clock FROM events
            WHERE object = '0' AND clock >= '$var_ini' AND clock <=
'$var_fim' AND source='0' AND value='1'
            ORDER BY eventid ASC ";
}

# $sql_1 = "SELECT DISTINCT objectID FROM events WHERE object = '0'
AND source='0' AND value='1' ORDER BY objectid DESC";
# echo $sql_1."\\n";
$query_1 = mysqli_query($conn, $sql_1);
$num_lines_1 = mysqli_num_rows($query_1);
if ($num_lines_1 > 0) {
    while ($list_1 = mysqli_fetch_array($query_1)) {
        # echo $list_1['objectID']."\\n";
        array_push($ArrayEvents_EventID, $list_1['eventid']);
        array_push($ArrayEvents_ObjectID, $list_1['objectID']);
        array_push($ArrayEvents_Clock, $list_1['clock']);
    }

    $Array_ObjectID = array();
    $Array_ObjectID = array_unique($ArrayEvents_ObjectID);
    $list_objectID = implode(",",$Array_ObjectID);
```



```
$ArrayEvents_Length = count($ArrayEvents_EventID);

$arrayTriggersTriggerid = array();
$arrayHostsID = array();
$arrayTriggersDescription = array();
$arrayHostsMaintenanceStatus = array();

$sql_2 = "SELECT hosts.hostid, hosts.maintenance_status,
            triggers.description, triggers.triggerid
        FROM hosts
        LEFT JOIN hosts_groups ON hosts_groups.hostid = hosts.hostid
        LEFT JOIN groups ON groups.groupid = hosts_groups.groupid
        LEFT JOIN items ON items.hostid = hosts.hostid
        LEFT JOIN functions ON functions.itemid=items.itemid
        LEFT JOIN triggers ON triggers.triggerid=functions.triggerid
        WHERE groups.internal = '0' AND groups.groupid != '1'
            AND hosts.status != '1' AND triggers.triggerid in (". $list_objectID. ")
        ORDER BY groups.name, hosts.host, triggers.description ASC";

//Diferente de Host Template
$query_2 = mysqli_query($conn, $sql_2);
#echo $sql_2."\\n";
$num_lines_2 = mysqli_num_rows($query_2);
$index =0;
while ($list_2 = mysqli_fetch_array($query_2)) {
    if (!in_array($list_2["triggerid"], $arrayTriggersTriggerid)){
        $arrayTriggersTriggerid[$index] = $list_2["triggerid"];
        $arrayHostsMaintenanceStatus[$index] =
$list_2["maintenance_status"];
```

```

        $arrayHostsID[$index] = $list_2['hostid'];
        $arrayTriggersDescription[$index] = $list_2['description'];
        $index++;
    }
}
$ArrayHost = array();

$ArrayTriggers_Length = count($arrayTriggersTriggerid);
$ArrayHost = array_unique($arrayHostsID);
$list_Hostid = implode(',',$ArrayHost);

$sql_2_1 = "SELECT * FROM hostmacro WHERE hostid in (". $list_Hostid. ")
ORDER BY hostmacroid ASC";
#echo $sql_2_1."\\n";
$query_2_1 = mysqli_query($conn, $sql_2_1);
while ($list_2_1 = mysqli_fetch_array($query_2_1)) {
    $hostid = $list_2_1['hostid'];
    $macro = $list_2_1['macro'];
    $macro = trim($macro,'{');
    $macro = trim($macro,'}');
    $value = $list_2_1['value'];
    #echo "MACRO: ".$macro." VALUE: ".$value."\\n";
    for ($idx_trigger=0; $idx_trigger<$ArrayTriggers_Length;
$idx_trigger++){
        if ( $hostid == $arrayHostsID[$idx_trigger] ){
            #echo "HOSTID : ".$hostid."\\n";
            #echo "Description:
".$arrayTriggersDescription[$idx_trigger]."\\n";
            #echo "MACRO: ".$macro." VALUE: ".$value."\\n";

```

```
        if (
        strpos($arrayTriggersDescription[$idx_trigger],$macro) != false ){
            $idx_str_init =
        strpos($arrayTriggersDescription[$idx_trigger],$macro );
            $idx_str_end =
        strpos($arrayTriggersDescription[$idx_trigger],'',$idx_str_init);
            $idx_str_end = $idx_str_end - $idx_str_init;
            $macro_diff =
        substr($arrayTriggersDescription[$idx_trigger], $idx_str_init, $idx_str_end );
            #echo "DESCR:
        ".$arrayTriggersDescription[$idx_trigger]."\n";
            #echo "INIT: ".$idx_str_init." END: ".$idx_str_end."
        LENGTH: ".strlen($arrayTriggersDescription[$idx_trigger])."\n";
            #echo "DIFF: ".$macro_diff."\n";
            $arrayTriggersDescription[$idx_trigger] =
        str_replace($macro_diff,$value,$arrayTriggersDescription[$idx_trigger]);
            # $arrayTriggersDescription[$idx_trigger] =
        str_replace($macro,$value,$arrayTriggersDescription[$idx_trigger]);
            $arrayTriggersDescription[$idx_trigger] =
        str_replace('{',",$arrayTriggersDescription[$idx_trigger]);
            $arrayTriggersDescription[$idx_trigger] =
        str_replace('}',",$arrayTriggersDescription[$idx_trigger]);
        #            echo "Host Description:
        ".$arrayTriggersDescription[$idx_trigger]."\n";
        }
    }
}
```

```

$sql_2_1 = "SELECT hosts_templates.hostid , hostmacro.macro,
hostmacro.value FROM hostmacro
        LEFT JOIN hosts_templates ON
hosts_templates.templateid=hostmacro.hostid
        WHERE hosts_templates.hostid in (". $list_Hostid. ") ORDER BY
hostmacroid ASC";
#echo $sql_2_1."\\n";
$query_2_1 = mysqli_query($conn, $sql_2_1);
while ($list_2_1 = mysqli_fetch_array($query_2_1)) {
    $hostid = $list_2_1['hostid'];
    $macro = $list_2_1['macro'];
    $macro = trim($macro,'{');
    $macro = trim($macro,'}');
    $value = $list_2_1['value'];
    #echo "MACRO: ".$macro." VALUE: ".$value."\\n";
    for ($idx_trigger =0; $idx_trigger<$ArrayTriggers_Length;
$idx_trigger++){
        if ( $hostid == $arrayHostsID[$idx_trigger] ){
            #echo "HOSTID :".$hostid."\\n";
            #echo "Description:
".$arrayTriggersDescription[$idx_trigger]."\\n";
            #echo "MACRO: ".$macro." VALUE: ".$value."\\n";
            if ( strpos($arrayTriggersDescription[$idx_trigger],$macro) !==
false ){
                $idx_str_init =
strpos($arrayTriggersDescription[$idx_trigger],$macro );
                $idx_str_end =
strpos($arrayTriggersDescription[$idx_trigger],'',$idx_str_init);

```

```

        $idx_str_end = $idx_str_end - $idx_str_init;
        $macro_diff =
substr($arrayTriggersDescription[$idx_trigger], $idx_str_init, $idx_str_end );
        #echo "DESCR:
".$arrayTriggersDescription[$idx_trigger]."\n";
        #echo "INIT: ".$idx_str_init." END: ".$idx_str_end."
LENGTH: ".strlen($arrayTriggersDescription[$idx_trigger])."\n";
        #echo "DIFF: ".$macro_diff."\n";
        $arrayTriggersDescription[$idx_trigger] =
str_replace($macro_diff,$value,$arrayTriggersDescription[$idx_trigger]);
        # $arrayTriggersDescription[$idx_trigger] =
str_replace($macro,$value,$arrayTriggersDescription[$idx_trigger]);
        $arrayTriggersDescription[$idx_trigger] =
str_replace('{','',$arrayTriggersDescription[$idx_trigger]);
        $arrayTriggersDescription[$idx_trigger] =
str_replace('}','',$arrayTriggersDescription[$idx_trigger]);
#        echo "Host Description:
".$arrayTriggersDescription[$idx_trigger]."\n";
    }
}
}

}

$ArrayGlobal_Macro = array();
$ArrayGlobal_Value = array();

$sql_2_2 = "SELECT * FROM globalmacro ORDER BY globalmacroid ASC";
$query_2_2 = mysqli_query($conn, $sql_2_2);

```

```

$num_lines_2_2 = mysqli_num_rows($query_2_2);
while ($list_2_2 = mysqli_fetch_array($query_2_2)) {
    array_push($ArrayGlobal_Macro,$list_2_2['macro']);
    array_push($ArrayGlobal_Value,$list_2_2['value']);
#    echo "MACRO: ".$macro." VALUE: ".$value."\n";
}
$ArrayGlobal_Length = count($ArrayGlobal_Macro);
for ($idx_macro = 0 ; $idx_macro<$ArrayGlobal_Length; $idx_macro++){
    $macro = $ArrayGlobal_Macro[$idx_macro];
    $macro = trim($macro,'{');
    $macro = trim($macro,'}');
    $value = $ArrayGlobal_Value[$idx_macro];
    #echo "MACRO: ".$macro." VALUE: ".$value."\n";
    for ($idx_trigger =0; $idx_trigger<$ArrayTriggers_Length;
$idx_trigger++){
        #echo "Description:
".$ArrayTriggersDescription[$idx_trigger]."\n";
        if ( strpos($arrayTriggersDescription[$idx_trigger],$macro ) !==
false){
            #echo "MACRO: ".$macro." VALUE: ".$value."\n";
            $idx_str_init =
strpos($arrayTriggersDescription[$idx_trigger],$macro );
            $idx_str_end =
strpos($arrayTriggersDescription[$idx_trigger],'',$idx_str_init);
            $idx_str_end = $idx_str_end - $idx_str_init;
            $macro_diff =
substr($arrayTriggersDescription[$idx_trigger], $idx_str_init, $idx_str_end );
            #echo "DESCR:
".$ArrayTriggersDescription[$idx_trigger]."\n";

```

```

        #echo "INIT: ".$idx_str_init." END: ".$idx_str_end."
LENGTH: ".strlen($arrayTriggersDescription[$idx_trigger])."\n";
        #echo "DIFF: ".$macro_diff."\n";

        $arrayTriggersDescription[$idx_trigger] =
str_replace($macro_diff,$value,$arrayTriggersDescription[$idx_trigger]);
        $arrayTriggersDescription[$idx_trigger] =
str_replace('{',"',$arrayTriggersDescription[$idx_trigger]);
        $arrayTriggersDescription[$idx_trigger] =
str_replace('}',"',',$arrayTriggersDescription[$idx_trigger]);
        #echo "MACRO: ".$macro." VALUE: ".$value."\n";
        #echo "Global Description:
".$arrayTriggersDescription[$idx_trigger]."\n";
    }
}
}

$row = array();
for ( $idx=0; $idx<$ArrayEvents_Length; $idx++){
    $UpTime = 0; $DownTime = 0; $UnknownTime = 0;
    $insert = "INSERT INTO events_tickets ";
    $update = "UPDATE events_tickets SET ";
    $row['clock'] = $ArrayEvents_Clock[$idx];
    $row['type'] = 0;
    $row['value'] = 1;
    $row['objectid'] = $ArrayEvents_ObjectID[$idx];
    $row['eventid'] = $ArrayEvents_EventID[$idx];

    $idx_trigger =

```

```

array_search($ArrayEvents_ObjectID[$idx],$arrayTriggersTriggerid);
    $trigger_description = $arrayTriggersDescription[$idx_trigger];
    $maintenance_status = $arrayHostsMaintenanceStatus[$idx_trigger];

    $sql_4 = NMX_get_next_event($row);
    $next_clock = 0;
    $query_4 = mysqli_query($conn, $sql_4);
    while ($list_4 = mysqli_fetch_array($query_4)) {
        $next_clock=$list_4['clock'];
    }
    $ack_clock = 0;
    $sql_5 = "SELECT * FROM acknowledges
              LEFT JOIN users ON users.userid =
acknowledges.userid
              WHERE acknowledges.eventid =
".$ArrayEvents_EventID[$idx].
              AND acknowledges.message like 'ticket %' ORDER by
acknowledges.clock DESC limit 1";
    $query_5 = mysqli_query($conn, $sql_5);
    while ($list_5 = mysqli_fetch_array($query_5)) {
        $ack_clock = $list_5['clock'];
        $ack_userid= $list_5['userid'];
        $ack_message=$list_5['message'];
    }
    # $insert = "Triggeird: ".$ArrayEvents_ObjectID[$idx]." EventID:
".$ArrayEvents_EventID[$idx]." EventClock_ON: ".$ArrayEvents_Clock[$idx];
    $insert_parameter = "eventsTicketID,eventid,objectID,
clk_event_create,trigger_description,maintenance";
    $insert_value =

```



```

$ArrayEvents_EventID[$idx].",".$ArrayEvents_EventID[$idx].",".$ArrayEvents
_ObjectID[$idx].",".$ArrayEvents_Clock[$idx].",".$trigger_description."",".$mai
ntenance_status;

$update .= " clk_event_create=".$ArrayEvents_Clock[$idx].",
objectID=".$ArrayEvents_ObjectID[$idx].",trigger_description=".$trigger_desc
ription.""";

#$update .= " clk_event_create=".$ArrayEvents_Clock[$idx];
if ($next_clock>0) {
    #$insert = $insert." EventClock_OFF: ".$next_clock;
    $insert_parameter .= ",clk_event_cleared";
    $insert_value .= ",".$next_clock;
    $update .= " ,clk_event_cleared=".$next_clock;
}
if ($ack_clock > 0){
    #$insert = $insert." AckClock_ON: ".$ack_clock;
    $insert_parameter .= ",clk_ack,userid,ticket";
    $insert_value .=
",".$ack_clock.",".$ack_userid.",".$ack_message.""";
    $update .=
",clk_ack=".$ack_clock.",userid=".$ack_userid.",ticket=".$ack_message.""";
}
# $sql_6 = "SELECT * FROM events_tickets WHERE eventid
=".$ArrayEvents_EventID[$idx];
# $query_6 = mysqli_query($connReport, $sql_6);
# $num_lines_6 = mysqli_num_rows($query_6);
# if ($num_lines_6 > 0){
# if (not
(in_array($ArrayEvents_EventID[$idx],$ArrayEventsTickets_EventID))) {
# $sql = $update." WHERE eventid

```

```

=".$ArrayEvents_EventID[$idx];
#    }else{
        $sql = $insert."(".$insert_parameter.") VALUES
('$insert_value.');"
#    }
    echo $sql."\n";
    if ($connReport->query($sql) === FALSE){
        echo "Error ".$connReport->error;
    }
}
#echo date('H:i:s') , " Rename worksheet" , EOL;
#$callStartTime = microtime(true);
}

mysqli_close($connReport);
mysqli_close($conn);
echo "FIM";
?>

```

7- criar o arquivo

/usr/share/zabbix/report/reportPortoAcompanhamentoTicketsDaily_Insert.sh

com o conteúdo abaixo:

```

#!/bin/bash

process=$(ps -ef | grep php | grep
/usr/share/zabbix/report/reportPortoAcompanhamentoTicketsDaily_Insert.php
| grep -v grep | wc -l)

```

```

        echo $process
    if [ $process -eq 0 ]
    then
        /usr/bin/php
        /usr/share/zabbix/report/reportPortoAcompanhamentoTicketsDaily_Insert.php
    fi

```

8- criar o arquivo

/usr/share/zabbix/report/reportPortoAcompanhamentoTicketsDaily_UpdateAck.php com o conteúdo abaixo:

```

<?php
////////////////////////////////////
// Script Name: reportPortoAcompanhamentoTicketsDaily_UpdateAck.php
// Project: NOC Porto Seguro
// Porto Seguro - Ageri
// Resp: Paulo Tetsuo Hoashi      Date: 27/02/17  Desc: Emission
// Function: Check de ack em events_tickets.
////////////////////////////////////
require_once dirname(__FILE__).'/../conf/config_db.php';
require_once dirname(__FILE__).'/includes/phpfunctions.php';
ini_set("memory_limit","256M");

$ano = date('Y', strtotime('-1month'));
$mes = date('m', strtotime('-1month'));
$dia = date('d', strtotime('-1month'));
#$ano=2017;
#$mes=4;
#$dia=2;

```

```
$var_ini = mktime("00","00","00",$mes,$dia,$ano);

#$var_fim = mktime();
#echo $dia."/".$mes."/".$ano."\n";

$data_geracao = date("Ymd_His");

$conn = @mysqli_connect($hostLocal, $userLocal, $passLocal, $dbLocal);
$connReport = @mysqli_connect($hostLocal, $userLocal, $passLocal,
$dbLocalReport);

$ArrayEvents_EventID = array();
$ArrayEvents_ObjectID = array();
$ArrayEvents_Clock = array();

$sql_1 = "SELECT DISTINCT eventid,objectID, clk_event_create FROM
events_tickets
          WHERE (isnull(clk_ack))
          AND clk_event_create >= ".$var_ini."
          ORDER BY eventid ASC ";

echo $sql_1."\n";
$query_1 = mysqli_query($connReport, $sql_1);
$num_lines_1 = mysqli_num_rows($query_1);
echo "LINE ".$num_lines_1."\n";

if ($num_lines_1>0){
while ($list_1 = mysqli_fetch_array($query_1)) {
#   echo $list_1['objectID']."\n";
    array_push($ArrayEvents_EventID, $list_1['eventid']);
}
```

```

        array_push($ArrayEvents_ObjectID, $list_1['objectID']);
        array_push($ArrayEvents_Clock, $list_1['clk_event_create']);
    }

    $ArrayEvents_Length = count($ArrayEvents_EventID);

    $row = array();
    for ( $idx=0; $idx<$ArrayEvents_Length; $idx++){
        $UpTime = 0; $DownTime = 0; $UnknownTime = 0;
        $update = "UPDATE events_tickets SET ";
        $row['clock'] = $ArrayEvents_Clock[$idx];
        $row['type'] = 0;
        $row['value'] = 1;
        $row['objectid'] = $ArrayEvents_ObjectID[$idx];
        $row['eventid'] = $ArrayEvents_EventID[$idx];

        $ack_clock = 0;
        # $sql_5 = "SELECT * FROM acknowledges
        # WHERE acknowledges.eventid = ".$ArrayEvents_EventID[$idx].". AND
        # ( acknowledges.userid=25) OR (acknowledges.userid=1))
        # ORDER by acknowledges.clock ASC LIMIT 1";
        $sql_5 = "SELECT * FROM acknowledges
        LEFT JOIN users ON users.userid = acknowledges.userid
        WHERE acknowledges.eventid = ".$ArrayEvents_EventID[$idx].". AND (
        (acknowledges.userid=25) OR (acknowledges.userid=1))
        ORDER by acknowledges.clock ASC LIMIT 1";

        $query_5 = mysqli_query($conn, $sql_5);
        while ($list_5 = mysqli_fetch_array($query_5)) {

```

```
$ack_clock = $list_5['clock'];
$ack_userid= $list_5['userid'];
$ack_alias= $list_5['alias'];
$ack_message=$list_5['message'];
}
$update .= " clk_event_create=".$ArrayEvents_Clock[$idx];
if ($ack_clock > 0){
    $update .=
    ",clk_ack=".$ack_clock.",userid=".$ack_userid.",ticket=".$ack_message.",alias=".$ack_alias.""";
    $sql = $update." WHERE eventid
    =".$ArrayEvents_EventID[$idx];
    echo $sql."\n";
    if ($connReport->query($sql) === FALSE){
        echo "Error ".$connReport->error;
    }
}
}
#echo date('H:i:s') , " Rename worksheet" , EOL;
#$callStartTime = microtime(true);
}

mysqli_close($connReport);
mysqli_close($conn);
echo "FIM";
?>
```

9- criar o arquivo

/usr/share/zabbix/report/reportPortoAcompanhamentoTicketsDaily_UpdateAck.sh com o conteúdo abaixo:

```
#!/bin/bash

process=$(ps -ef | grep php | grep
/usr/share/zabbix/report/reportPortoAcompanhamentoTicketsDaily_UpdateAck.php | grep -v grep | wc -l)
    echo $process
if [ $process -eq 0 ]
then
    /usr/bin/php
/usr/share/zabbix/report/reportPortoAcompanhamentoTicketsDaily_UpdateAck.php
fi
```

10- criar o arquivo

/usr/share/zabbix/report/reportPortoAcompanhamentoTicketsDaily_UpdateEvent.php com o conteúdo abaixo:

```
<?php
////////////////////////////////////
// Script Name: reportPortoAcompanhamentoTicketsDaily_UpdateEvent.php
// Project: NOC Porto Seguro
// Porto Seguro - Ageri
// Resp: Paulo Tetsuo Hoashi      Date: 27/02/17  Desc: Emission
// Function: Check evento de normalizacao na tabela de events_tickets
////////////////////////////////////
require_once dirname(__FILE__).'/../conf/config_db.php';
```

```
require_once dirname(__FILE__).'/includes/phpfunctions.php';
ini_set("memory_limit","256M");

$ano = date('Y', strtotime('-1month'));
$mes = date('m', strtotime('-1month'));
$dia = date('d', strtotime('-1month'));

$var_ini = mktime("00","00","00",$mes,$dia,$ano);

#$var_fim = mktime();
#echo $dia."/".$mes."/".$ano."\n";

$data_geracao = date("Ymd_His");

$conn = @mysqli_connect($hostLocal, $userLocal, $passLocal, $dbLocal);
$connReport = @mysqli_connect($hostLocal, $userLocal, $passLocal,
$dbLocalReport);

$arrayEvents_EventID = array();
$arrayEvents_ObjectID = array();
$arrayEvents_Clock = array();

$sql_1 = "SELECT DISTINCT eventid,objectID, clk_event_create FROM
events_tickets
        WHERE (isnull(clk_event_cleared))
        AND clk_event_create >= ".$var_ini." AND (isnull(clk_disable))
        ORDER BY eventid ASC ";

#echo $sql_1."\n";
$query_1 = mysqli_query($connReport, $sql_1);
```



```
$num_lines_1 = mysqli_num_rows($query_1);
echo "Number events PROBLEM: ".$num_lines_1."\n";

if ($num_lines_1>0){
while ($list_1 = mysqli_fetch_array($query_1)) {
#    echo $list_1['objectID']."\n";
    array_push($ArrayEvents_EventID, $list_1['eventid']);
    array_push($ArrayEvents_ObjectID, $list_1['objectID']);
    array_push($ArrayEvents_Clock, $list_1['clk_event_create']);
}

$ArrayEvents_Length = count($ArrayEvents_EventID);

$row = array();
for ( $idx=0; $idx<$ArrayEvents_Length; $idx++){
    $UpTime = 0; $DownTime = 0; $UnknownTime = 0;
    $update = "UPDATE events_tickets SET ";
    $row['clock'] = $ArrayEvents_Clock[$idx];
    $row['type'] = 0;
    $row['value'] = 1;
    $row['objectid'] =$ArrayEvents_ObjectID[$idx];
    $row['eventid'] =$ArrayEvents_EventID[$idx];

    $sql_4 = NMX_get_next_event2($row);
    $next_clock = 0;
    $query_4 = mysqli_query($conn, $sql_4);
    while ($list_4 = mysqli_fetch_array($query_4)) {
        $next_clock=$list_4['clock'];
    }
}
```

```

$update .= " clk_event_create=".$ArrayEvents_Clock[$idx];
if ($next_clock>0) {
    $update .= " ,clk_event_cleared=".$next_clock;
    $sql = $update." WHERE eventid
=".$ArrayEvents_EventID[$idx];
    echo $sql."\n";
    if ($connReport->query($sql) === FALSE){
        echo "Error ".$connReport->error;
    }
}
}
else{
    echo $idx." eventid: ".$ArrayEvents_EventID[$idx]." Data:
".$ArrayEvents_Clock[$idx]."\n";
}

}

#echo date('H:i:s') , " Rename worksheet" , EOL;
#$callStartTime = microtime(true);
}

mysqli_close($connReport);
mysqli_close($conn);
echo "FIM";
?>

```

11- criar o arquivo

/usr/share/zabbix/report/reportPortoAcompanhamentoTicketsDaily_UpdateEvent.sh com o conteúdo abaixo:

```
#!/bin/bash
```

```

process=$(ps -ef | grep php | grep
/usr/share/zabbix/report/reportPortoAcompanhamentoTicketsDaily_UpdateEv
ent.php | grep -v grep | wc -l)
    echo $process
if [ $process -eq 0 ]
then
    /usr/bin/php
/usr/share/zabbix/report/reportPortoAcompanhamentoTicketsDaily_UpdateEv
ent.php
fi

```

12- criar o arquivo

/usr/share/zabbix/report/reportPortoAcompanhamentoTicketsDaily_DeleteTicket.php com o conteúdo abaixo:

```

<?php
////////////////////////////////////
// Script Name: reportPortoAcompanhamentoTicketsDaily_DeleteTicket.php
// Project: NOC Porto Seguro
// Porto Seguro - Ageri
// Resp: Paulo Tetsuo Hoashi      Date: 27/02/17  Desc: Emission
// Function: Deleta os eventos anteriores a X tempos da tabela events_tickets
- cron executado diariamente
////////////////////////////////////
require_once dirname(__FILE__).'../conf/config_db.php';
ini_set("memory_limit","256M");

$ano = date('Y', strtotime('-7month'));

```

```
$mes = date('m', strtotime('-7month'));
$dia = date('d', strtotime('-7month'));

$var_ini = mktime("00","00","00",$mes,$dia,$ano);

$data_geracao = date("Ymd_His");

$conn = @mysqli_connect($hostLocal, $userLocal, $passLocal, $dbLocal);
$connReport = @mysqli_connect($hostLocal, $userLocal, $passLocal,
$dbLocalReport);

$sql_1 = "DELETE FROM events_tickets WHERE clk_event_create <
".$var_ini.";";
echo $sql_1."\n";
if ($connReport->query($sql_1) === FALSE){
    echo "Error ".$connReport->error;
}

mysqli_close($connReport);
#mysqli_close($conn);
echo "FIM";
?>
```

13- criar o arquivo

/usr/share/zabbix/report/reportPortoAcompanhamentoTicketsDaily_DeleteTicket
.sh com o conteúdo abaixo:

```
#!/bin/bash
```

```

process=$(ps -ef | grep php | grep
/usr/share/zabbix/report/reportPortoAcompanhamentoTicketsDaily_DeleteTick
et.php | grep -v grep | wc -l)
    echo $process
if [ $process -eq 0 ]
then
    /usr/bin/php
/usr/share/zabbix/report/reportPortoAcompanhamentoTicketsDaily_DeleteTick
et.php
fi

```

14- adicionar as linhas abaixo no arquivo /etc/crontab.

```

#Relatorio zabbix
*/1 * * * * root
/usr/share/zabbix/report/reportPortoAcompanhamentoTicketsDaily_Insert.sh
*/1 * * * * root
/usr/share/zabbix/report/reportPortoAcompanhamentoTicketsDaily_UpdateAc
k.sh
*/1 * * * * root
/usr/share/zabbix/report/reportPortoAcompanhamentoTicketsDaily_UpdateEv
ent.sh
00 1 * * * root
/usr/share/zabbix/report/reportPortoAcompanhamentoTicketsDaily_DeleteTic
ket.sh

```

15- adicionar a permissão de execução dos scripts

```

chmod +x /usr/share/zabbix/report/*.sh

```

16- criar o arquivo /usr/share/zabbix/reportTicket.php com o conteúdo abaixo:

```
<?php
/*
** Zabbix
** Copyright (C) 2001-2017 Zabbix SIA
**
** This program is free software; you can redistribute it and/or modify
** it under the terms of the GNU General Public License as published by
** the Free Software Foundation; either version 2 of the License, or
** (at your option) any later version.
**
** This program is distributed in the hope that it will be useful,
** but WITHOUT ANY WARRANTY; without even the implied warranty of
** MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See
the
** GNU General Public License for more details.
**
** You should have received a copy of the GNU General Public License
** along with this program; if not, write to the Free Software
** Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301,
USA.
**/
////////////////////////////////////
// Script Name: reportTicket.php
// Project: NOC Porto Seguro
// Porto Seguro - Ageri
// Resp: Paulo Tetsuo Hoashi      Date: 08/05/17  Desc: Emission
//                               Date: 18/08/17  Desc: Mudanca no SELECT
```

```
// Function: Relatorio de Acompanhamento de Ticket
////////////////////////////////////

require_once dirname(__FILE__).'/include/config.inc.php';
require_once dirname(__FILE__).'/include/hosts.inc.php';
require_once dirname(__FILE__).'/include/events.inc.php';
require_once dirname(__FILE__).'/include/actions.inc.php';
require_once dirname(__FILE__).'/include/discovery.inc.php';
require_once dirname(__FILE__).'/include/html.inc.php';

require_once dirname(__FILE__).'/conf/config_db.php';
require_once dirname(__FILE__).'/report/includes/phpfunctions.php';

if (hasRequest('csv_export')) {
    $csvExport = true;
    $csvRows = [];

    $page['type'] = detect_page_type(PAGE_TYPE_CSV);
    $page['file'] = 'zbx_events_export.csv';

    require_once dirname(__FILE__).'/include/func.inc.php';
}
else {
    $csvExport = false;

    $page['title'] = _('Ticket Report');
    $page['file'] = 'reportTicket.php';
    $page['scripts'] = ['class.calendar.js', 'gtlc.js'];
}
```

```

$page['type'] = detect_page_type(PAGE_TYPE_HTML);

if (PAGE_TYPE_HTML == $page['type']) {
    define('ZBX_PAGE_DO_REFRESH', 1);
}
}

require_once dirname(__FILE__).'/include/page_header.php';

$allow_discovery = check_right_on_discovery();
$allowed_sources[] = EVENT_SOURCE_TRIGGERS;
if ($allow_discovery) {
    $allowed_sources[] = EVENT_SOURCE_DISCOVERY;
}

//          VAR          TYPE OPTIONAL  FLAGS      VALIDATION
EXCEPTION
$fields = [
    'source'=>          [T_ZBX_INT, O_OPT, P_SYS,
    IN($allowed_sources), null],
    'groupid'=>          [T_ZBX_INT, O_OPT, P_SYS,   DB_ID,
    null],
    'hostid'=>           [T_ZBX_INT, O_OPT, P_SYS,   DB_ID,
    null],
    'triggerid'=>        [T_ZBX_INT, O_OPT, P_SYS,   DB_ID,
    null],
    'period'=>           [T_ZBX_INT, O_OPT, null, null,      null],
    'stime'=>            [T_ZBX_INT, O_OPT, null, null,      null],
    'load'=>              [T_ZBX_STR, O_OPT, P_SYS,   NULL,

```



```

    null],
    'fullscreen'=>      [T_ZBX_INT, O_OPT, P_SYS,   IN('0,1'),
    null],
    'csv_export'=>      [T_ZBX_STR, O_OPT, P_SYS,   null,
    null],
    'filter_rst'=>      [T_ZBX_STR, O_OPT, P_SYS,   null,
    null],
    'filter_set'=>      [T_ZBX_STR, O_OPT, P_SYS,   null,
    null],
    'filter_timesince' => [T_ZBX_STR, O_OPT, P_UNSET_EMPTY, null,
null],
    'filter_timetill' => [T_ZBX_STR, O_OPT, P_UNSET_EMPTY, null,
null]

    // ajax
    //'favobj'=>      [T_ZBX_STR, O_OPT, P_ACT,   null,
    null],
    //'favid'=>      [T_ZBX_INT, O_OPT, P_ACT,   null,
    null]

];
check_fields($fields);

#echo "0-TIMESINCE: ".$_REQUEST['filter_timesince']."<BR>";
#echo "0-TIMETILL: ".$_REQUEST['filter_timetill']."<BR>";
/*
* Permissions
*/
if (getRequest('groupid') && !API::HostGroup()-
>isReadable([getRequest('groupid')])) {
    access_deny();

```

```
}  
if (getRequest('hostid') && !API::Host()->isReadable([getRequest('hostid')])) {  
    access_deny();  
}  
if (getRequest('triggerid') && !API::Trigger()-  
>isReadable([getRequest('triggerid')])) {  
    access_deny();  
}  
  
/*  
 * Ajax  
 */  
/*if (hasRequest('favobj')) {  
    // saving fixed/dynamic setting to profile  
    if ('timelinefixedperiod' == getRequest('favobj')) {  
        if (hasRequest('favid')) {  
            CProfile::update('web.reporTicket.timelinefixed',  
getRequest('favid'), PROFILE_TYPE_INT);  
        }  
    }  
}  
*/  
if ($page['type'] == PAGE_TYPE_JS || $page['type'] ==  
PAGE_TYPE_HTML_BLOCK) {  
    require_once dirname(__FILE__).'/include/page_footer.php';  
    exit;  
}  
  
/*
```

```
* Filter
*/
if (hasRequest('filter_set')) {
    CProfile::update('web.reporTicket.filter.triggerid', getRequest('triggerid',
0), PROFILE_TYPE_ID);
}
elseif (hasRequest('filter_rst')) {
    DBStart();
    CProfile::delete('web.reporTicket.filter.triggerid');
    DBend();
    $_REQUEST['filter_timesince'] = 0;
    $_REQUEST['filter_timetill'] = 0;
}

if (!hasRequest('filter_rst')) {
/*    if ( empty($_REQUEST['filter_timesince'])) {
        $ano = date('Y');
        $mes = date('m');
        $dia = date('d');
        #$hora = date('H');
        #$minuto = date('i');
        $_REQUEST['filter_timesince']=$ano."".$mes."".$dia."000000";
    }
    if ( empty($_REQUEST['filter_timetill'])) {
        $ano = date('Y');
        $mes = date('m');
        $dia = date('d');
        #$hora = date('H');
```

```
        # $minuto = date('i');
        $_REQUEST['filter_timetill'] = $ano."".$mes."".$dia."235959";
    }*/
    $_REQUEST['filter_timesince'] = getRequest('filter_timesince',
CProfile::get('web.reportTicket.timesince', 0) );
    $_REQUEST['filter_timetill'] = getRequest('filter_timetill',
CProfile::get('web.reportTicket.timetill', 0) );
    #echo "TIMES: ".$_REQUEST['filter_timesince']."<br>";
}
CProfile::update('web.reportTicket.timesince', getRequest('filter_timesince', 0),
    PROFILE_TYPE_STR
);
CProfile::update('web.reportTicket.timetill', getRequest('filter_timetill', 0),
    PROFILE_TYPE_STR
);

$config = select_config();

$_REQUEST['filter_timesince'] =
zbxDateToTime($_REQUEST['filter_timesince']
    ? $_REQUEST['filter_timesince'] :
date(TIMESTAMP_FORMAT_ZERO_TIME, time() - SEC_PER_DAY));
$_REQUEST['filter_timetill'] = zbxDateToTime($_REQUEST['filter_timetill']
    ? $_REQUEST['filter_timetill'] :
date(TIMESTAMP_FORMAT_ZERO_TIME, time()));
#echo "3-TIMESINCE: ".$_REQUEST['filter_timesince']."<BR>";
#echo "3-TIMETILL: ".$_REQUEST['filter_timetill']."<BR>";

$triggerId = CProfile::get('web.reporTicket.filter.triggerid', 0);
```

```
$source = ($triggerId != 0 && hasRequest('filter_set'))
    ? EVENT_SOURCE_TRIGGERS
    : getRequest('source', CProfile::get('web.reporTicket.source',
EVENT_SOURCE_TRIGGERS));

CProfile::update('web.reporTicket.source', $source, PROFILE_TYPE_INT);
$from=$_REQUEST['filter_timesince'];
$till=$_REQUEST['filter_timetill'];

/*
 * Display
 */
if ($source == EVENT_SOURCE_TRIGGERS) {
    if ($triggerId != 0 && hasRequest('filter_set')) {
        $host = API::Host()->get([
            'output' => ['hostid'],
            'selectGroups' => ['groupid'],
            'triggerids' => [$triggerId],
            'limit' => 1
        ]);

        $host = reset($host);
        $hostid = $host['hostid'];
        $group = reset($host['groups']);
        $groupid = $group['groupid'];
    }
    else {
        $groupid = getRequest('groupid');
```

```
$hostid = getRequest('hostid');
}

$pageFilter = new CPageFilter([
    'groups' => [
        'monitored_hosts' => true,
        'with_monitored_triggers' => true
    ],
    'hosts' => [
        'monitored_hosts' => true,
        'with_monitored_triggers' => true
    ],
    'hostid' => $hostid,
    'groupid' => $groupid
]);
}

if ($csvExport) {
    if (!hasRequest('hostid')) {
        $_REQUEST['hostid'] = 0;
    }
    if (!hasRequest('groupid')) {
        $_REQUEST['groupid'] = 0;
    }
}
else {
    if ($source == EVENT_SOURCE_TRIGGERS) {
        // try to find matching trigger when host is changed
        // use the host ID from the page filter since it may not be present
```

in the request

```
// if all hosts are selected, preserve the selected trigger
if ($triggerId != 0 && $pageFilter->hostid != 0) {
    $old_triggers = API::Trigger()->get([
        'output' => ['description', 'expression'],
        'selectHosts' => ['hostid', 'host'],
        'triggerids' => [$triggerId]
    ]);
    $old_trigger = reset($old_triggers);

    $old_trigger['hosts'] = zbx_toHash($old_trigger['hosts'],
'hostid');

    // if the trigger doesn't belong to the selected host - find a
new one on that host
    if (!array_key_exists($pageFilter->hostid,
$old_trigger['hosts'])) {
        $triggerId = 0;

        $old_expression =
CMacrosResolverHelper::resolveTriggerExpression($old_trigger['expression'])
;

        $new_triggers = API::Trigger()->get([
            'output' => ['triggerid', 'description',
'expression'],
            'selectHosts' => ['hostid', 'host'],
            'filter' => ['description' =>
$old_trigger['description']],
```

```
        'hostids' => [$pageFilter->hostid]
    );

    $new_triggers =
CMacrosResolverHelper::resolveTriggerExpressions($new_triggers);

    foreach ($new_triggers as $new_trigger) {
        $new_trigger['hosts'] =
zbx_toHash($new_trigger['hosts'], 'hostid');

        foreach ($old_trigger['hosts'] as $old_host) {
            $new_expression =
triggerExpressionReplaceHost($new_trigger['expression'],

                $new_trigger['hosts'][$pageFilter->hostid]['host'], $old_host['host']
            );

            if ($old_expression ===
$new_expression) {

                CProfile::update('web.reporTicket.filter.triggerid',
$new_trigger['triggerid'], PROFILE_TYPE_ID);

                $triggerId =
$new_trigger['triggerid'];

                break 2;
            }
        }
    }
}
```



```
    }  
}  
  
$eventsWidget = (new CWidget()->setTitle(_('Ticket Report'));  
  
// header  
$frmForm = (new CForm('get'))  
    ->addVar('page', getPageNumber(), 'page_csv');  
if (hasRequest('source')) {  
    $frmForm->addVar('source', getRequest('source'), 'source_csv');  
}  
  
if ($source == EVENT_SOURCE_TRIGGERS) {  
    $frmForm->addVar('groupid', $pageFilter->groupid,  
'groupid_csv');  
    $frmForm->addVar('hostid', $pageFilter->hostid, 'hostid_csv');  
  
    if ($triggerId) {  
        $frmForm->addVar('triggerid', $triggerId, 'triggerid_csv');  
    }  
}  
  
$frmForm->addVar('fullscreen', getRequest('fullscreen'));  
  
$controls = new CList();  
  
// add host and group filters to the form  
if ($source == EVENT_SOURCE_TRIGGERS) {  
    if (getRequest('triggerid') != 0) {
```

```
$frmForm->addVar('triggerid', getRequest('triggerid'),
'triggerid_filter');
    }

    $controls
        ->addItem([
            new CLabel(_('Group'), 'groupid'),
            (new CDiv())-
>addClass(ZBX_STYLE_FORM_INPUT_MARGIN),
            $pageFilter->getGroupsCB()
        ])
        ->addItem([
            new CLabel(_('Host'), 'hostid'),
            (new CDiv())-
>addClass(ZBX_STYLE_FORM_INPUT_MARGIN),
            $pageFilter->getHostsCB()
        ]);
    }

    $controls
        ->addItem(new CSubmit('csv_export', _('Export to CSV')))
        ->addItem(get_icon('fullscreen', ['fullscreen' =>
getRequest('fullscreen')]));

    $frmForm->addItem($controls);
    $eventsWidget->setControls($frmForm);

    $filterForm = (new CFilter('web.reporTicket.filter.state'))
```

```

->addVar('fullscreen', getRequest('fullscreen'));

if ($source == EVENT_SOURCE_TRIGGERS) {
    $filterForm->addVar('triggerid', $triggerId);
    $filterForm-
>addVar('filter_timesince',$_REQUEST['filter_timesince']);
    $filterForm->addVar('filter_timetill',$_REQUEST['filter_timetill']);
    $filterForm->addVar('groupid', $pageFilter->groupid);
    $filterForm->addVar('hostid', $pageFilter->hostid);

    if ($triggerId > 0) {
        $dbTrigger = API::Trigger()->get([
            'triggerids' => $triggerId,
            'output' => ['description', 'expression'],
            'selectHosts' => ['name'],
            'preservekeys' => true,
            'expandDescription' => true
        ]);
        if ($dbTrigger) {
            $dbTrigger = reset($dbTrigger);
            $host = reset($dbTrigger['hosts']);

            $trigger =
$host['name'].NAME_DELIMITER.$dbTrigger['description'];
        }
        else {
            $triggerId = 0;
        }
    }
}

```

```

        if (!isset($trigger)) {
            $trigger = "";
        }

        $filterColumn = new CFormList();

        $filterColumn->addRow(
            _('Trigger'),
            [
                (new CTextBox('trigger', $trigger, true))-
>setWidth(ZBX_TEXTAREA_FILTER_BIG_WIDTH),
                (new CDiv())-
>addClass(ZBX_STYLE_FORM_INPUT_MARGIN),
                (new CButton('btn1', _('Select'))
                    ->addClass(ZBX_STYLE_BTN_GREY)
                    ->onClick('return PopUp("popup.php?".
                        'dstfrm=zbx_filter'.
                        '&dstfld1=triggerid'.
                        '&dstfld2=trigger'.
                        '&srctbl=triggers'.
                        '&srcfld1=triggerid'.
                        '&srcfld2=description'.
                        '&real_hosts=1'.
                        '&monitored_hosts=1'.
                        '&with_monitored_triggers=1'.
                        ($pageFilter->hostid ?
'&only_hostid='.$pageFilter->hostid : "").
                        ''));'
            ])
        )

```

```

        ]

    );

#echo "4-TIMESINCE: ".$_REQUEST['filter_timesince']. "<BR>";
#echo "4-TIMETILL: ".$_REQUEST['filter_timetill']. "<BR>";

#        $filterForm->addColumn($filterColumn);
#        $filterColumn2 = new CFormList();
#        // filter period
#        $filterColumn->addRow(_('From'), createDateSelector('filter_timesince',
$_REQUEST['filter_timesince'], 'filter_timetill'));
#        $filterColumn2->addRow(_('From'), createDateSelector('stime',
$stime, 'stime'));
#        $filterColumn->addRow(_('To'), createDateSelector('filter_timetill',
$_REQUEST['filter_timetill'], 'filter_timesince'));
#        $filterColumn2->addRow(_('To'), createDateSelector('period', $period,
'period'));

#        $filterForm->addColumn($filterColumn);
#    }

#    $filterForm->addNavigator();

#    $eventsWidget->addItem($filterForm);

#    $table = new CTableInfo();
#}

# $config = select_config();

```

```
// headers
$header = [
    ($pageFilter->hostid == 0) ? _('Device Name') : null,
    _("Network Address"),
    _("Created On"),
    _("Cleared On"),
    _("Alarm Title"),
    _("Severity"),
    _("Alarm Duration (HH:MM:SS)"),
    _("Time to Trouble Ticket (HH:MM:SS)"),
    _("Trouble Ticket ID"),
    _("Ticketed By"),
    _("Maintenance"),
    _("ALARM KEY")
];

if ($csvExport) {
    $csvRows[] = $header;
}
else {
    $table->setHeader($header);
}

if ($csvExport || $pageFilter->hostsSelected || $triggerId != 0) {
    $knownTriggerIds = [];
    $validTriggerIds = [];

    $triggerOptions = [
        'output' => ['triggerid'],
    ]
}
```

```

        'preservekeys' => true,
        'monitored' => true
    ];

    $allEventsSliceLimit = $config['search_limit'];
#    echo $from." ".$till." DIFF:."($till - $from)."<br>";
    $eventOptions = [
        'source' => EVENT_SOURCE_TRIGGERS,
        'object' => EVENT_OBJECT_TRIGGER,
        'time_from' => $from,
        'time_till' => $till,
        'output' => ['eventid', 'objectid'],
        'sortfield' => ['clock', 'eventid'],
        'sortorder' => ZBX_SORT_DOWN,
        'limit' => $allEventsSliceLimit + 1
    ];

    $conn = @mysqli_connect($hostLocal, $userLocal, $passLocal,
$dbLocal);

    $connReport = @mysqli_connect($hostLocal, $userLocal,
$passLocal, $dbLocalReport);

    $ArrayObjectID = array();

    $sql_1 = "SELECT DISTINCT objectID FROM events_tickets
WHERE clk_event_create >= '$from' AND clk_event_create <= '$till' ORDER
BY objectID DESC";

    #echo $sql_1."<br>\n";
    $query_1 = mysqli_query($connReport, $sql_1);

```

```

$num_lines_1 = mysqli_num_rows($query_1);

if ($num_lines_1>0){
while ($list_1 = mysqli_fetch_array($query_1)) {
#      echo $list_1['objectID']."\n";
      array_push($ArrayObjectID, $list_1['objectID']);
}
$list_objectID = implode(",", $ArrayObjectID);
#echo $list_objectID."\n";

$row = array();
/*
$arrayHostsName = array();
$arrayInterface = array();
$arrayTriggersTriggerid = array();
$arrayTriggersDescription = array();
$arrayTriggersPriority = array();
*/

#echo "hostid: $hostid <br>";
$sql_2="SELECT distinct hosts.host,hosts.hostid, hosts.name,
interface.ip, interface.dns,
triggers.description, triggers.priority, triggers.triggerid,
zabbix_report.events_tickets.eventid,
zabbix_report.events_tickets.trigger_description,
zabbix_report.events_tickets.clk_event_create,
zabbix_report.events_tickets.clk_event_cleared,
zabbix_report.events_tickets.clk_ack,
zabbix_report.events_tickets.ticket,

zabbix_report.events_tickets.clk_disable,zabbix_report.events_tickets.

```



```

maintenance ,

    zabbix_report.events_tickets.alias,zabbix_report.events_tickets.eventid
    FROM zabbix_report.events_tickets
    INNER JOIN triggers ON
triggers.triggerid=zabbix_report.events_tickets.objectID
    LEFT JOIN functions ON functions.triggerid=triggers.triggerid
    LEFT JOIN items ON items.itemid=functions.itemid
    LEFT JOIN hosts ON hosts.hostid=items.hostid
    LEFT JOIN hosts_groups ON hosts_groups.hostid =
hosts.hostid
    LEFT JOIN interface ON interface.hostid=hosts.hostid
    LEFT JOIN groups ON groups.groupid = hosts_groups.groupid
WHERE groups.internal = '0' AND groups.groupid != '1' ";
if ($hostid>0){
    $sql_2 .= " AND hosts.hostid=".$hostid;
}
if ($groupid>0){
    $sql_2 .= " AND groups.groupid=".$groupid;
}
if ($triggerId>0){
    $sql_2 .= " AND triggers.triggerid=".$triggerId;
}
# $sql_2 .= " AND hosts.status != '1' AND triggers.triggerid in
('.$list_objectID.') AND zabbix_report.events_tickets.clk_event_create
between '$from' AND '$till' ORDER BY hosts.host, triggers.description,
zabbix_report.events_tickets.clk_event_create,zabbix_report.events_tickets.e
ventid ASC"; //Diferente de Host Template
    $sql_2 .= " AND hosts.status != '1' AND triggers.triggerid in

```

```
(".$list_objectID.") AND zabbix_report.events_tickets.clk_event_create  
between '$from' AND '$till' ORDER BY hosts.host,  
zabbix_report.events_tickets.clk_event_create,  
zabbix_report.events_tickets.eventid ASC"; //Diferente de Host Template
```

```
#echo $sql_2."\\n";  
$query_2 = mysqli_query($conn, $sql_2);  
  
$num_lines_2 = mysqli_num_rows($query_2);  
#echo "lines: ".$num_lines_2;  
if ($num_lines_2 > 0) {  
    $index = 0;  
    while ($list_2 = mysqli_fetch_array($query_2)) {  
        if ($list_2['name'] == ""){  
            $HostsName = $list_2['host'];  
        }else{  
            $HostsName = $list_2['name'];  
        }  
        $Interface = $list_2['ip'];  
        $TriggersPriority = $list_2['priority'];  
        $ack_manager="";  
        $UpTime = 0; $DownTime = 0; $UnknownTime = 0;  
        $ack_clock = 0;  
        $ack_user = "";  
        $ack_msg = "";  
        if ( strpos($list_2['ticket'], 'Ticket') !== false){  
            # $ack_clock = $list_2['clk_ack'];  
            # $ack_user = $list_2['alias'];  
        }  
    }  
}
```

```

        #$ack_msg = str_replace ("Ticket:", "", $list_2['ticket']);
        $sql_4 = "SELECT * FROM acknowledges
                LEFT JOIN users ON users.userid =
acknowledges.userid
                WHERE acknowledges.eventid = ".$list_2['eventid']." AND
acknowledges.message like 'Ticket %'
                ORDER by acknowledges.clock DESC LIMIT 1";
        $query_4 = mysqli_query($conn, $sql_4);
        while ($list_4 = mysqli_fetch_array($query_4)) {
            $ack_clock = $list_4['clock'];
            $ack_user = $list_4['alias'];
            $ack_msg = str_replace ("Ticket:", "", $list_4['message']);
        }
    }
    if ( $list_2['clk_event_cleared'] > 0){
        $clk_event_cleared =
zbx_date2str(DATE_TIME_FORMAT_SECONDS,
$list_2['clk_event_cleared']); //Cleared On
        #$alarm_Duration = zbx_date2age($list_2['clk_event_create']
,$list_2['clk_event_cleared']); //Alarm Duration (HH:MM:SS)
        $alarm_Duration = abs($list_2['clk_event_cleared'] -
$list_2['clk_event_create']); // DELTA CLEARED - CREATE
        $alarm_Duration = age2date($alarm_Duration); //Alarm
Duration (HH:MM:SS)
    }else{
        $clk_event_cleared = ""; //Cleared On
        $alarm_Duration = ""; //Alarm Duration (HH:MM:SS)
    }
    if ($ack_clock > 0){

```

```

        #$sopenTicket =
zbx_date2age($list_2['clk_event_create'],$ack_clock); //Time to Trouble
Ticket (HH:MM:SS)

        $sopenTicket = $ack_clock - $list_2['clk_event_create'];
//Time to Trouble Ticket (HH:MM:SS)

        $sopenTicket = age2date($sopenTicket); //Time to
Trouble Ticket (HH:MM:SS)
    }else{
        $sopenTicket = "";
        $ack_msg = ""; //Trouble Ticket ID
        $ack_user = ""; //Acknowledged By
        //$objPHPExcel->setActiveSheetIndex(0)-
>setCellValue("K$rowIndex", ); // Trouble Ticketed By
    }
    $maintenance=$list_2['maintenance'];
    if ( empty ($list_2['maintenance']) ) {
        $maintenance=0;
    }
    if ($csvExport) {
        $csvRows[] = [
            ($pageFilter->hostid == 0) ? $HostsName :
null,
            $Interface,
            zbx_date2str(DATE_TIME_FORMAT_SECONDS,
$list_2['clk_event_create']),
            $clk_event_cleared,
            $list_2['trigger_description'],
            getSeverityName1($list_2['priority']),
            $alarm_Duration,

```

```
        $opendTicket,
        $ack_msg,
            $ack_user,
            $maintenance,
        $list_2['eventid']
    ];
} else {
    $table->addRow([
        ($pageFilter->hostid == 0) ? $HostsName : null,
        $Interface,
        zbx_date2str(DATE_TIME_FORMAT_SECONDS,
$list_2['clk_event_create']),
        $clk_event_cleared,
        $list_2['trigger_description'],
        getSeverityName1($list_2['priority']),
        $alarm_Duration,
        $opendTicket,
        $ack_msg,
            $ack_user,
            $maintenance,
        $list_2['eventid']
    ]);
}
}
}
}
else {
    if (!$csvExport) {
```

```
$events = [];  
  
$url = (new CUrl('events.php'))  
    ->setArgument('fullscreen',  
getRequest('fullscreen'))  
    ->setArgument('groupid', $pageFilter->groupid)  
    ->setArgument('hostid', $pageFilter->hostid);  
  
$paging = getPagingLine($events, ZBX_SORT_UP, $url);  
    }  
}  
  
if ($csvExport) {  
    echo zbx_toCSV($csvRows);  
}  
else {  
    $eventsWidget->addItem($table);  
  
/*    $timeline = [  
        'period' => $period,  
        'starttime' => date(TIMESTAMP_FORMAT, null),  
        'usertime' => date(TIMESTAMP_FORMAT, $till)  
    ];  
  
    $objData = [  
        'id' => 'timeline_1',  
        'loadSBox' => 0,  
        'loadImage' => 0,
```

```

        'loadScroll' => 1,
        'dynamic' => 0,
        'mainObject' => 1
    ];
*/
    #zbx_add_post_js('jqBlink.blink();');
    #zbx_add_post_js('timeControl.addObject("scroll_events_id",
'.zbx_jsvalue($timeline).', '.zbx_jsvalue($objData).');');
    #zbx_add_post_js('timeControl.processObjects();');

    $eventsWidget->show();

    require_once dirname(__FILE__).'/include/page_footer.php';
}

```

17- Alterar o arquivo /usr/share/zabbix/include/menu.inc.php para adicionar o Ticket Report no menu de report do front-end.

```

[
    'url' => 'auditlogs.php',
    'label' => _('Audit'),
    'user_type' => USER_TYPE_SUPER_ADMIN
],
[
    'url' => 'auditacts.php',
    'label' => _('Action log'),
    'user_type' => USER_TYPE_SUPER_ADMIN
],
[
    'url' => 'report4.php',

```

```
'label' => _('Notifications'),  
'user_type' => USER_TYPE_ZABBIX_ADMIN  
],  
[  
    'url' => 'reportTicket.php',  
    'label' => _('Ticket Report'),  
],  
[
```


Capacity

O objetivo desse módulo é exportar dados coletados diariamente: 00:00:00 às 23:59:59, dos parâmetros e hosts configurados.

Os dados exportados serão gravados no diretório /opt/capacity/ZABBIX/ com o formato <HOST>_<AAAAMMDD>.csv.

Exemplo: u630_20170903.csv

Conteúdo do arquivo csv.

```
time;Active virtual pages percent;time;CPU - Total idle time;time;Length of the
run queue;time;Pages page in;time;Pages page out;
03-09-2017 00:00:54;57.0577;03-09-2017 00:04:35;7.4681;03-09-2017
00:04:38;1.0000;03-09-2017 00:00:45;0.0100;03-09-2017 00:01:04;4.0421;
03-09-2017 00:05:54;57.1106;03-09-2017 00:09:35;6.9681;03-09-2017
00:09:38;1.0000;03-09-2017 00:05:45;0.0100;03-09-2017 00:06:04;6.0410;
03-09-2017 00:10:53;56.9544;03-09-2017 00:14:35;5.9791;03-09-2017
00:14:38;0.0000;03-09-2017
```

Configuração

Configuração dos hosts

O relatório de capacity é gerado para todos os hosts que estejam dentro de um dos 3 grupos abaixo:

- Capacity_Linux
- Capacity_Unix
- Capacity_Windows

Configuração do XML

A configuração dos itens que devem ser exportados para o capacity, é configurado nos arquivos XML com o seguinte formato:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<zabbix>
  <groupname> </groupname>
  <items>
    <item>
      <xls_name> </xls_name>
      <zabbix_name> </zabbix_name>
    </item>
  </items>
</zabbix>
```

- groupname: grupo de hosts que contém estes itens para serem exportados
- xls_name: Nome do item que estará no cabeçalho no arquivo CSV.
- zabbix_name: Nome do item (items: name) que está configurado no Zabbix

Instalação

1- criar o arquivo /usr/share/zabbix/report/capacityLinux.xml com o conteúdo abaixo:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<zabbix>
  <groupname>Capacity_Linux</groupname>
  <items>
    <item>
```

```

        <xls_name>Average CPU Utilization %</xls_name>
        <zabbix_name>CPU - Total idle time</zabbix_name>
    </item>
    <item>
        <xls_name>Phisical Memory Utilization %</xls_name>
        <zabbix_name>Memory - Used %</zabbix_name>
    </item>
    <item>
        <xls_name>Pages Paged In</xls_name>
        <zabbix_name>Pages page In</zabbix_name>
    </item>
    <item>
        <xls_name>Pages Paged Out</xls_name>
        <zabbix_name>Pages page Out</zabbix_name>
    </item>
    <item>
        <xls_name>Run Queue Length</xls_name>
        <zabbix_name>CPU Load Average (5 min)</zabbix_name>
    </item>
</items>
</zabbix>

```

2- criar o arquivo /usr/share/zabbix/report/capacityUnix.xml com o conteúdo abaixo:

```

<?xml version="1.0" encoding="ISO-8859-1" ?>
<zabbix>
    <groupname>Capacity_Unix</groupname>
    <items>

```

```

        <item>
            <xls_name>Average CPU Utilization %</xls_name>
            <zabbix_name>CPU - Total idle time</zabbix_name>
        </item>
        <item>
            <xls_name>Comp %</xls_name>
            <zabbix_name>Active virtual pages percent</zabbix_name>
        </item>
        <item>
            <xls_name>Pages Paged In</xls_name>
            <zabbix_name>Pages page in</zabbix_name>
        </item>
        <item>
            <xls_name>Pages Paged Out</xls_name>
            <zabbix_name>Pages page out</zabbix_name>
        </item>
        <item>
            <xls_name>Run Queue Length</xls_name>
            <zabbix_name>Length of the run queue</zabbix_name>
        </item>
    </items>
</zabbix>

```

3- criar o arquivo /usr/share/zabbix/report/capacityWindows.xml com o conteúdo abaixo:

```

<?xml version="1.0" encoding="ISO-8859-1" ?>
<zabbix>
    <groupname>Capacity_Windows</groupname>

```

```

<items>
  <item>
    <xls_name>Average CPU Utilization %</xls_name>
    <zabbix_name>CPU - Processamento</zabbix_name>
  </item>
  <item>
    <xls_name>Physical Memory Utilization %</xls_name>
    <zabbix_name>Memory - Used %</zabbix_name>
  </item>
  <item>
    <xls_name>Pages Paged In</xls_name>
    <zabbix_name>Memory - Pages/sec</zabbix_name>
  </item>
  <item>
    <xls_name>Pages Paged Out</xls_name>
    <zabbix_name>Memory - Pages Output/sec</zabbix_name>
  </item>
  <item>
    <xls_name>Run Queue Length</xls_name>
    <zabbix_name>CPU Run Queue Length</zabbix_name>
  </item>
</items>
</zabbix>

```

4- criar o arquivo

/usr/share/zabbix/report/reportPortoCapacity_Daily_XML_CSV.php com o conteúdo abaixo:

```
<?php
```

```
////////////////////////////////////
// Script Name: reportPortoCapacity_Daily_XML_CSV.php
// Project: NOC Porto Seguro
// Porto Seguro - Ageri
// Resp: Paulo Tetsuo Hoashi      Date: 13/04/17  Desc: Emission
// Function: Dados para capacity diaria - cron executado diariamente
////////////////////////////////////
require_once dirname(__FILE__).'../conf/config_db.php';
require_once dirname(__FILE__).'../includes/phpfunctions.php';
ini_set("memory_limit","512M");

$dir_capacity = "/opt/capacity/ZABBIX/";

$ano = date('Y', strtotime('-1day'));
$mes = date('m', strtotime('-1day'));
$dia = date('d', strtotime('-1day'));

#$ano = 2017;
#$mes = 4;
#$dia = 12;

$var_ini = mktime("00","00","00",$mes,$dia,$ano);
$var_fim = mktime("23","59","59",$mes,$dia,$ano);
#echo $dia."/".$mes."/".$ano."\n";

if (isset($argv[1])){

$file_config = $argv[1];
```

```
#echo $file_config."\n";
if (file_exists( $file_config)){

$arrItemXLS = array();
$arrItemZabbix = array();
$row_item =0;
$xml = simplexml_load_file($file_config) or die("Error");
    $groupname = $xml->groupname;
    foreach($xml->items->item as $item) {
        $arrItemXLS[$row_item] = $item->xls_name;
        $arrItemZabbix[$row_item] = "".$item->zabbix_name."";
        #echo $item->xls_name."\t";
        #echo $item->zabbix_name."\n";
        $row_item++;
    }

if (count($arrItemZabbix) >0){
$list_items = implode(",",$arrItemZabbix);
echo $groupname."\n";
echo "$list_items\n";

$conn = @mysqli_connect($hostLocal, $userLocal, $passLocal, $dbLocal);

$arrayHostsId = array();

$sql_1 = "SELECT hosts.host,hosts.hostid
        FROM hosts
        LEFT JOIN hosts_groups ON hosts_groups.hostid = hosts.hostid
```

```

LEFT JOIN groups ON groups.groupid = hosts_groups.groupid
WHERE groups.internal = '0' AND groups.groupid != '1' AND
groups.name = ".$groupname."
ORDER BY groups.name,hosts.host ASC"; //Diferente de Host
Template
#echo $sql_1."\\n";
$query_1 = mysqli_query($conn, $sql_1);
$num_lines_1 = mysqli_num_rows($query_1);
if ($num_lines_1 >0) {
    $idx =0;
    while ($list_1 = mysqli_fetch_array($query_1)) {
        $arrayHostsName[$idx] = $list_1['host'];
        $arrayHostsId[$idx] = $list_1['hostid'];
        $idx++;
    }
    foreach ( $arrayHostsId as $hostid )    {
        $idx_hostid = array_search($hostid,$arrayHostsId);
        $hostName = $arrayHostsName[$idx_hostid];
        #echo "index: $idx_hostid hostid: $hostid hostname: $hostName
\\n";

        $data_geracao = date("Ymd_His");
        #$nome_arq =
dirname(__FILE__)."/files/".$hostName."_".$ano.$mes.$dia."_".$data_geracao.".xlsx";

        $nome_arq =
$dir_capacity.$hostName."_".$ano.$mes.$dia.".csv";
        #$nome_arq =
"files/Report_".$ano.$mes.$dia."_".$data_geracao.".xlsx";
        #echo "Nome do arquivo: ".$nome_arq."\\n";

```



```

$nome_arq = "/tmp/Report_". $data_geracao. ".xlsx";
$file_csv = fopen($nome_arq, "w") ;
#echo date("H:i:s") , " Inicio: reportPortoCapacity_Daily_XML.php
$nome_arq" , EOL;

$arrayItemsItemid = array();
$arrayItemsValueType = array();
$arrayItemsName = array();

$sql_2 = "SELECT itemid, name, value_type
FROM items
WHERE items.hostid=". $hostid. " AND status != 1 AND
value_type in (0,3)
and name IN (". $list_items. ")
ORDER BY name ASC";
#
echo $sql_2. "\n";
$query_2 = mysqli_query($conn, $sql_2);
$num_lines_2 = mysqli_num_rows($query_2);
if ($num_lines_2 > 0) {
    $idx = 0;
    while ($list_2 = mysqli_fetch_array($query_2)) {
        $arrayItemsName[$idx] = $list_2['name'];
        $arrayItemsItemid[$idx] = $list_2['itemid'];
        $arrayItemsValueType[$idx] = $list_2['value_type'];
        $idx++;
        $txt = "time;". $list_2['name']. ";";
        fwrite($file_csv, $txt);
    }
    fwrite($file_csv, "\n");
}

```

```

#$arrayColumnTime = array();
#$arrayColumnValue = array();
for( $item_idx = 0; $item_idx < $idx ; $item_idx++){
    $itemid = $arrayItemsItemid[$item_idx];
    $itemName = $arrayItemsName[$item_idx];
    $value_type = $arrayItemsValueType[$item_idx];
    #echo "$hostid | $itemid | $value_type
|$itemName \n";

    if ($value_type == 0 ){
        $sql_3 = "SELECT clock,value
                FROM history
                WHERE itemid=".$itemid." AND clock
between ".$var_ini." AND ".$var_fim."
                ORDER BY clock ASC";
    }else{
        $sql_3 = "SELECT clock,value
                FROM history_uint
                WHERE itemid=".$itemid." AND clock between
".$var_ini." AND ".$var_fim."
                ORDER BY clock ASC";
    }
    $arrayTime[$item_idx] = array();
    $arrayValue[$item_idx] = array();

    #echo $sql_3."\n";
    $query_3 = mysqli_query($conn, $sql_3);
    $num_lines_3 = mysqli_num_rows($query_3);
    $idxHistory=2;
    if ($num_lines_3 >0) {

```

```

                                while ($list_3 =
mysqli_fetch_array($query_3)){

                                array_push($arrayTime[$item_idx],date("d-m-Y H:i:s",$list_3['clock']));
                                if ($itemName == 'CPU - Total idle
time'){

                                array_push($arrayValue[$item_idx], 100 - $list_3['value']);
                                }else{

                                array_push($arrayValue[$item_idx], $list_3['value']);
                                }
                                $idxHistory++;
                                }
                                }
                                $max=0;
                                for( $item_idx = 0; $item_idx < $idx ; $item_idx++){
                                    if ($max < count($arrayTime[$item_idx])){
                                        $max = count($arrayTime[$item_idx]);
                                    }
                                }
                                for( $time_idx = 0; $time_idx < $max ; $time_idx++){
                                    for( $item_idx = 0; $item_idx < $idx ; $item_idx++){
                                        if ( $time_idx <
count($arrayTime[$item_idx]) ){
                                            fwrite($file_csv,
$arrayTime[$item_idx][$time_idx].";".$arrayValue[$item_idx][$time_idx].";");
                                        }else{

```

```
                fwrite($file_csv, ";;");
            }
        }
        fwrite($file_csv, "\n");
    }
}

#echo date('H:i:s') , " Write to Excel2007 format" , EOL;
#$callStartTime = microtime(true);
fclose($file_csv);
}

}
#echo date('H:i:s') , " FIM: reportPortoAcompanhamentoTickets.php
$nome_arq " , EOL;
mysqli_close($conn);
}else{
    echo "Nao existem items no arquivo.\n";
}
}else{
    echo "O arquivo: $file_config nao existe.\n";
}
}else{
    echo "php reportCapacity_Dayli_XML.php <nome_arquivo.xml>\n";
}
echo "FIM";

?>
```

- o local de gravação dos arquivos podem ser alterados no parâmetro

\$dir_capacity. Pode ser adicionado um diretório de rede, basta adicionar um ponto de montagem no arquivo /etc/fstab.

5- Adicionar as linhas abaixo no arquivo /etc/crontab:

```
#Relatorio capacity csv
00 1 * * * root /usr/bin/php
/usr/share/zabbix/report/reportPortoCapacity_Daily_XML_CSV.php
/usr/share/zabbix/report/capacityLinux.xml
00 1 * * * root /usr/bin/php
/usr/share/zabbix/report/reportPortoCapacity_Daily_XML_CSV.php
/usr/share/zabbix/report/capacityUnix.xml
00 1 * * * root /usr/bin/php
/usr/share/zabbix/report/reportPortoCapacity_Daily_XML_CSV.php
/usr/share/zabbix/report/capacityWindows.xml
```

Backups

O backup dos servidores estão configurados para os seguintes diretórios:

Servidor	Serviço	Backup
li2454	Zabbix-Server-master	/etc/* /var/log/* /usr/lib/zabbix/*
li2614	Zabbix-Server-slave	/etc/* /var/log/* /usr/lib/zabbix/*
li2455	Web	/etc/* /var/log/* /usr/share/*
li2456	MySQL-Master	/etc/* /var/log/* /usr/lib/zabbix/*
li2457	MySQL-Slave	/etc/* /var/log/* /usr/lib/zabbix/*
li2458	proxy-thpp	/etc/* /var/log/* /usr/lib/zabbix/*
li2459	proxy-windows	/etc/* /var/log/* /usr/lib/zabbix/*
li2460	proxy-nix	/etc/* /var/log/* /usr/lib/zabbix/*

li2461	proxy-snmp	/etc/* /var/log/* /usr/lib/zabbix/*
li2497	proxy-vmware	/etc/* /var/log/* /usr/lib/zabbix/*
i410	proxy-dmz	/etc/* /var/log/* /usr/lib/zabbix/*
LI2615	Zabbix-server + WEB DB	/etc/* /var/log/* /usr/share/* /usr/lib/zabbix/*
LI2616	MySQL DB	/etc/* /var/log/* /usr/lib/zabbix/* /mysql_backup/*
LI2617	proxy-thpp DB	/etc/* /var/log/* /usr/lib/zabbix/*
LI2618	proxy-windows DB	/etc/* /var/log/* /usr/lib/zabbix/*
LI2619	proxy-nix DB	/etc/* /var/log/* /usr/lib/zabbix/*
LI2620	proxy-snmp DB	/etc/* /var/log/*

		/usr/lib/zabbix/*
LI2621	proxy-vmware DB	/etc/* /var/log/* /usr/lib/zabbix/*

Banco de dados

Para realizar o backup da base de dados, é necessário adicionar os scripts abaixo:

Para os servidores LI2456 e LI2457 foram configurados backups apenas das configurações do Zabbix, excluindo as tabelas de histórico de itens e eventos. As configurações FULL do banco de dados está sendo feita no servidor LI2616.

Configuração

LI2456 e LI2457

1- Salvar script no diretório /opt/zabbix/scripts/backup.sh.

```
#!/usr/bin/env bash
#
# NAME
#   zabbix-mysql-dump - Configuration Backup for Zabbix with MySQL
#
# VERSION
#   0.8.2
#
# SYNOPSIS
```



```
# This is a MySQL configuration backup script for Zabbix 1.x, 2.x and 3.0.x.
# It does a full backup of all configuration tables, but only a schema
# backup of large data tables.
#
# The script is based on a script by Ricardo Santos
# (http://zabbixzone.com/zabbix/backuper-only-the-zabbix-configuration/)
#
# CONTRIBUTORS
# - Ricardo Santos
# - Jens Berthold (maxhq)
# - Oleksiy Zagorskyi (zalex)
# - Petr Jendrejovsky
# - Jonathan Bayer
# - Andreas Niedermann (dre-)
# - Mişu Moldovan (dumol)
# - Daniel Schneller (dschneller)
# - Ruslan Ohitin (ruslan-ohitin)
# - Jonathan Wright (neonardo1)
# - msjmeyer
#
# AUTHOR
# Jens Berthold (maxhq), 2016
#
# LICENSE
# This script is released under the MIT License (see LICENSE.txt)
#
# DEFAULT VALUES
```

```
#
# DO NOT EDIT THESE VALUES!
# Instead, use command line parameters or a config file to specify options.
#
#DUMPPDIR="$PWD"
DUMPPDIR="/opt/zabbix/backup"
DBHOST="localhost"
DBNAME="zabbix_db"
DBUSER="root"
DBPASS="F1K@%Cxy6Bhf"
COMPRESSION="gz"
QUIET="no"
REVERSELOOKUP="yes"
GENERATIONSTOKEEP=3

#
# SHOW HELP
#
if [ -z "$1" ]; then
    cat <<EOF
USAGE
    $(basename $BASH_SOURCE) [options]

OPTIONS
    -h HOST
        Hostname/IP of MySQL server.
        Default: $DBHOST

    -d DATABASE
```

Zabbix database name.

Default: \$DBNAME

-u USER

MySQL user to access Zabbix database.

Default: \$DBUSER

-p PASSWORD

MySQL user password (specify "-" for a prompt).

Default: no password

-o DIR

Save Zabbix MySQL dumps to DIR.

Default: \$DUMPPDIR

-c FILE

Use FILE for MySQL options (passed via --defaults-extra-file).

PLEASE NOTE:

mysqldump needs the database to be specified via command line.

So the first "database" options found in the config file is

used for mysqldump.

-r NUM

Rotate backups while keeping up to NUM generations.

Uses filename to match.

Default: keep all backups

-x

Compress using xz instead of gz

PLEASE NOTE:

xz compression will take much longer and consume more CPU time but the resulting backup will be about half the size of the same sql file compressed using gz. Your mileage may vary.

-0

Do not compress the sql dump

-n

Skip reverse lookup of IP address for host.

-q

Quiet mode: no output except for errors (for batch/crontab use).

EXAMPLES

```
$(basename $BASH_SOURCE) -h 1.2.3.4 -d zabbixdb -u zabbix -p test
```

```
$(basename $BASH_SOURCE) -u zabbix -p - -o /tmp
```

```
$(basename $BASH_SOURCE) -c /etc/mysql/mysql.cnf
```

```
$(basename $BASH_SOURCE) -c /etc/mysql/mysql.cnf -d zabbixdb
```

EOF

```
exit 1
```

fi

#

PARSE COMMAND LINE ARGUMENTS

#

DB_GIVEN=0

```
while getopts ":h:d:u:p:o:r:c:x0qn" opt; do
```

```
case $opt in
  h) DBHOST="$OPTARG" ;;
  d) DBNAME="$OPTARG"; DB_GIVEN=1 ;;
  u) DBUSER="$OPTARG" ;;
  p) DBPASS="$OPTARG" ;;
  c) CNFFILE="$OPTARG" ;;
  o) DUMPPDIR="$OPTARG" ;;
  r) GENERATIONSTOKEEP=$(printf "%.0f" "$OPTARG") ;;
  x) COMPRESSION="xz" ;;
  0) COMPRESSION="" ;;
  n) REVERSELOOKUP="no" ;;
  q) QUIET="yes" ;;
  \?) echo "Invalid option: -$OPTARG" >&2; exit 1 ;;
  :) echo "Option -$OPTARG requires an argument" >&2; exit 1 ;;
esac
done

# Password prompt
if [ "$DBPASS" = "-" ]; then
  read -s -p "Enter MySQL password for user '$DBUSER' (input will be
hidden): " DBPASS
  echo ""
fi

# Config file validations
if [ ! -z "$CNFFILE" ]; then
  if [ ! -r "$CNFFILE" ]; then
    echo "ERROR: Cannot read configuration file $CNFFILE" >&2
    exit 1
  fi
fi
```

```
fi
# Database name needs special treatment:
# For mysqldump it has to be specified on the command line!
# Therefore we need to get it from the config file
if [ $DB_GIVEN -eq 0 ]; then
    DBNAME=$(grep -m 1 ^database= "$CNFFILE" | cut -d= -f2)
fi
fi

#
# CONSTANTS
#
SUFFIX=""; test ! -z $COMPRESSION && SUFFIX=".${COMPRESSION}"

MYSQL_OPTS=()
[ ! -z "$CNFFILE" ] && MYSQL_OPTS=("${MYSQL_OPTS[@]}" --defaults-
extra-file="$CNFFILE")
[ ! -z "$DBHOST" ] && MYSQL_OPTS=("${MYSQL_OPTS[@]}" -h $DBHOST)
[ ! -z "$DBUSER" ] && MYSQL_OPTS=("${MYSQL_OPTS[@]}" -u $DBUSER)
[ ! -z "$DBPASS" ] && MYSQL_OPTS=("${MYSQL_OPTS[@]}" -
p"$DBPASS")

MYSQL_OPTS_BATCH=("${MYSQL_OPTS[@]}" --batch --silent)
[ ! -z "$DBNAME" ] &&
MYSQL_OPTS_BATCH=("${MYSQL_OPTS_BATCH[@]}" -D $DBNAME)

# Log file for errors
ERRORLOG=$(mktemp)
```

```
# Host name: try reverse lookup if IP is given
DBHOSTNAME="$DBHOST"
command -v dig >/dev/null 2>&1
FIND_DIG=$?
if [ "$REVERSELOOKUP" == "yes" -a $FIND_DIG -eq 0 ]; then
    # Try resolving a given host ip
    newHostname=$(dig +noall +answer -x $DBHOST | sed -r
's/((\S+)\s+)([^\s.]+)\s+.*\3/')
    test ! -z "$newHostname" && DBHOSTNAME="$newHostname"
fi

#
# CONFIG DUMP
#
if [ "$QUIET" == "no" ]; then
    cat <<-EOF
    Configuration:
    - host:    $DBHOST ($DBHOSTNAME)
    - database: $DBNAME
    - user:    $DBUSER
    - output:  $DUMPPDIR

EOF
fi

#
# FUNCTIONS
#
```

```

# Returns TRUE if argument 1 is part of the given array (remaining
arguments)
elementIn () {
    local e
    for e in "${@:2}"; do [[ "$e" == "$1" ]] && return 0; done
    return 1
}

#
# CHECKS
#
if [ ! -x /usr/bin/mysqldump ]; then
    echo "mysqldump not found." >&2
    echo "(with Debian, \"apt-get install mysql-client\" will help)" >&2
    exit 1
fi

#
# READ TABLE LIST from __DATA__ section at the end of this script
# (http://stackoverflow.com/a/3477269/2983301)
#
DATA_TABLES=()
while read line; do
    table=$(echo "$line" | cut -d" " -f1)
    echo "$line" | cut -d" " -f5 | grep -qi "DATA"
    test $? -eq 0 && DATA_TABLES+=($table)
done <<(sed '0,/^__DATA__$/d' "$BASH_SOURCE" | tr -s " ")

# paranoid check

```



```

if [ ${#DATA_TABLES[@]} -lt 5 ]; then
    echo "ERROR: The number of large data tables configured in this script is
less than 5." >&2
    exit 1
fi

#
# BACKUP
#
# Read table list from database
[ "$QUIET" == "no" ] && echo "Fetching list of existing tables..."
DB_TABLES=$(mysql "${MYSQL_OPTS_BATCH[@]}" -e "SELECT
table_name FROM information_schema.tables WHERE table_schema =
'$DBNAME'" 2>$ERRORLOG)
if [ $? -ne 0 ]; then echo "ERROR while trying to access database:" 2>&1; cat
$ERRORLOG 2>&1; exit 1; fi
DB_TABLES=$(echo "$DB_TABLES" | sort)
DB_TABLE_NUM=$(echo "$DB_TABLES" | wc -l)

# Query Zabbix database version
VERSION=""
DB_VER=$(mysql "${MYSQL_OPTS_BATCH[@]}" -N -e "select optional from
dbversion;" 2>/dev/null)
if [ $? -eq 0 ]; then
    # version string is like: 02030015
    re='(.*)([0-9]{2})([0-9]{4})'
    if [[ $DB_VER =~ $re ]]; then
        VERSION="_db-${BASH_REMATCH[1]}.${(( ${BASH_REMATCH[2]} + 0
)).${(( ${BASH_REMATCH[3]} + 0 ))}"
    fi
fi

```

```
fi
fi

# Assemble file name
DUMPFILENAME_PREFIX="zabbix_cfg_${DBHOSTNAME}"
DUMPFILBASE="${DUMPFILENAME_PREFIX}_${date +%Y%m%d-%H%M}%${VERSION}.sql"
DUMPFIL="${DUMPDIR}/${DUMPFILBASE}"

PROCESSED_DATA_TABLES=()
i=0

mkdir -p "${DUMPDIR}"

[ "$QUIET" == "no" ] && echo "Starting table backups..."
while read table; do
    # large data tables: only store schema
    if elementIn "$table" "${DATA_TABLES[@]}"; then
        dump_opt="--no-data"
        PROCESSED_DATA_TABLES+=($table)
    # configuration tables: full dump
    else
        dump_opt="--extended-insert=FALSE"
    fi

    mysqldump "${MYSQL_OPTS[@]}" \
        --routines --opt --single-transaction --skip-lock-tables \
        $dump_opt \
        $DBNAME --tables ${table} >> "$DUMPFIL" 2>$ERRORLOG
```

```

    if [ $? -ne 0 ]; then echo -e "\nERROR: Could not backup table ${table}:"
    >&2; cat $ERRORLOG >&2; exit 1; fi

    if [ "$QUIET" == "no" ]; then
        # show percentage
        i=$((i+1)); i_percent=$((($i * 100 / $DB_TABLE_NUM))
        if [ $($i_percent % 12) -eq 0 ]; then
            echo -n "${i_percent}%"
        else
            if [ $($i_percent % 2) -eq 0 ]; then echo -n "."; fi
        fi
    fi
done <<<"$DB_TABLES"

rm $ERRORLOG

#
# COMPRESS BACKUP
#
if [ "$QUIET" == "no" ]; then
    echo -e "\n"
    echo "For the following large tables only the schema (without data) was
stored:"
    for table in "${PROCESSED_DATA_TABLES[@]}"; do echo " - $table";
done

    echo
    echo "Compressing backup file..."

```

```
fi

EXITCODE=0
if [ "$COMPRESSION" == "gz" ]; then gzip -f "$DUMPFIL" ; EXITCODE=$?;
fi
if [ "$COMPRESSION" == "xz" ]; then xz -f "$DUMPFIL" ; EXITCODE=$?; fi
if [ $EXITCODE -ne 0 ]; then
    echo -e "\nERROR: Could not compress backup file, see previous
messages" >&2
    exit 1
fi

[ "$QUIET" == "no" ] && echo -e "\nBackup
Completed:\n${DUMPFIL}${SUFFIX}"

#
# ROTATE OLD BACKUPS
#
if [ $GENERATIONSTOKEEP -gt 0 ]; then
    [ "$QUIET" == "no" ] && echo "Removing old backups, keeping up to
$GENERATIONSTOKEEP"
    REMOVE_OLD_CMD="cd \"$DUMPDIR\" && ls -t
\"${DUMPFILNAME_PREFIX}\"* | /usr/bin/awk
\"NR>${GENERATIONSTOKEEP}\" | xargs rm -f "
    eval ${REMOVE_OLD_CMD}
    if [ $? -ne 0 ]; then
        echo "ERROR: Could not rotate old backups" >&2
        exit 1
    fi
fi
```

```
fi

exit 0

#####
#####
# List of all known table names and a flag indicating data (=large) tables
#

__DATA__
acknowledges      1.3.1 - 3.0.3 DATA
actions           1.3.1 - 3.0.3
alerts            1.3.1 - 3.0.3 DATA
application_discovery  2.5.0 - 3.0.3
application_prototype  2.5.0 - 3.0.3
application_template  2.1.0 - 3.0.3
applications       1.3.1 - 3.0.3
auditlog           1.3.1 - 3.0.3 DATA
auditlog_details   1.7 - 3.0.3 DATA
autoreg            1.3.1 - 1.3.4
autoreg_host       1.7 - 3.0.3
conditions         1.3.1 - 3.0.3
config             1.3.1 - 3.0.3
dbversion          2.1.0 - 3.0.3
dchecks            1.3.4 - 3.0.3
dhosts             1.3.4 - 3.0.3
drules             1.3.4 - 3.0.3
dservices          1.3.4 - 3.0.3
escalations        1.5.3 - 3.0.3
```

events	1.3.1	- 3.0.3	DATA
expressions	1.7	- 3.0.3	
functions	1.3.1	- 3.0.3	
globalmacro	1.7	- 3.0.3	
globalvars	1.9.6	- 3.0.3	
graph_discovery	1.9.0	- 3.0.3	
graph_theme	1.7	- 3.0.3	
graphs	1.3.1	- 3.0.3	
graphs_items	1.3.1	- 3.0.3	
group_discovery	2.1.4	- 3.0.3	
group_prototype	2.1.4	- 3.0.3	
groups	1.3.1	- 3.0.3	
help_items	1.3.1	- 2.1.8	
history	1.3.1	- 3.0.3	DATA
history_log	1.3.1	- 3.0.3	DATA
history_str	1.3.1	- 3.0.3	DATA
history_str_sync	1.3.1	- 2.2.13	DATA
history_sync	1.3.1	- 2.2.13	DATA
history_text	1.3.1	- 3.0.3	DATA
history_uint	1.3.1	- 3.0.3	DATA
history_uint_sync	1.3.1	- 2.2.13	DATA
host_discovery	2.1.4	- 3.0.3	
host_inventory	1.9.6	- 3.0.3	
host_profile	1.9.3	- 1.9.5	
hostmacro	1.7	- 3.0.3	
hosts	1.3.1	- 3.0.3	
hosts_groups	1.3.1	- 3.0.3	
hosts_profiles	1.3.1	- 1.9.2	
hosts_profiles_ext	1.6	- 1.9.2	

hosts_templates	1.3.1	- 3.0.3
housekeeper	1.3.1	- 3.0.3
httpstep	1.3.3	- 3.0.3
httpstepitem	1.3.3	- 3.0.3
httptest	1.3.3	- 3.0.3
httptestitem	1.3.3	- 3.0.3
icon_map	1.9.6	- 3.0.3
icon_mapping	1.9.6	- 3.0.3
ids	1.3.3	- 3.0.3
images	1.3.1	- 3.0.3
interface	1.9.1	- 3.0.3
interface_discovery	2.1.4	- 3.0.3
item_application_prototype	2.5.0	- 3.0.3
item_condition	2.3.0	- 3.0.3
item_discovery	1.9.0	- 3.0.3
items	1.3.1	- 3.0.3
items_applications	1.3.1	- 3.0.3
maintenances	1.7	- 3.0.3
maintenances_groups	1.7	- 3.0.3
maintenances_hosts	1.7	- 3.0.3
maintenances_windows	1.7	- 3.0.3
mappings	1.3.1	- 3.0.3
media	1.3.1	- 3.0.3
media_type	1.3.1	- 3.0.3
node_cksum	1.3.1	- 2.2.13
node_configlog	1.3.1	- 1.4.7
nodes	1.3.1	- 2.2.13
opcommand	1.9.4	- 3.0.3
opcommand_grp	1.9.2	- 3.0.3

opcommand_hst	1.9.2	- 3.0.3
opconditions	1.5.3	- 3.0.3
operations	1.3.4	- 3.0.3
opgroup	1.9.2	- 3.0.3
opinventory	3.0.0	- 3.0.3
opmediatypes	1.7	- 1.8.22
opmessage	1.9.2	- 3.0.3
opmessage_grp	1.9.2	- 3.0.3
opmessage_usr	1.9.2	- 3.0.3
optemplate	1.9.2	- 3.0.3
profiles	1.3.1	- 3.0.3
proxy_autoreg_host	1.7	- 3.0.3
proxy_dhistory	1.5	- 3.0.3
proxy_history	1.5.1	- 3.0.3
regexps	1.7	- 3.0.3
rights	1.3.1	- 3.0.3
screen_user	3.0.0	- 3.0.3
screen_usrgrp	3.0.0	- 3.0.3
screens	1.3.1	- 3.0.3
screens_items	1.3.1	- 3.0.3
scripts	1.5	- 3.0.3
service_alarms	1.3.1	- 3.0.3
services	1.3.1	- 3.0.3
services_links	1.3.1	- 3.0.3
services_times	1.3.1	- 3.0.3
sessions	1.3.1	- 3.0.3
slides	1.3.4	- 3.0.3
slideshow_user	3.0.0	- 3.0.3
slideshow_usrgrp	3.0.0	- 3.0.3

slideshows	1.3.4	- 3.0.3
sysmap_element_url	1.9.0	- 3.0.3
sysmap_url	1.9.0	- 3.0.3
sysmap_user	3.0.0	- 3.0.3
sysmap_usrgrp	3.0.0	- 3.0.3
sysmaps	1.3.1	- 3.0.3
sysmaps_elements	1.3.1	- 3.0.3
sysmaps_link_triggers	1.5	- 3.0.3
sysmaps_links	1.3.1	- 3.0.3
timeperiods	1.7	- 3.0.3
trends	1.3.1	- 3.0.3 DATA
trends_uint	1.5	- 3.0.3 DATA
trigger_depends	1.3.1	- 3.0.3
trigger_discovery	1.9.0	- 3.0.3
triggers	1.3.1	- 3.0.3
user_history	1.7	- 2.4.8
users	1.3.1	- 3.0.3
users_groups	1.3.1	- 3.0.3
usrgrp	1.3.1	- 3.0.3
valuemaps	1.3.1	- 3.0.3

2- Adicionar a linha abaixo no arquivo /etc/crontab.

```
00 1 * * * root /opt/zabbix/scripts/backup.sh -o /opt/zabbix/backup
```

3- adicionar permissão de execução no script de backup

```
chmod +x /opt/zabbix/scripts/backup.sh
```

LI2616

1- criar o script de backup /etc/zabbix/scripts/backup_ALL.sh com o conteúdo abaixo:

```
#!/bin/bash
DUMPPDIR="/mysql_backup"
DUMPPFILE="${DUMPPDIR}/mysql-dump-$(date "+%d%m%y-%H%M.sql.gz")"
DBHOST="localhost"
DBUSER="root"
DBPASS="F1K@%Cxy6Bhf"
DELETE_OLD="15"

mysqldump -u${DBUSER} -p${DBPASS} --all-databases --single-transaction |
gzip > ${DUMPPFILE}

find ${DUMPPDIR} -type f -mtime +${DELETE_OLD} -exec rm {} \;
```

2- Adicionar a linha abaixo no arquivo /etc/crontab.

```
00 23 * * 5 /etc/zabbix/scripts/backup_ALL.sh
```

3- adicionar permissão de execução no script de backup

```
chmod +x /etc/zabbix/scripts/backup_ALL.sh
```

Apêndice

- A - Instalação agente Zabbix Linux THPP**
- B - Instalação agente Zabbix Linux Produção**
- C - Instalação agente Zabbix Linux DMZ**
- D - Instalação agente Zabbix Windows THPP**
- E - Instalação agente Zabbix Windows Produção**
- F - Instalação agente Zabbix Windows DMZ**
- G - Instalação agente Zabbix Unix THPP**
- H - Instalação agente Zabbix Unix Produção**
- I - Instalação agente Zabbix HP-UX THPP**
- J - Instalação agente Zabbix HP-UX Produção**
- K - Instalação agente Zabbix Xenserver Produção**
- L- Troubleshooting agente Zabbix Linux**
- M- Troubleshooting agente Zabbix Windows**
- N- Troubleshooting agente Zabbix Unix**
- O- Failover Zabbix Proxy THPP**
- P- Failover Zabbix Proxy NIX**
- Q- Failover Zabbix Proxy Windows**
- R- Failover Zabbix Proxy SNMP**
- S- Failover Zabbix Proxy VMWARE**
- T- Failover Zabbix WEB**
- U- Failover Cluster Zabbix Server**
- V- Failover Cluster MySQL**
- W- Planilha Hosts x Interfaces**

X- Planilha Hosts x Groups

Y- Planilha Hosts x Templates

Z- Planilha Templates Itens e Triggers

AA- Planilha Host x itens avulsos

AB- Planilha check WEB

AC- Planilha LLD Filters Custom

AD- Planilha Macros

AE- Planilha Regexp

AF- Planilha Value Mappings