Published on *GeckoTechnology.com* (http://geckotechnology.com)

NAGIOS - Win32APIProxy

NAGIOS - Win32APIProxy

guitoo - Mon, 2010-03-15 16:24

This NAGIOS plugin allows to remotely monitor Windows hosts in a **agentless** fashion.

 [1]

Technically it is using the Win32 API from a Windows proxy server to the remote host. Syntax wise it mirrors NSClient++ [2] features.

Historically my friend decided to move away from HP SiteScope to Nagios. A large amount of 'monitors' (SiteScope lingo, 'service' in Nagios terminology) were related to Windows servers, monitored via the Win32 API, what Sitescope refers as NetBIOS protocol. Installing the NSClient++ agent, new to the team, was a concern for both security and stability of each server. Then naturally came the need to support the existing server infrastructure setup without any change.

Architecture

The Win32APIProxy is a small Perl script which runs on a Windows host. This proxy acts as a bridge between the UNIX world and the Windows world.

Nagios connects to the proxy via HTTP(s) protocol using a POST command, the request is received by Apache Web Server and the proxy, configured as a CGI script, issues a Win32 call to the target Windows host.

 [3]

The link between the proxy and the proxy must be setup and security on remote host properly

setup.

Download Win32APIProxy

Source code can be downloaded from SourceForge [4].

How to Install Win32APIProxy - Proxy Server

You need a Windows OS machine to install the proxy.

1. Install Apache Web Server. Binary can be downloaded from http://httpd.apache.org/download.cgi [5]
2. Install Perl. It has been tested with Active Perl, its binary can be downloaded from http://www.activestate.com/activeperl/downloads/ [6]
3. Install Proxy CGI script win32apiproxy.pl and its configuration file. Just place win32apiproxy.pl and win32apiproxy.config inside Apache2\cgi-bin\ directory
4. Create data directory, for example c:\win32apiproxy\hostdata\
5. Update win32apiproxy.config with directory from step 4, for example:

```
HOST_DATA_DIR=c:\win32apiproxy\hostdata\
PROXY_LOG=c:\win32apiproxy\proxy.log
```

7. Test with html page. Put temporarily win32test.html inside Apache2\htdocs, open it with your browser, type 'version' in the Command field and click Submit. It should result to a page showing the proxy version, such as 0.01
8. Optionaly setup proper security (HTTPs, BASIC authentication, OS hardening, Web server hardening, etc). This is out of scope of this page

How to Install Win32APIProxy - Nagios Server

1. Ensure Nagios::Plugin Perl module is installed. CPAN upgrade module can be used, such as:

```
cpaninstall "Nagios::Plugin"
```

2. Install check_win32apiproxy.pl script and its configuration file check_win32apiproxy.config inside NAGIOS_HOME/libexec (typically NAGIOS_HOME = /usr/local/nagios). Make sure config file location in check_win32apiproxy.pl is correct ($PROXYCONFIG variable)
3. Configure remote proxy address and security inside check_win32apiproxy.config. For example:

```
PROXY_URL=http://192.168.1.4:8080/cgi-bin/win32apiproxy.pl
#Security (used if USERNAME is not empty)
NETLOC=192.168.1.4:8080
REALM=myrealm
USERNAME=nagiossvr
PASSWORD=pass123
```

4. Add new command to Nagios inside NAGIOS_HOME/etc/objects/commands.cfg by adding the few lines bellow

```
# 'check_win32apiproxy' command definition
define command{
        command_name    check_win32apiproxy
        command_line    $USER1$/check_win32apiproxy.pl -H $HOSTADDRE
        }
```

5. Add Proxy monitoring by checking proxy version

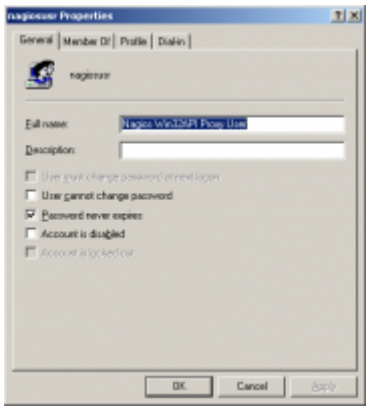
```
define host{
        use             windows-server
        host_name       Win32APIProxyServer
        alias           WIN32API Proxy Server
        address         192.168.1.4
        }
define service{
        use                     generic-service
        host_name               Win32APIProxyServer
        service_description     Proxy Version
        check_command           check_win32apiproxy!PROXYVERSION
        }
```

How to Monitor a Host

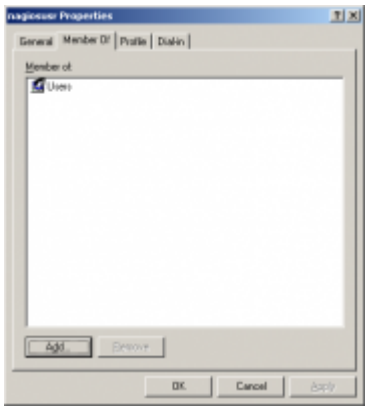
To check if step 1 and 2 are required, you can use either the HTML test page or the check script and submit one operation.

1. Configure host to accept remote inquiries from proxy. Follow steps from http://support.microsoft.com/kb/164018 [7] (which replaces KB Q158438). As a reference, I did this setting for a Windows 2003 Server host

a. Create remote user (in the example 'nagiosusr')

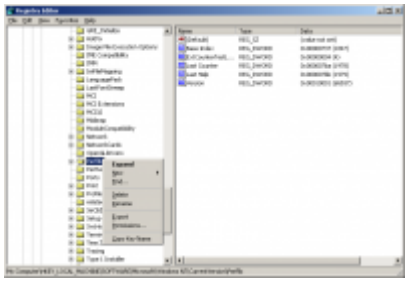


[8]

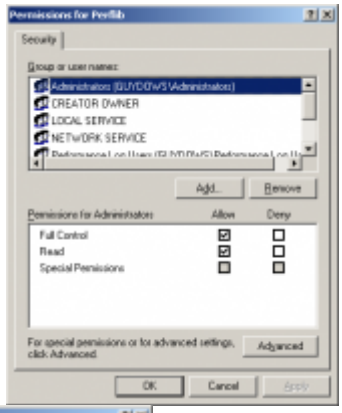


[9]

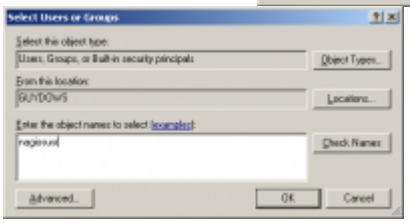
b. Grant read access of HKLM\Software\Microsoft\WindowsNT\CurrentVersion\PerfLib registry entry to 'nagiosusr'. If you forget this step, you will be able to create IPC$ connection, but registry browing will not be possible



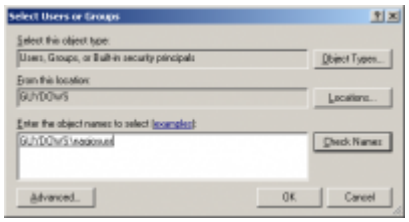
[10]



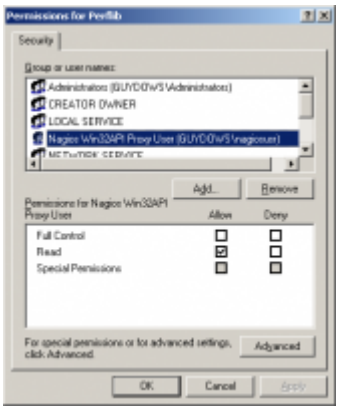
[11]



[12]

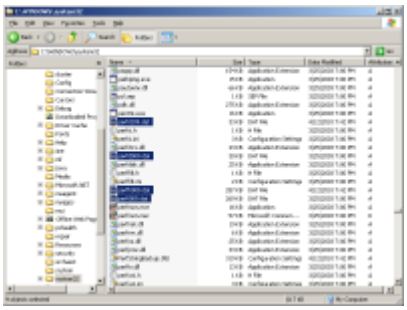


[13]



[14]

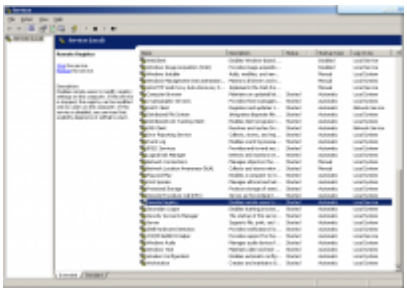
c. Ensure %windir%\System32\PERFCxxx.DAT and PERFHxxx.DAT can be read by 'nagiosusr'. xxx is language ID, 009 for english



[15]

[16]

d. Verify services are running (by default they are): Remote Procedure Call(RPC), server, Remote Registry



[17]

2. Optionally open firewall from proxy to remote host. Ports are:

   Port 135/TCP (I did not see traffic on this port though)
   Port 138/UDP
   Port 139/TCP
   Port 445/TCP

3. Optionally if the user used to run the proxy process is different that the one configured on the host, create a IPC connection

```
D:\>net use \\winserver /user:winserver\nagiosusr xxxpwdxxx /PERSIST
The command completed successfully.
D:\>net use
New connections will be remembered.
Status       Local     Remote                    Network
-------------------------------------------------------------------------
OK                     \\winserver\IPC$          Microsoft Windows N
The command completed successfully.
D:\>
```

4. Add this host to Nagios configuration file

```
define host{
        use             windows-server
```

```
        host_name       winserver
        alias           My Win32 Windows Server
        address         192.168.1.5
        }
```

5. Add services for this host

```
define service{
        use                     generic-service
        host_name               winserver
        service_description     Uptime
        check_command           check_win32apiproxy!UPTIME
        }
define service{
        use                     generic-service
        host_name               winserver
        service_description     CPU Load
        check_command           check_win32apiproxy!CPULOAD!-w 80 -c
        }
define service{
        use                     generic-service
        host_name               winserver
        service_description     Memory Usage
        check_command           check_win32apiproxy!MEMUSE!-w 80 -c
        }
define service{
        use                     generic-service
        host_name               winserver
        service_description     C:\ Drive Space
        check_command           check_win32apiproxy!USEDDISKSPACE!-l
        }
define service{
        use                     generic-service
        host_name               winserver
        service_description     ALG Service
        check_command           check_win32apiproxy!SERVICESTATE!-l
        }
define service{
        use                     generic-service
        host_name               winserver
        service_description     Explorer
        check_command           check_win32apiproxy!PROCSTATE!-l Exp
        }
```

Script Syntax

```
$ ./check_win32apiproxy.pl -h
Program: check_win32apiproxy.pl, version:0.01
Usage: check_win32apiproxy.pl -H host -v variable [-w warning] [-c critical]
-H, --hostname=HOST
   Name of the host to check
-w, --warning=INTEGER
   Threshold which will result in a warning status
-c, --critical=INTEGER
   Threshold which will result in a critical status
-l, --params=PARAMS
   Threshold which will result in a critical status
```

```
-t, --timeout=INTEGER
   Seconds before connection attempt times out (default: 10)
-h, --help
   Print this help screen
-V, --version
   Print version information
-v, --variable=STRING
   Variable to check
Valid variables are:
 PROXYVERSION = Get the remote win32apiproxy version
  Will return warning if check script and proxy version differ
 UPTIME =
  Get the uptime of the machine
  No specific parameters
  Warning and critical thresholds (in seconds) can be specified with -w and
 CPULOAD =
  Average CPU load since the last query
  Warning and critical thresholds (in CPU busy %) can be specified with -w ar
 USEDDISKSPACE =
  Size (GB) and percentage of disk use
  Request a -l parameter containing the drive letter only
  Warning and critical thresholds (in disk used %) can be specified with -w a
 MEMUSE =
  Virtual and Physical Memory use (MB).
  Warning and critical thresholds (in virtual memory used %) can be specifie
 SERVICESTATE =
  Check the state of one or several services. Return critical if at least 1 s
  Request a -l parameters with the following syntax: -l <service1>,<service2>
 PROCSTATE =
  Check if one or several process are running
  Same syntax as SERVICESTATE
```

Sample Script Outputs

```
$ ./check_win32apiproxy.pl -v PROXYVERSION
Proxy Version: 0.01
$ ./check_win32apiproxy.pl -v UPTIME
-H winserver -w 604800
System Uptime: 0 day(s) 2 hour(s) 21 minute(s)|uptime=8518;604800;
$ ./check_win32apiproxy.pl -v CPULOAD
-H winserver -w 40 -c 80
CPU Load: 52.3%|busy%=52.3;40;80
$ ./check_win32apiproxy.pl -v MEMUSE
-H winserver -c 30
Virtual Memory Usage: total: 2441.17 MB - used: 578.14 MB (23.7%) - free: 180
pages/sec: 7.5|virtused=578.14MB,virtused%=23.7;;30,physused=366.00MB,pages/s
$ ./check_win32apiproxy.pl -v USEDDISKSPACE
-H winserver -l c -w 80 -c 90
C:\ - total: 16.28 GB - used: 8.88 GB (54.5%) - free: 7.40 GB (45.5%) |'C:\ U
$ ./check_win32apiproxy.pl -v SERVICESTATE
-H winserver -l alg,alerter
Service State: alg:SERVICE_RUNNING alerter:SERVICE_STOPPED
$ ./check_win32apiproxy.pl -v PROCSTATE
-H winserver -l svchost,explorer,zzz
Process State: svchost:Running explorer:Running zzz:Not running
```

Report Issues or Request Enhancements

Just click on the "Contact" link inside the top left box. In case of issue, I am glad to track down what went wrong and get the Win32API Proxy fixed ASAP.

Up-Coming Enhancements

- List all services and running processes
- Display counters
- Automatic creation of IPC link from proxy to host
- Manage timeout from proxy to host
- Test HTTPs link from Nagios to proxy

Version History

Version	Date	Notes
0.01	17-mar-2010	Initial release
0.02	TBD	Fix: IP hostname regex is incorrect from from IPs Display counters

Drupal

Source URL: http://geckotechnology.com/Win32APIProxy

Links:
[1] http://geckotechnology.com/sites/default/files/win32apiproxy_screen1.PNG
[2] http://nsclient.org
[3] http://geckotechnology.com/sites/default/files/win32apiproxy_archdiag.png
[4] http://sourceforge.net/projects/win32apiproxy/files/
[5] http://httpd.apache.org/download.cgi
[6] http://www.activestate.com/activeperl/downloads/
[7] http://support.microsoft.com/kb/164018
[8] http://geckotechnology.com/sites/default/files/win32apiproxy_rhost_1.png
[9] http://geckotechnology.com/sites/default/files/win32apiproxy_rhost_2.png
[10] http://geckotechnology.com/sites/default/files/win32apiproxy_rhost_3.png
[11] http://geckotechnology.com/sites/default/files/win32apiproxy_rhost_4.png
[12] http://geckotechnology.com/sites/default/files/win32apiproxy_rhost_5.png
[13] http://geckotechnology.com/sites/default/files/win32apiproxy_rhost_6.png
[14] http://geckotechnology.com/sites/default/files/win32apiproxy_rhost_7.png
[15] http://geckotechnology.com/sites/default/files/win32apiproxy_rhost_8.png
[16] http://geckotechnology.com/sites/default/files/win32apiproxy_rhost_9.png
[17] http://geckotechnology.com/sites/default/files/win32apiproxy_rhost_A.png