Interested in learning
more about security?

SANS Institute
InfoSec Reading Room

This paper is from the SANS Institute Reading Room site. Reposting is not permitted without express written permission.

Winquisitor: Windows Information Gathering Tool

Gathering and reviewing information from multiple systems in a timely manner is a critical function for Windows administrators. This information allows administrators to respond to threats in order to minimize risks to their environments. Winquisitor is a tool that facilitates the timely retrieval of information from multiple Windows systems enabling the administrator to respond in an appropriate amount of time. Unlike other command line tools, Winquisitor allows multiple types of queries in a single command with s...

Copyright SANS Institute
Author Retains Full Rights

AD

Winquisitor: Windows Information Gathering Tool

GIAC (GCIH) Gold Certification

Author: Mike Cardosa, mcardosa@gmail.com
Advisor: Rick Wanner

Accepted: TBD

Abstract

Gathering and reviewing information from multiple systems in a timely manner is a critical function for Windows administrators. This information allows administrators to respond to threats in order to minimize risks to their environments. Winquisitor is a tool that facilitates the timely retrieval of information from multiple Windows systems enabling the administrator to respond in an appropriate amount of time. Unlike other command line tools, Winquisitor allows multiple types of queries in a single command with several output formats. This saves the administrator the time it would take to combine the results from multiple command line tools into a usable and actionable format.

© 2010 SANS Institute, Author retains full rights.

1. Introduction

Administrators who manage Microsoft Windows computers on their networks need to gather information from these systems in order to ensure that they are operating smoothly and securely. Some of the required information includes:

- Patch levels
- Possible virus infections
- Registry settings and values
- Local users and groups
- The state of processes and services

While most administrators are capable of querying this information using Windows GUI tools, it is an approach that does not scale easily to more than a few individual systems. Administrators can leverage command-line tools to extract the information, but in many instances, these tools are not well known or documented. Furthermore, it is often difficult to combine the output from multiple tools or queries without manually editing files or scripting an entire solution. As a result, IT organizations often maintain a collection of disparate scripts and Excel files that must be manually updated.

Many IT organizations do not have the necessary time or resources to develop a custom scripting solution. They are therefore required to buy a commercial tool or collect data on a less timely basis, which increases risk. Winquisitor is an attempt to streamline the data collection process so that administrators can react and deal with potential issues in a more appropriate timeframe.

2. Existing Tools

Microsoft provides a number of tools that enable administrators to query information from both the local and remote Windows computers. These include WMI, sc, the net commands, and the reg commands. While these tools are extremely useful, there are

© 2010 The SANS Institute As part of the Information Security Reading Room Author retains full rights.

many for an administrator to be aware of and it is not easy to combine the output from multiple tools. It is also not easy to construct a single command that will execute multiple types of tests against multiple target systems. Winquisitor is an attempt to solve this problem.

3. Design Considerations

The solution was designed to meet the following requirements:

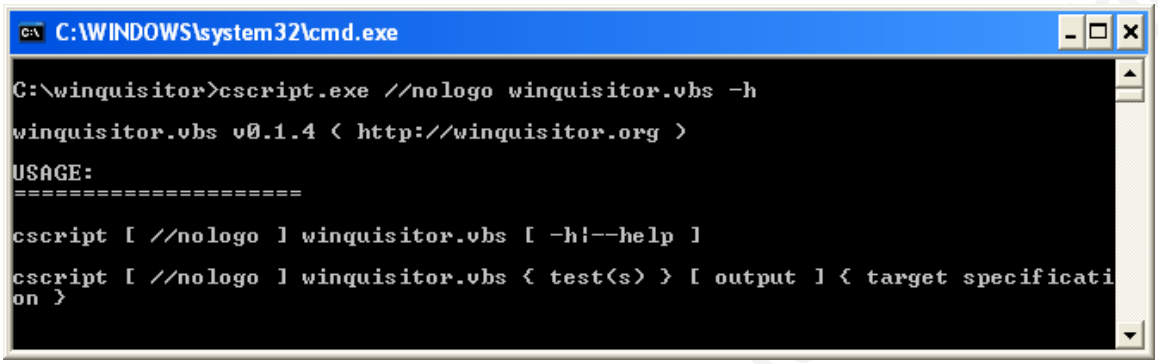
- Compatible with the majority of current and planned Windows operating system versions including:
  - Windows 2000
  - Windows XP
  - Windows 2003
  - Windows Vista
  - Windows 7
  - Windows 2008
- No additional software to install other than the tool itself
- No need to compile source code
- Able to run queries against one or more Windows computers
- Able to simultaneously run multiple types of queries
- Support multiple output formats (CSV and XML)
- Allow the user to run a custom query that the tool's author had not considered

Given the above requirements, a script written using VBScript seemed to be the most appropriate format for the solution. Although Microsoft is currently encouraging the adoption of PowerShell, only recent versions of Windows have PowerShell installed by default. VBScript, on the other hand, is compatible with the widest range of Windows versions and Microsoft is not planning to discontinue support for it anytime soon. (Ed Wilson and Craig Liebendorfer, 2009)

© 2010 The SANS Institute As part of the Information Security Reading Room Author retains full rights.

4. Functionality Overview

Winquisitor consists of a single VBScript file, winquisitor.vbs. Additionally, there is an optional XSL file, winquisitor.xsl, which can be used to view an XML-formatted output file in a web browser. Winquisitor can be run on any modern Windows system using cscript.exe as the script host.



The following sections will highlight Winquisitor's functions and primary arguments. All of Winquisitor's options and arguments can be found in the README.txt file included in the Appendix or downloaded from http://www.winquisitor.org.

4.1. Specifying Target Machines

Winquisitor's strength lies in its ability to run multiple tests against multiple target systems. There are several command line arguments that are used to specify target systems.

Target (-t, --target)
Specifies an individual target. This could be in the form of a hostname or an IP address. A user can supply multiple target specifications to the script.

Target file (-T, --target-file)
Rather than specify target hosts individually on the command line, the user can instead supply a text file that contains the names of all the target hosts. The file should contain one host per line.

© 2010 The SANS Institute As part of the Information Security Reading Room Author retains full rights.

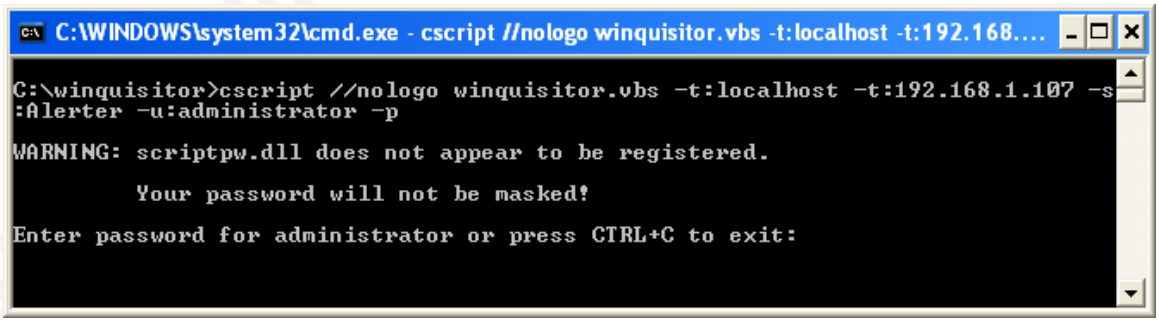
Username (-u, --username)

By default, Winquisitor runs with the credentials of the user running the script. The user can provide an alternate username to Winquisitor that will be used to authenticate to each of the target systems. This can be either a local or a domain account.

Password (-p, --password)

If the user specifies an alternate username, a password must also be supplied to the script. There are two methods for providing the password. If the user wishes, the password can be given on the command line in the format: **-p:"password123"**.

If the user prefers not to specify the password on the command line as an argument, Winquisitor will prompt for the password interactively at runtime. Winquisitor uses the ScriptPW.Password object (Microsoft Corporation, 2004) to mask the password as it is typed. However, ScriptPW.Password is only available on Windows 2003 and Windows XP systems. (Microsoft Corporation, 2008) On non-XP/2003 systems, there is no way to mask the password. A warning message is presented to the user if the password cannot be masked.



Warning message alerting the user that there is no way to mask their password

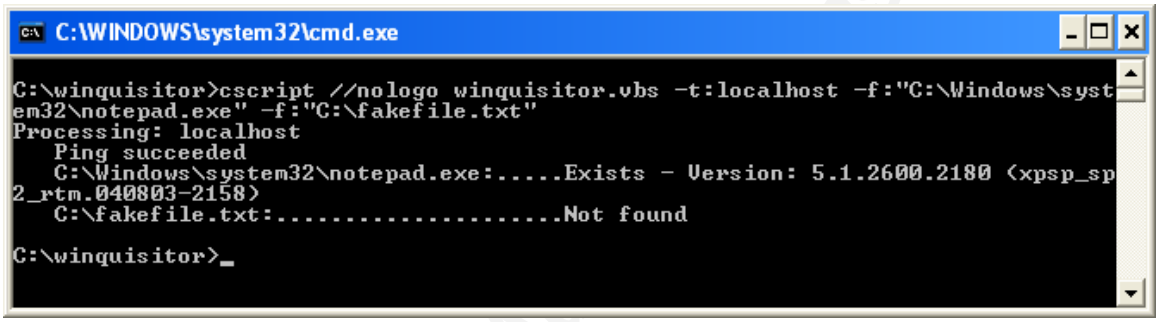
© 2010 The SANS Institute As part of the Information Security Reading Room Author retains full rights.

4.2. Specifying Tests and Queries

Winquisitor provides an easy way to run multiple types of tests against target systems. A user can run any number or combination of the following tests against an arbitrary number of target systems.

File test (-f, --file)

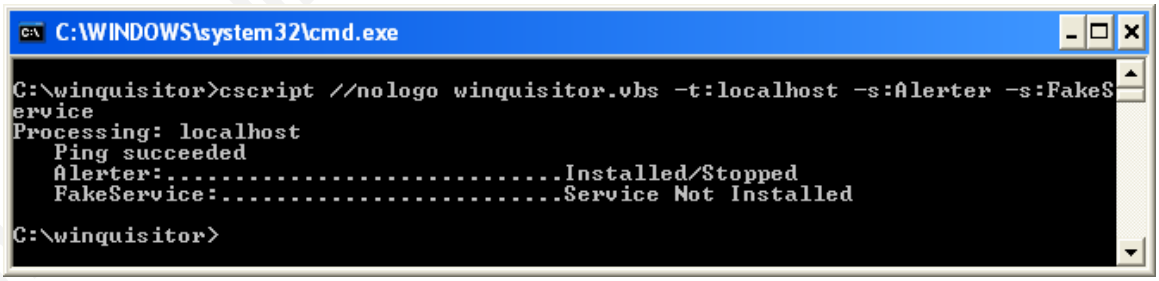
Test for the existence of a file on a target system. If the file is found, the file version information is also retrieved. Winquisitor queries the CIM_Datafile class (Microsoft Corporation) for this information.



Example of file tests

Service test (-s, --service)

Test for the state of a given service on a target system. Winquisitor queries the Win32_Service class (Microsoft Corporation) for this information.



Example of service tests

Patch/Update test (-pa, --patch)

© 2010 The SANS Institute As part of the Information Security Reading Room Author retains full rights.

Test whether or not a given patch/update has been installed on a target system. Winquisitor queries the Win32_QuickFixEngineering class (Microsoft Corporation) for this information.[1]

```
C:\WINDOWS\system32\cmd.exe
C:\winquisitor>cscript //nologo winquisitor.vbs -t:localhost -pa:KB975467 -pa:KB
000000
Processing: localhost
   Ping succeeded
   KB975467:..............................Installed
   KB000000:..............................Not Found

C:\winquisitor>
```

Example of patch tests

Process test (-pr, --process)

Test whether or not a given process is currently running on a target system. Winquisitor queries the Win32_Process class (Microsoft Corporation) for this information.

```
C:\WINDOWS\system32\cmd.exe
C:\winquisitor>cscript //nologo winquisitor.vbs -t:localhost -pr:notepad.exe -pr
:svchost.exe -pr:fakeprocess.exe
Processing: localhost
   Ping succeeded
   notepad.exe:...........................1 Instance Running
   svchost.exe:...........................7 Instances Running
   fakeprocess.exe:.......................Not found

C:\winquisitor>_
```

Example of process tests

Registry key test (-rk, --registry-key)

Test whether or not a given registry key exists on a target system. Winquisitor queries the StdRegProv class (Microsoft Corporation) for this information.

[1] Starting with Windows Vista, the Win32_QuickFixEngineering Class "returns only the updates supplied by Component Based Servicing (CBS)." (Microsoft Corporation) The next version of Winquisitor will supplement the information with that retrieved from the Windows Update Agent API (Microsoft Corporation, 2009) to return accurate results.

© 2010 The SANS Institute As part of the Information Security Reading Room Author retains full rights.

```
C:\WINDOWS\system32\cmd.exe
C:\winquisitor>cscript //nologo winquisitor.vbs -t:localhost -rk:"HKEY_LOCAL_MAC
HINE\SOFTWARE\Microsoft\Windows\CurrentVersion" -rk:"HKLM\SOFTWARE\Microsoft\Win
dows\CurrentVersion\Run" -rk:HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\PrevV
ersion"
Processing: localhost
   Ping succeeded
   HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion:
     ..................................Exists
   HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run:
     ..................................Exists
   HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\PrevVersion:
     ..................................Not found

C:\winquisitor>_
```

Example of registry key tests

Registry value test (-rv, --registry-value)

Retrieve the value of a given registry key if it exists on a target system. Winquisitor queries the StdRegProv class (Microsoft Corporation) for this information.

```
C:\winquisitor>cscript //nologo winquisitor.vbs -t:localhost -rv:"HKEY_LOCAL_MAC
HINE\SOFTWARE\Microsoft\Windows\CurrentVersion\DateTime\Servers\1" -rv:"HKLM\SOF
TWARE\Microsoft\Windows\CurrentVersion\DateTime\Servers\2" -rv:"HKEY_LOCAL_MACHI
NE\SOFTWARE\Microsoft\Windows\CurrentVersion\DateTime\Servers\3"
Processing: localhost
   Ping succeeded
   HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\DateTime\Servers
\1:
     ..................................(REG_SZ) = time.windows.com
   HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\DateTime\Servers\2:
     ..................................(REG_SZ) = time.nist.gov
   HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\DateTime\Servers
\3:
     ..................................Registry Key Not found

C:\winquisitor>_
```

Example of registry value tests

Local user test (-lu, --local-user)

Test for the existence of a local user on a target system. Winquisitor queries the Win32_UserAccount class (Microsoft Corporation) for this information.

```
C:\winquisitor>cscript //nologo winquisitor.vbs -t:localhost -lu:administrator -
lu:johnny
Processing: localhost
   Ping succeeded
   administrator:.......................Exists
   johnny:..............................Not Found

C:\winquisitor>
```

Example of local user tests

Mike Cardosa

© 2010 The SANS Institute As part of the Information Security Reading Room Author retains full rights.

Local group test (-lg, --local-group)

Test for the existence of a local group. If the local group exists, enumerate all members. Winquisitor uses the Group object of the WinNT provider (Microsoft Corporation, 2009) to retrieve this information.

```
C:\WINDOWS\system32\cmd.exe

C:\winquisitor>cscript //nologo winquisitor.vbs -t:localhost -lg:guests -lg:"Evi
l Users"
Processing: localhost
   Ping succeeded
   guests:
      ..................................Guest
   Evil Users:...........................Group not found

C:\winquisitor>_
```

Example of local group tests

Custom query (-cq, --custom-query)

One of the most powerful features of Winquisitor is the ability to specify one or more custom queries to retrieve information from the **root\cimv2** namespace. This allows the user to gather information that might not already be included in Winquisitor's functionality. For example, this could be used to retrieve hardware or custom settings information.

By default, Winquisitor will provide the number of records that the custom query returned.

```
C:\WINDOWS\system32\cmd.exe

C:\winquisitor>cscript //nologo winquisitor.vbs -t:localhost -cq:"select name fr
om win32_logicaldisk"
Processing: localhost
   Ping succeeded
   select name from win32_logicaldisk:..3 Records

C:\winquisitor>
```

Example of a custom query with summarized results

If the user would like to receive detailed query results, he/she may provide the **--result-detail** argument to Winquisitor.

© 2010 The SANS Institute As part of the Information Security Reading Room Author retains full rights.

```
C:\WINDOWS\system32\cmd.exe
C:\winquisitor>cscript //nologo winquisitor.vbs -t:localhost -cq:"select name fr
om win32_logicaldisk" --result-detail
Processing: localhost
   Ping succeeded
   select name from win32_logicaldisk:
      ..................................DeviceID: C: || Name: C:
      ..................................DeviceID: D: || Name: D:
      ..................................DeviceID: M: || Name: M:

C:\winquisitor>_
```

Example of a custom query with detailed results

4.3. Output Formats

In addition to the default output on standard out, Winquisitor provides two other output formats: CSV and XML. These formats not only supply an easy way for a user to visualize the results in a spreadsheet or browser but also give the user the option of parsing the results programmatically if needed.

For example, the user can have the below Winquisitor command result in any of the following formats.

Winquisitor command

```
C:\WINDOWS\system32\cmd.exe
C:\winquisitor>cscript //nologo winquisitor.vbs -t:localhost -t:192.168.1.107 -s
:Alerter
```

Default output

```
C:\WINDOWS\system32\cmd.exe
C:\winquisitor>cscript //nologo winquisitor.vbs -t:localhost -t:192.168.1.107 -s
:Alerter
Processing: localhost
   Ping succeeded
   Alerter:..............................Installed/Stopped
Processing: 192.168.1.107
   Ping succeeded
   Alerter:..............................Installed/Stopped

C:\winquisitor>_
```

CSV output (viewed as a spreadsheet)

© 2010 The SANS Institute As part of the Information Security Reading Room Author retains full rights.

Computer	Connection	TestType	Parameter	Result
localhost	Success	Service	Alerter	Installed/Stopped
192.168.1.107	Success	Service	Alerter	Installed/Stopped

XML output (unformatted)


```
– <winquisitor_audit>
  – <scan>
    – <scan_info>
        winquisitor.vbs -t:localhost -t:192.168.1.107 -s:Alerter -oX:results.xml
      </scan_info>
      <start_date>1/9/2010</start_date>
      <start_time>2:11:40 PM</start_time>
    – <target>
        <computer>localhost</computer>
        <connection>Success</connection>
      – <test>
          <type>Service</type>
          <value>Alerter</value>
          <result>Installed/Stopped</result>
        </test>
      </target>
    – <target>
        <computer>192.168.1.107</computer>
        <connection>Success</connection>
      – <test>
          <type>Service</type>
          <value>Alerter</value>
          <result>Installed/Stopped</result>
        </test>
      </target>
      <end_date>1/9/2010</end_date>
      <end_time>2:11:41 PM</end_time>
    </scan>
  </winquisitor_audit>
```


XML output (formatted with the default winquisitor.xsl file)[2]

[2] The user may specify an alternate XSL file in order to customize the display format.

© 2010 The SANS Institute As part of the Information Security Reading Room Author retains full rights.

Winquisitor Results

Scan	winquisitor.vbs -t:localhost -t:192.168.1.107 -s:Alerter -oX:results.xml
Started	2:15:22 PM on 1/9/2010
Ended	2:15:22 PM on 1/9/2010

Computer	Test type	Parameter	Result
192.168.1.107			
	Service	Alerter	Installed/Stopped
localhost			
	Service	Alerter	Installed/Stopped

5. Putting It All Together: Trojan.Bredolab Example

Testing for the existence of a virus or trojan on a target system is a common Winquisitor use case. For example, an IDS might generate an alert that a certain system is exhibiting symptoms of the Bredolab Trojan. Symantec's description of this malware includes specific technical details of an infection. (Symantec Corporation, 2009) An administrator can leverage this information to confirm the existence of the Trojan on the identified system as well as proactively scan the other systems that he/she manages for signs of infection.

The following command would accomplish this task and present the results in an XML format suitable for printing.

```
C:\WINDOWS\system32\cmd.exe
C:\winquisitor>cscript //nologo winquisitor.vbs -t:infectedhost.example.com -T:a
ccounting-workstations.txt -f:"C:\windows\system32\wbem\grpconv.exe" -f:"C:\wind
ows\system32\grpconv.exe" -f:"C:\windows\system32\dllcache\grpconv.exe" -oX:resu
lts.xml
```

6. Future Enhancements

There are several planned updates to Winquisitor:

Mike Cardosa

© 2010 The SANS Institute As part of the Information Security Reading Room Author retains full rights.

1. **Configuration file –** Command line arguments are extremely useful for scripts, but they can become tedious when the script must be run multiple times with similar arguments. The option to create a configuration file containing all of the arguments to the script would be more convenient in these situations.
2. **Searching with regular expressions –** There are some instances where the file name might not be known. For example, a virus or worm might create the file "C:\Windows\system32\[7 random numbers].exe". The ability to search for the file using a regular expression would locate this file if it existed, while the current iteration of Winquisitor could not.
3. **Support for multiple threads –** Although Winquisitor can save administrators a large amount of time over gathering information manually, the script does take several seconds to scan a single computer. Since it will only scan one system at a time, those seconds can add up when scanning an entire network. While it is possible for an administrator to manually start Winquisitor multiple times with different targets, the capability to scan multiple systems in parallel would simplify this process.

7. Conclusion

Although Winquisitor originated as a GCIH Gold Certification Project, it will continue as a supported tool for Windows administrators. The hope is that these administrators can save time by leveraging an existing tool to retrieve information from their systems rather than developing custom scripts. However, if script development does make more sense in certain situations, script developers could leverage the classes and methods used by Winquisitor as starting points for their scripts.

Winquisitor is available for download at http://www.winquisitor.org.

Mike Cardosa

© 2010 SANS Institute, Author retains full rights.

8. Appendix

8.1. Winquisitor README.txt

winquisitor.vbs v0.1.4 (http://winquisitor.org)

DESCRIPTION:
==========

Winquisitor aims to simplify the tasks that Windows administrators must perform by providing a simple way to gather information from a number of Windows systems, reducing custom script development.

DISCLAIMER:
==========

The author makes no representations about the suitability of this software for any purpose. This software is provided AS IS and without any express or implied warranties, including, without limitation, the implied warranties of merchantability and fitness for a particular purpose. The entire risk arising out of the use or performance of this script and documentation remains with you. In no event shall the author, or anyone else involved in the creation, production, or delivery of the scripts be liable for any damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information, or other pecuniary loss) arising out of the use of or inability to use the script or documentation, even if the author has been advised of the possibility of such damages.

INSTALLATION:
==========

Simply extract winquisitor.vbs to any local directory.

If you wish to view XML in a browser formatted using the included winquisitor.xsl, copy winquisitor.xsl to the report directory or specify the path to the XSL file on the command line with the -xsl option.

Mike Cardosa

© 2010 SANS Institute, Author retains full rights.

```
USAGE:
======================

cscript [ //nologo ] winquisitor.vbs [ -h|--help ]

cscript [ //nologo ] winquisitor.vbs { test(s) } [ output ] { target specification }


PARAMETERS:
======================

 OUTPUT:
 --------------------
  -h,--help                 Display this usage screen
  -v                       Enable verbose output
  -vv                      Enable very verbose output
  -d,--debug                Enable debugging output
  -q,--quiet                Suppress output
  -oC:file                  Output CSV results to the given file
  -oX:file                  Output XML results to the given file
  -xsl:file                 Reference the given XSL document in the
                           XML output file instead of the default
                           winquisitor.xsl
  --web-xsl                 Reference the XSL file hosted on winquisitor.org
                           in the XML output file instead of the
                           default winquisitor.xsl
  --append-output           Append to the given output file instead of
                           overwriting

 TARGET SPECIFICATION:
 --------------------
  -t,--target:computer      Add the given computer to the list of computers
                           to test
  -T,--target-file:file     Read targets from the given file
                           (one target per line)
  -np,--no-ping             Do not ping targets before trying to connect
  -u,--username:username    Connect to targets with the given username
  -p,--password:password    Connect to targets with the given password
                           If a username was given and a password was
                           not specified, then the user will be prompted
                           for a password.

 TESTS:
 --------------------
  -f,--file:file            Test the existence and version of the given file
```

Mike Cardosa

© 2010 The SANS Institute As part of the Information Security Reading Room Author retains full rights.

```
 -s,--service:service         Test the state of the given service
 -pa,--patch:patch            Test whether a given patch has been applied
 -pr,--process:process        Test whether or not a process is running
 -rk,--registry-key:key       Test the existence and/or value of the
                              given registry key
 -rv,--regisry-value:value    Test the given registry value
 -lu,--local-user:username    Test the existence of the given user
 -lg,--local-group:groupname  Enumerate the members of the given local group
 -cq,--custom-query:query     WMI query against the CIMV2 namespace
 --result-detail              When used with -cq, detailed query results
                              are provided instead of a summary


EXAMPLES:
=====================

 EXAMPLE 1:
 --------------------
  Test for the Alerter service on machines 192.168.1.10 and 192.168.1.11
    and record results in XML format to results.xml

   winquisitor.vbs -t:192.168.1.10 -t:192.168.1.11 -s:Alerter -oX:results.xml


 EXAMPLE 2:
 --------------------
  Test for the existence of the file "C:\Windows\system32\evil.exe" and
    the running process trojan.exe against 192.168.1.10, 192.168.1.1, and all
    hosts listed in targets.txt. Record results in XML format to results.xml

   winquisitor.vbs -t:192.168.1.10 -t:192.168.1.11 -T:targets.txt
     -f:"C:\Windows\system32\evil.exe" -p:"trojan.exe" -oX:results.xml


 EXAMPLE 3:
 --------------------
  Check for patch KB890046 and run a custom query against 192.168.1.11
    displaying detailed results. Do not ping the target first. Append the
    results in CSV format to results.csv

   winquisitor.vbs -t:192.168.1.11 -np -pa:KB890046 -oC:results.csv
     -cq:"select caption from win32_useraccount" --result-detail --append-output
```

Mike Cardosa

© 2010 The SANS Institute As part of the Information Security Reading Room Author retains full rights.

9. Works Cited

Ed Wilson and Craig Liebendorfer, S. G. (2009, March 4). *Hey, Scripting Guy! How Do I Migrate My VBScript WMI Queries to Windows PowerShell?* Retrieved January 5, 2010, from Hey, Scripting Guy! Blog: http://blogs.technet.com/heyscriptingguy/archive/2009/03/04/how-do-i-migrate-my-vbscript-wmi-queries-to-windows-powershell.aspx

Microsoft Corporation. (2008). *2008 Winter Scripting Games Tip of the Week: Working With User Input*. Retrieved January 3, 2010, from Microsoft TechNet: http://www.microsoft.com/technet/scriptcenter/funzone/games/tips08/gtip0208.mspx

Microsoft Corporation. (2009, November 12). *ADSI Objects of WinNT (Windows)*. Retrieved January 8, 2010, from MSDN Windows Developer Center: http://msdn.microsoft.com/en-us/library/aa772211%28VS.85%29.aspx

Microsoft Corporation. (n.d.). *CIM_Datafile Class (Windows)*. Retrieved January 3, 2010, from MSDN Windows Developer Center: http://msdn.microsoft.com/en-us/library/aa387236%28VS.85%29.aspx

Microsoft Corporation. (2004, February 12). *Masking Passwords in XP and Windows 2003*. Retrieved January 3, 2010, from The Scripting Guys' First Blog: http://blogs.msdn.com/gstemp/archive/2004/02/12/71903.aspx

Microsoft Corporation. (n.d.). *StdRegProv Class (Windows)*. Retrieved January 8, 2010, from MSDN Windows Developer Center: http://msdn.microsoft.com/en-us/library/aa393664%28VS.85%29.aspx

© 2010 SANS Institute, Author retains full rights.

© 2010 The SANS Institute As part of the Information Security Reading Room Author retains full rights.

Microsoft Corporation. (n.d.). *Win32_Process Class (Windows)*. Retrieved January 8, 2010, from MSDN Windows Developer Center: http://msdn.microsoft.com/en-us/library/aa394372%28VS.85%29.aspx

Microsoft Corporation. (n.d.). *Win32_QuickFixEngineering Class (Windows)*. Retrieved January 9, 2010, from MSDN Windows Developer Center: http://msdn.microsoft.com/en-us/library/aa394391%28VS.85%29.aspx

Microsoft Corporation. (n.d.). *Win32_Service Class (Windows)*. Retrieved January 9, 2010, from MSDN Windows Developer Center: http://msdn.microsoft.com/en-us/library/aa394418%28VS.85%29.aspx

Microsoft Corporation. (n.d.). *Win32_UserAccount Class (Windows)*. Retrieved January 8, 2010, from MSDN Windows Developer Center: http://msdn.microsoft.com/en-us/library/aa394507%28VS.85%29.aspx

Microsoft Corporation. (2009, November 18). *Windows Update Agent API (Windows)*. Retrieved January 9, 2010, from MSDN Windows Developer Center: http://msdn.microsoft.com/en-us/library/aa387099%28VS.85%29.aspx

Symantec Corporation. (2009, May 29). *Trojan.Bredolab Technical Details*. Retrieved January 8, 2010, from Symantec: http://www.symantec.com/security_response/writeup.jsp?docid=2009-052907-2436-99&tabid=2

© 2010 SANS Institute, Author retains full rights.

© 2010 The SANS Institute As part of the Information Security Reading Room Author retains full rights.

Last Updated: September 3rd, 2015

Upcoming SANS Training

Click Here for a full list of all Upcoming SANS Events by Location

SANS Crystal City 2015	Crystal City, VAUS	Sep 08, 2015 - Sep 13, 2015	Live Event
SANS Network Security 2015	Las Vegas, NVUS	Sep 12, 2015 - Sep 21, 2015	Live Event
SANS Seoul 2015	Seoul, KR	Sep 14, 2015 - Sep 19, 2015	Live Event
Data Breach Investigation Summit & Training	Dallas, TXUS	Sep 21, 2015 - Sep 26, 2015	Live Event
SANS Baltimore 2015	Baltimore, MDUS	Sep 21, 2015 - Sep 26, 2015	Live Event
SANS Perth 2015	Perth, AU	Sep 21, 2015 - Sep 26, 2015	Live Event
SANS Tallinn 2015	Tallinn, EE	Sep 21, 2015 - Sep 26, 2015	Live Event
SANS ICS Amsterdam 2015	Amsterdam, NL	Sep 22, 2015 - Sep 28, 2015	Live Event
SANS Bangalore 2015	Bangalore, IN	Sep 28, 2015 - Oct 17, 2015	Live Event
SANS DFIR Prague 2015	Prague, CZ	Oct 05, 2015 - Oct 17, 2015	Live Event
SANS Seattle 2015	Seattle, WAUS	Oct 05, 2015 - Oct 10, 2015	Live Event
SANS Tysons Corner 2015	Tysons Corner, VAUS	Oct 12, 2015 - Oct 17, 2015	Live Event
SOS: SANS October Singapore 2015	Singapore, SG	Oct 12, 2015 - Oct 24, 2015	Live Event
SANS Gulf Region 2015	Dubai, AE	Oct 17, 2015 - Oct 29, 2015	Live Event
SANS Cyber Defense San Diego 2015	San Diego, CAUS	Oct 19, 2015 - Oct 24, 2015	Live Event
SANS Tokyo Autumn 2015	Tokyo, JP	Oct 19, 2015 - Oct 31, 2015	Live Event
SANS South Florida 2015	Fort Lauderdale, FLUS	Nov 09, 2015 - Nov 14, 2015	Live Event
SANS Sydney 2015	Sydney, AU	Nov 09, 2015 - Nov 21, 2015	Live Event
SANS London 2015	London, GB	Nov 14, 2015 - Nov 23, 2015	Live Event
Pen Test Hackfest Summit & Training	Alexandria, VAUS	Nov 16, 2015 - Nov 23, 2015	Live Event
SANS Hyderabad 2015	Hyderabad, IN	Nov 24, 2015 - Dec 04, 2015	Live Event
SANS Cape Town 2015	Cape Town, ZA	Nov 30, 2015 - Dec 05, 2015	Live Event
SANS San Francisco 2015	San Francisco, CAUS	Nov 30, 2015 - Dec 05, 2015	Live Event
SANS Milan 2015	OnlineIT	Sep 07, 2015 - Sep 12, 2015	Live Event
SANS OnDemand	Books & MP3s OnlyUS	Anytime	Self Paced