Check_Eventwatch

This check is used to read ALL events from ALL Eventlogs and return all warnings and errors. It can use a Blacklist to ignore events that are not wanted to be reported. Its possible to create temporary blacklist which will expire after a predefined timeframe. If you want to add new Events to the Blacklist easily take a look at -CreateTempBlacklist and -ConvertTempBlacklists2xml. They work very well Together. Usage: use -help switch for more details - default path for blacklist and configuration files is C:\ProgramData\icinga2\var\EventWatch_config

```
<#

V1.2 - Rafael Voss - 25.08.2017
Source: it-wiki.eu
License: GPLv3


Based on EventWatch.ps1 (c) from  c't/Peter Siering, released under GPLv3
More information for original Eventwatchscript can be found in c't 10/2012,
S. 148 (Version 1)
and c't 13/14, S. ??? (Version 1.1)


Description:
This check is used to read ALL events from ALL Eventlogs and return all
warnings and errors. It can use a Blacklist to ignore events that are not
wanted to be reported.
Its possible to create temporary blacklists, which will expire after a
predefined timeframe or/and create a localblacklist from thhese temp lists
to save them locally.
Usage: use -help switch for more details - default path for blacklist and
configuration files is C:\ProgramData\icinga2\var\EventWatch_config and can
be changed in the "Settings" section.
#>

Param(
    [switch]$LongOutput,
    [string]$EventAge = 24,
    [string]$TempBlacklistExpire = 24,
    [switch]$AddTempBlacklists2localBlacklist,
    [switch]$RemoveTempBlacklists,
    [switch]$CreateTempBlacklist,
    [switch]$IgnoreWmiErrors,
    [switch]$Help
)

####################
#     Settings     #
```

```
####################

#Set configuration location.
$configpath = 'C:\ProgramData\icinga2\var\EventWatch_config'

# Online event Blacklist, f.e. http://mycompany.com/blacklist.xml
# If not set, local blacklist is used
$blacklisturl = ""

#Icinga Status, when errors are found
$exitcode = 1 #1=Warning, 2=Critical, 3 = Unknown



#MAINSCRIPT - no changed needed after this point
#creating folder for configuration and temporarly files, like temp blacklist

if ($Help -or (!(test-path $configpath))) {
write-host "
Check_Eventwatch.ps1 - V1.1 - last modified by Rafael Voss

Before the check is executed the first time, you need to move your global
Eventwatch-blacklist.txt into $configpath,
otherwise your Blacklist will not be recognized and you will get this help
text over and over again.
If no Blacklist is needed, the folder still needs to be created for other
configuration files.
All Events that are in the Blacklist will be ignored by the check.

Switches:
-EventAge : sets how long the check should go back in time to look for
events (in hours). Defaults to 24h
-longOutput : set the output to full output. The complete event message will
send back to Icinga
-createTempBlacklist :- Creates a new temporary blacklist that will expire
after some hours
  (default = 24h, can be changed with -tempblacklistExpire parameter)
-tempblacklistexpire : set the expire date in hours of temp blacklists
-convertTempBlacklists2xml - converts all existing temporary blacklisted
event id's to a new blacklist named new_events.txt
  This new file can be used to easily add the events to your blacklist, or
for using the information for research.
-AddTempBlacklists2localBlacklist - Adds all temporary blacklists to the
permanent local blacklist: $configpath\Eventwatch-local-blacklist.txt
-ignoreWmiErrors _ This switch can be used to ignore Errors on fetching
Eventlogs.
  This can be usefull if you don't have permissions for all Eventlogs and
you can't fix this because of policy
-RemoveTempBlacklists - Removes the blacklist and all content of
'$configpath\temporary_blacklists\'
-help - Displays this helptext
```

```
Troubleshooting:
- If you get 'Get-WinEvent -ListLog' Errors, that is mostly because the
Icingsa Service User has no permissions to read the log, you can use -
ignoreErrors to ignore it.
- This Check is a little time an CPU intensive, maybe you Icinga Check
Timeout is to low. Slowest experience so far 32 seconds.

Hints:
- If you want to add new events to the blacklist easily take a look at -
CreateTempBlacklist and -ConvertTempBlacklists2xml. They work very well
Together.
  First create a tempblacklist, than convert this list to a xml like format.
Open the created txt file and copy and paste the strings to your Blacklist.
  You will get all needed file locations when you are using the -
ConvertTempBlacklists2xml switch.
- If you want to get rid of the Temporary Blacklists, just remove the folder
$configpath\temporary_blacklists.
" -ForegroundColor gray
exit
}


#FUNCTIONS
#Add temporary Blacklists to local-blacklist
function Add-TempBlacklist2LocalBlacklist {
  write-host "Adding temporary blacklists from
$configpath\temporary_blacklists\* to $configpath\Eventwatch-local-
blacklist.xml and removing all temp blacklists"
  #$events = Get-Content "$configpath\temporary_blacklists\*" | out-file
"$configpath\Eventwatch-local-blacklist.txt" -append
  if (test-path $configpath\Eventwatch-local-blacklist.xml) {
    $events = import-clixml  "$configpath\Eventwatch-local-blacklist.xml"
  }
  $events += Import-Clixml "$configpath\temporary_blacklists\*"
  $events | export-clixml  "$configpath\Eventwatch-local-blacklist.xml" #-
append
  Remove-Item $configpath\temporary_blacklists\*
}

#copy temp blacklists to local black list and remove added temp bölacklists.
After that exit the script
if ($psboundparameters.count -eq 1 -And $AddTempBlacklists2localBlacklist) {
Add-TempBlacklist2LocalBlacklist; exit }

#remove all temporary blacklists
if ($RemoveTempBlacklists) {
  write-host "removing temporary blacklists and exiting script"
  remove-item $configpath\temporary_blacklists\*
  exit
}
```

```
if (!(test-path $configpath\temporary_blacklists)) {
    mkdir $configpath\temporary_blacklists | Out-Null
}

# get online blacklist, if defined, otherwise use local blacklist
if ( $blacklisturl -ne "") {
  try {
    $wget= New-Object System.Net.Webclient
    $wget.DownLoadFile($blacklisturl,"$configpath\Eventwatch-blacklist.xml")
  }
  catch {
    Write-Warning "Error fetching blacklist"
  }
}

#get global blacklist
try {
  #$blacklist= Get-Content "$configpath\Eventwatch-blacklist.txt"
  $blacklist= import-clixml "$configpath\Eventwatch-blacklist.xml"
}
catch {
  $blacklist = @()
}

#add local blacklist
try {
  #$blacklist = $blacklist + (Get-Content "$configpath\Eventwatch-local-
blacklist.txt" -ErrorAction SilentlyContinue)
  $blacklist = $blacklist + (import-clixml "$configpath\Eventwatch-local-
blacklist.xml" -ErrorAction SilentlyContinue)
}
catch {
  #DEBUGCODE no local blacklist found"
}

# get unixtime
$startdate= Get-Date
$date1 = Get-Date -Date "01/01/1970"
$unixHours = [INT](New-TimeSpan -Start $date1 -End $startdate).TotalHours


#get temporary blacklist
$tempBlacklist = @()
try {
    $tempblacklists = Get-ChildItem "$configpath\temporary_blacklists\"
} catch {
    $tempBlacklist = @()
}

foreach ($tempblacklistfilename in $tempblacklists) {
```

```
  #write-host "Tempblacklist: $tempBlacklistfilename"
  if ($tempblacklistfilename.Name -gt $unixHours-24) {
    #write-host "FULLNAME:$($tempblacklistfilename.FullName)"
    #$tempBlacklist = $tempBlacklist + (get-content
$tempblacklistfilename.FullName)
    #$tempBlacklist += $tempBlacklist + (get-content
$tempblacklistfilename.FullName)
    $tempBlacklist += Import-Clixml "$configpath\temporary_blacklists\*"
  }
}

$lasttimestamp = $startdate-(New-TimeSpan -hour $EventAge)

#query critical events (1) and errors (2)
$eventcount=0
$msgs=@()
if ($IgnoreWmiErrors) {
  $lognames = Get-WinEvent -ListLog * -ErrorAction SilentlyContinue
} else {
  $lognames = Get-WinEvent -ListLog *
}
foreach ( $level in @(1, 2, 3)) {
  foreach ($logname in $lognames) {
    # Ereignisprotokoll(e) auslesen
    $events= Get-WinEvent -FilterHashTable @{
    LogName = $logname.LogName; level = $level; Starttime=$lasttimestamp } -
MaxEvents 30000 -EA SilentlyContinue
    $eventcount+= ($events | Measure-Object).Count
    # Weiterverarbeitung nur, wenn Nachrichten vorliegen
    if ( ($events | Measure-Object).Count -gt 0 ) {
      foreach ( $event in $events) {
        if ($LongOutput) {
          $msg=$event.Message
        } else {
          $msg= ""
        }
        $logname=$event.LogName
        if ( $logname) {
          $logname=$logname -ireplace( "Microsoft-Windows-", "")
        }
        $provname=$event.ProviderName
        if ( $provname) {
          $provname=$provname -ireplace( "Microsoft-Windows-", "")
        }
        if ( $logname.Contains( $provname)) {
          $logname=""
        }
        if (( $provname -eq $logname) -or ( $logname -eq "")) {
          $msg= ( $provname+ " - "+ $msg + "("+ $event.Id+ ") *")
        } else {
          $msg= ( $logname + "/"+ $provname+ " - "+ $msg + "("+ $event.Id+
```

```
") *")
        }
        # Check against global blacklist
        foreach ( $line in $blacklist) {
          if (!( $line.StartsWith("#") )) {
            if ( $msg -like $line) {
              $msg=""
              $eventcount--
              break
            }
          }
        }
         #check against local blacklist
        foreach ( $line in $localblacklist) {
          if (!( $line.StartsWith("#") )) {
            if ( $msg -like $line) {
              $msg=""
              $eventcount--
              break
            }
          }
        }
        # Gegen Blacklist pruefen
        if (!($tempBlacklist -eq "")) {
          foreach ( $line in $tempBlacklist) {
            if (!( $line.StartsWith("#") )) {
              if ( $msg -like $line) {
                $msg=""
                $eventcount--
                break
              }
            }
          }
        }
        if ( $msg -ne "") {
          $msgs+= $msg
        }
      }
    } else {
      $msg=""
    }
  }
}

#create temporary blacklist to blacklist all existing events (warning/error)
if ($CreateTempBlacklist) {
  $msgs = $msgs | select -Unique
  #$msgs| out-file $configPath\temporary_blacklists\$unixHours
  $msgs| export-clixml $configPath\temporary_blacklists\$unixHours
}
```

```
if ($AddTempBlacklists2localBlacklist) { Add-TempBlacklists2localBlacklist }

#Generate monitoring output
if ( $eventcount -gt 0 ) {
  write-host "WARNING - $($msgs.Count) ($(($msgs |select -Unique).Count)
different) new Event(s)! "
  write-host ($($msgs |select -Unique) + "| Events=$(($msgs |select -
Unique).Count)")
  exit $exitcode
} else {
  write-host "OK - No new events!"
  write-host "| Events=$($msgs.Count)"
  exit 0
}
write-host "Something went terrible wrong! - Time for a Coffee!"
exit 3
```

Example Blacklist

```
<!-- VERSION 1.0 -->

<Objs Version="1.1.0.1"
xmlns="http://schemas.microsoft.com/powershell/2004/04">

<S>*Service Control Manager*Das Laden folgender Boot*cdrom (7026)</S>

<S>*GroupPolicy*Fehler*Der Computer befindet sich nicht an einem Standort.
Fehlercode 0x77F. (7320)*</S>

<S>*CodeIntegrity*Die Abbildintegrit*l3codeca.acm*nicht gefunden wurde.
(3002)</S>

<S>*Application/WMI - Ereignisfilter mit Abfrage*CIMV2*nicht durch diesen
Filter geschickt*(10)</S>
<S>*kernel-eventtracing/admin/Kernel-EventTracing - Beim Starten *Circular
Kernel Context Logger*0xC0000035. (2)</S>
<S>*WinRM - Der Client kann keine Verbindung mit dem in der Anforderung
angegebenen Ziel herstellen.*"winrm quickconfig"*(161)</S>
<S>WinRM - Fehler bei WSMan-Vorgang "Identify". Fehlercode: 2150858770
(142)*</S>
<S>TaskScheduler - Die Aufgabenplanung konnte die Aufgabe*für den
Benutzer*"DTWSxx\user"</S>

<S>WMI-Activity - ID:*;*Komponente: Unknown*Vorgang: Start
IWbemServices::CreateInstanceEnum*0x8004100c*Unknown*(5858)*</S>
<S>*(101)*</S>
<S>*(311)*</S>
<S>*(3002)*</S>
```

```
<S>Application*SharePoint*(2159)*</S>

<!-- Clients haben sich nicht beim Updateserver gemeldet -->

<S>Application*Windows*Update*Services*(13032)*</S>
<S>*Application/Windows Server Update Services - (13032)*</S>

<!-- Blackberry wenn nicht genutzt -->
<S>*Application/BlackBerry Messaging Agent BES Agent 1 - (20710)*</S>
<S>*Application/BlackBerry Controller - (20000)*</S>
<S>*Application/BlackBerry Messaging Agent BES Agent 1 - (40783)*</S>
<S>*Application/BlackBerry Messaging Agent BES Agent 1 - (40575)*</S>
<S>*Application/Attachment Service Client - (10000)*</S>

<!-- Kaspersky - bisher keine Lösung bekannt -->
<S>*Kaspersky Anti-Virus/Real-time file protection - (6041)*</S>
<S>*Application/KSCM8 - (1007)*</S>

<!-- No Risiko by ignoring this - at least if you trust the informations
from Microsoft -->
<S>*Kernel*EventTracing*(2)*</S>
<!-- WMI ERROR - social.technet.microsoft.com/Forums/windowsserver/en-
US/84d42b34-6941-4b60-9908-450ef8305813/event-5858-from-wmiactivity -->
<S>*WMI*Activity*(5858)*</S>
<S>*WinRM*(163)*</S>

<S>*Application/MSExchangeSA - (9327)*</S>

<!-- RDP Printers cant be connected -->
<S>*TerminalServices-PnPDevices - (36)*</S>

<!-- https://support.microsoft.com/de-de/kb/947238 -->
<S>*Application/User Profiles Service - (1530)*</S>

<!-- Internet Explorer -
http://blogs.technet.com/b/silvana/archive/2014/03/14/schannel-errors-on-sco
m-agent.aspx -->
<S>*System/Schannel*(36888)*</S>
<S>*System/Schannel*(36874)*</S>
<S>*System/Schannel*(36887)*</S>


<!-- Powershell Errors - There are mostly lots of it, so we ignore them,
because we need days to fix it -->
<S>*PowerShell - (4100)*</S>

<!-- Taskplaner kein Leerlauf deshalb keine ausführung -->
<S>*TaskScheduler - (135)*</S>

<S>*Known Folders API Service/KnownFolders - (1002)*</S>
```

```
<!-- Knows errors by design on Small Business Server 2011  -->
<S>*PowerShell*(32784)*</S>
<S>*WinRM*(129)*</S>
<S>*WinRM*(142)*</S>
<S>*Microsoft*Windows*CAPI2*(4107)*</S>
<S>*DCOM*(10016)*</S>
<S>*DCOM*(10009)*</S>
<S>*Application/Microsoft-SharePoint Products-SharePoint Foundation -
(5586)*</S>
<S>*Application/Microsoft-SharePoint Products-SharePoint Foundation -
(6772)*</S>
<S>*SharePoint*(6398)*</S>
<S>*MSExchange*CmdletLogs*(8)*</S>
<S>*MSExchange*CmdletLogs*(6)*</S>

<!-- If Sharepoint is used - Call me if you use and like sharepoint. Never
spoke to someone where both is true :) -->
<S>*Application/Microsoft-SharePoint Products-SharePoint Foundation -
(2159)*</S>

</Objs>
```