

Custom Script Template - XYMon

Does it require Sudo? Grany XYMon PWDless Sudo Access

```
sudo visudo && sudo service sudo restart
```

```
xymon ALL=(ALL) NOPASSWD: /path/2/binary
```

OR

note: There IS a space when you do NOPASSWD: /path2/binary, but there IS NOT a space when you do NOPASSWD:ALL

```
xymon ALL=(ALL) NOPASSWD:ALL
```

Test XYMon Sudo Access:

```
su -s /bin/sh xymon  
sudo /path/2/binary
```

UBUNTU

Configure Script REPLACE [SERVICE]

CLIENT SIDE SCRIPT

```

cat << 'EOL' >/usr/lib/xymon/client/ext/[SERVICE].sh
#!/bin/sh

COLUMN=[SERVICE]
COLOR=green
MSG="Testing Command ($var)"

#In this test, if var = empty = RED
var="$(sudo /usr/StorMan/arcconf GETCONFIG 1 LD 0 | grep Optimal)"
var1="$(sudo /usr/StorMan/arcconf GETCONFIG 1 LD 1 | grep Optimal)"
cmd="$(sudo /usr/StorMan/arcconf GETCONFIG 1)"

if [ -z "$var" ] || [ -z "$var1" ]

then

COLOR=red
MSG="{MSG}"
FAILED
" "
"${cmd}"
"
else

MSG="{MSG}"
SUCCESS
" "
"${cmd}"
"
fi

# Leave the rest of script alone
# Tell Xymon about it
$XYMON $XYMSRV "status $MACHINE.$COLUMN $COLOR `date`

${MSG}
"

exit 0

EOL

```

SERVER SIDE SCRIPT

```

cat << 'EOL' >/usr/lib/xymon/client/ext/[SERVICE].sh
#!/bin/sh

#This tag goes in hosts.cfg
HOSTTAG=[SERVICE]
COLUMN=$HOSTTAG

##XYMon Configs -- Leave Alone##
$XYMONHOME/bin/xymongrep $HOSTTAG | while read L
do
    set $L          # To get one line of output from xymongrep

    HOSTIP="$1"
    MACHINEDOTS="$2"
    MACHINE=`echo $2 | $SED -e's/\./,/g'`

    COLOR=green
    MSG="$HOSTTAG status for host $MACHINEDOTS"

##Begin Custom If/Then Script##

#In this test, if var = empty = RED
var="$(sudo /usr/StorMan/arcconf GETCONFIG 1 LD 0 | grep Optimal)"
var1="$(sudo /usr/StorMan/arcconf GETCONFIG 1 LD 1 | grep Optimal)"
cmd="$(sudo /usr/StorMan/arcconf GETCONFIG 1)"

if [ -z "$var" ] || [ -z "$var1" ]

then

COLOR=red
MSG="{MSG}
FAILED
"
"$cmd"
"
else

MSG="{MSG}
SUCCESS
"
"$cmd"
"
fi

# Leave the rest of script alone
# Tell Xymon about it
$XYMON $XYMSRV "status $MACHINE.$COLUMN $COLOR `date`

${MSG}
"
done

exit 0
EOL

```

```

sudo chown xymon:xymon /usr/lib/xymon/client/ext/[SERVICE].sh
sudo chmod 777 /usr/lib/xymon/client/ext/[SERVICE].sh

```

```
cat << 'EOL' >/etc/xymon/clientlaunch.d/[SERVICE].cfg
[[SERVICE]]
ENVFILE $XYMONCLIENTHOME/etc/xymonclient.cfg
CMD $XYMONCLIENTHOME/ext/[SERVICE].sh
LOGFILE $XYMONCLIENTHOME/logs/[SERVICE].log
INTERVAL 15m
EOL
```

```
sudo chmod 777 /etc/xymon/clientlaunch.d/*
sudo chown xymon:xymon /etc/xymon/clientlaunch.d/*
```

```
/usr/lib/xymon/client/bin/xymoncmd /usr/lib/xymon/client/ext/[SERVICE].sh
```

Restart Service :

```
sudo /etc/init.d/xymon-client restart
```

On Server :

```
sudo /etc/init.d/xymon restart && sudo /etc/init.d/apache2 restart
```

CentOS:

```

cat << 'EOL' >/usr/share/xymon-client/ext/[SERVICE].sh
#!/bin/sh

COLUMN=[SERVICE]
COLOR=green
MSG="timedatectl | grep {TimeZone,[SERVICE] Enabled,[SERVICE] Synchronized}"

#In this test, if var = empty = RED
var="$(timedatectl | grep "Timezone: America/New_York")"
var1="$(timedatectl | grep "[SERVICE] enabled: yes")"
var2="$(timedatectl | grep "[SERVICE] synchronized: yes")"
cmd="$(timedatectl && [SERVICE]q -p)"

if [ -z "$var" ] || [ -z "$var1" ] || [ -z "$var2" ]

then

COLOR=red
MSG="{MSG}"
FAILED
" "
"${cmd}"
"
else

MSG="{MSG}"
SUCCESS
" "
"${cmd}"
"
fi

# Leave the rest of script alone
# Tell Xymon about it
$XYMON $XYMSRV "status $MACHINE.$COLUMN $COLOR `date`"

${MSG}
"

exit 0
EOL

chown xymon:xymon /usr/share/xymon-client/ext/[SERVICE].sh
chmod 777 /usr/share/xymon-client/ext/[SERVICE].sh

cat << 'EOL' >/etc/xymon-client/client.d/[SERVICE].cfg
[[SERVICE]]
ENVFILE $XYMONCLIENTHOME/etc/xymonclient.cfg
CMD $XYMONCLIENTHOME/ext/[SERVICE].sh
LOGFILE $XYMONCLIENTHOME/logs/[SERVICE].log
INTERVAL 15m
EOL

chown xymon:xymon /etc/xymon-client/client.d/[SERVICE].cfg
chmod 777 /etc/xymon-client/client.d/[SERVICE].cfg

/usr/share/xymon-client/bin/xymoncmd /usr/share/xymon-client/ext/[SERVICE].sh

/etc/init.d/xymon-client restart

```