Target Client Configuration

- How Does Open-AudIT Work?
- Windows
  - User Credentials Requirements
  - Enabling the Administrator account on non-Domained machines
  - Windows 7 and 2008 R2 not submitting audit using HTTPS
  - DCOM
  - UAC
  - Local Security Policies
  - Simple File Sharing (XP)
  - WMI
  - Networking with WMI
  - Testing Remote WMI
  - Testing Remote WMI #2
  - Matching Discovery Logs to WMI issues
  - Winexe requirements (Linux only) on Windows machines
  - AntiVirus
  - Windows Firewall
- Linux
  - SSHGuard
- ESX
- OSX
- AIX
- Solaris
- SNMP

The audit function of Open-AudIT is designed to work "out of the box" as much as possible with the default settings of target devices. Below are the requirements for the audit to work and some hints for items to configure when things are not working as planned.

How Does Open-AudIT Work?

Open-AudIT runs an Nmap discovery on each target IP address. Open-AudIT scans the Nmap top 1000 TCP ports, as well as UDP 62078 (Apple IOS) and UDP 161 (SNMP). For Open-AudIT to consider a target IP to have a device responding, any of the Nmap Top 1000 TCP Ports must be responding or the UDP 62078. A target that responds to UDP 161 (SNMP) only and NO other ports (TCP or UDP 62078 / 161) is not considered to be responding.

Why do we not consider a UDP port 161 response enough? Because it is very common for firewalls separating network segments to respond with UDP 161 for a target IP regardless of there being an actual device present at that IP address.

Why don't we simply scan every port TCP and UDP? On local networks this is usually OK to do, but on remote subnets this can take (literally) hours ***PER IP ADDRESS***.

> **UDP port 161 Workaround**
>
> OA needs to see more than just UDP port 161 open on a device to consider it a legitimate device. If OA is only seeing UDP port 161 open OA will consider it a false positive and move on. If this is your situation you can edit the /usr/local/open-audit/other/discover_subnet.sh file and set consider_161_enough to “y"

Windows

On Windows, Open-AudIT uses WMI via VBscript as it's primary method of auditing. SNMP is also supported (as detailed below). Windows has a notorious reputation where remote WMI is concerned. It tends to either "just work" or some mystery item on the target requires changing. If you are experiencing difficulty auditing remote Windows PCs, we have created a script called test_windows_client.vbs. You can run this script LOCALLY on the machine in question, after signing on as the user that is used by Open-AudIT to perform the audit. The script makes NO CHANGES to the target system. It checks most (not all) of the items below and generates PASS, FAIL and INFO tags for various properties. NOTE - If your target system is being audited correctly, you should not change any settings. Some of the below don't exist on Windows PCs that are able to be audited and some do exist. Only change settings if yours audits on particular PCs are not working as intended.

In addition to the below, other items that should be checked are the time between the client and the domain controller and that DNS is resolving correctly (both forwards and backward).

Some users have also stated that removing and then rejoining the client PC to the domain has enabled auditing to work.

Note - All commands below should be entered into an elevated command prompt. To do this click Start -> All Programs -> Accessories -> Command Prompt, right click then "Run as administrator".

Microsoft Article, Connecting to WMI Remotely Starting with Windows Vista - http://msdn.microsoft.com/en-us/library/aa822854(v=vs.85).aspx

Another good Microsoft article, https://msdn.microsoft.com/en-us/library/aa826699%28v=vs.85%29.aspx?f=255&MSPPError=-2147217396

User Credentials Requirements

To audit a machine, you must have credentials and administrator level access.

- To audit a remote machine that is not in a domain, you must use the Administrator account (not "an" admin account, "the" Administrator account) on the target PC. ** and #1
- To audit a remote machine on an Active Directory domain, your supplied user (or if none provided, the user running the script) must be a member of the target machines Administrators group (or subgroup).
- To audit localhost, any supplied credentials are disregarded and the connection is made using the details of the user running the script.
- The account must have a password; WMI does not allow blank passwords.
- The account password must NOT contain " (double quotes). This is because cscript (and wscript) cannot parse argument values containing double quotes. They are simply stripped. No (before you ask) escaping will not work. This is a cscript limitation and nothing to do with Open-AudIT.

** NOTE - To enable a remote machine (Vista or above) to be audited that is not on a domain, by an account in the Administrators group, other than the actual Administrator account see the below section on UAC.

#1 NOTE - You *can* audit a remote machine without the using the actual Administrator account by creating a registry key. Create the below key on every machine to be audited and make sure the user credentials used are in the Administrators group.

```
\HKEY_USERS\.DEFAULT\Software\Microsoft\Windows Script\Settings
```

Enabling the Administrator account on non-Domained machines

Open a command prompt with administrative rights (Windows key, type 'cmd' (sans quotes) right click Command Prompt and select Run as Administrator.

In the command window type

```
net user Administrator
```

You should see Active set to false. Enable it with

```
net user Administrator /active:yes
```

Then run the first command again and confirm Active is now set. Then set the password with:

```
net user Administrator *
```

And type 'exit' to close the window.

Windows 7 and 2008 R2 not submitting audit using HTTPS

We have been advised that some Windows 7 and Windows 2008 R2 machine will not submit their audit result to the Open-AudIT server when running HTTPS. If this affects you , please see the following Microsoft article - https://support.microsoft.com/en-us/help/3140245/update-to-enable-tls-1-1-and-tls-1-2-as-default-secure-protocols-in-wi

DCOM

Run the DCOM utility and verify (or set) the below attributes. Start -> Run, Enter DCOMCNFG and press OK. This will open the DCOMCNFG window.

Browse down the tree to Console Root -> Component Services -> Computers -> My Computer

Right click on "My Computer" and select properties

Select the "Default Properties" tab

- Enable Distributed COM on this computer - Option is checked
- Default Authentication Level - Set to Connect
- Default Impersonation Level - Set to Identify

Select the "COM Security" tab

Click on Access Permissions ' Edit Default

- Add "Anonymous", "Everyone", "Interactive", "Network", "System" with Local and Remote access permissions set.

Click on Launch and Activation Permissions ' Edit Default

- Add "Anonymous", "Everyone", "Interactive", "Network", "System" with Local and Remote access permissions set.

Click on OK and close the DCOMCNFG window.

The above changes will require a reboot to take effect.

UAC

If you are getting an Access Denied scan error it might be UAC blocking inbound requests on the remote device. If the remote computer you are trying to query is in a workgroup (or not joined to a domain), UAC prevents remote queries by default, even if the account being used is in the Administrators group. Completely disabling UAC on the remote device allows you to get around this, but it is preferable to disable the subcomponent of UAC instead. You can do this by adding or editing this registry key on the remote device you are scanning and setting its value to 1:

HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System\LocalAccountTokenFilterPolicy

Note that Windows 8 and Server 2012 do not have a way to completely disable UAC (adjusting the slider in the GUI just disables notifications). You'll need to use the registry key method.

You can use this command from a command prompt on the remote device to quickly add the registry key:

```
reg add HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System /v
LocalAccountTokenFilterPolicy /t REG_DWORD /d 1 /f
```

Reference - https://support.microsoft.com/kb/942817

The above change will require a reboot to take effect.

Local Security Policies

Run one of the following three Microsoft Management Console (MMC) snap-ins:

- the Local Security Policy snap-in (secpol.msc) for member servers, or
- the Default Domain Security Policy snap-in (dompol.msc) if you wish to configure these settings domain-wide as a GPO, or
- the Default Domain Controller Security Settings snap-in (dcpol.msc) if you wish to assign the rights only on domain controllers.

Expand Security Settings -> Local Policies -> User Rights Assignment.

Check the Administrators Group has at least the following rights:

- Act as part of the operating system
- Log on as a batch job
- Log on as a service
- Replace a process level token

Go to Start -> Control Panel -> Administrative Tools -> Local Security Policy

Navigate to Security\Local Policies\Security Options

- DCOM: Machine Access Restrictions - Add Anonymous, Everyone, Interactive, Network, System with full rights options set.
- Network Access: Let everyone permissions apply to anonymous users - Set to Enabled
- Network Access: Sharing security model for local accounts - Set to Classic
- User Account Control: Run all Administrators in Admin Approval Mode - Set to Disabled

The above changes will require a reboot to take effect.

Simple File Sharing (XP)

Windows XP Professional computers in a workgroup environment will need simple file sharing disabled. You can make this change through the registry by setting the following key to a value of 0.

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\ForceGuest

If this key does not exist, you can add it using a command prompt by:

```
reg add HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa /v ForceGuest /t REG_DWORD /d 0 /f
```

You can also do this through settings by performing the following steps:

Click Start Control Panel Folder Options.

Select the View tab and scroll to the bottom of the Advanced Settings: section.

Uncheck the Use simple file sharing (Recommended) to disable the option and click the OK button.

SSPI means Security Support Provider Interface and is the interface used by VBscript / WMI to validate the user.

- If ForceGuest is enabled (set to 1), SSPI will always try to log on using the Guest account.
- If the Guest account is enabled, an SSPI logon will succeed as Guest for any user credentials.
- If the Guest account is disabled, an SSPI logon will fail even for valid credentials.
- If ForceGuest is disabled (set to 0), SSPI will log on as the specified user.

The above changes will require a reboot to take effect.

WMI

Windows WMI (Windows Management Interface) is used by the audit script for most of it's information retrieval. WMI can (at times) become corrupted. Microsoft have released a tool to enable you to check for this corruption.

The tool is available from Microsoft, here - http://www.microsoft.com/en-au/download/details.aspx?id=7684

Using the tool is detailed here - http://blogs.technet.com/b/askperf/archive/2012/02/03/wmidiag-2-1-is-here.aspx

For ***Windows Core*** servers, ensure you allow the firewall connections as per - http://blogs.technet.com/b/brad_rutkowski/archive/2007/10/22/unable-to-remotely-manage-a-server-core-machine-mmc-wmi-device-manager.aspx

Networking with WMI

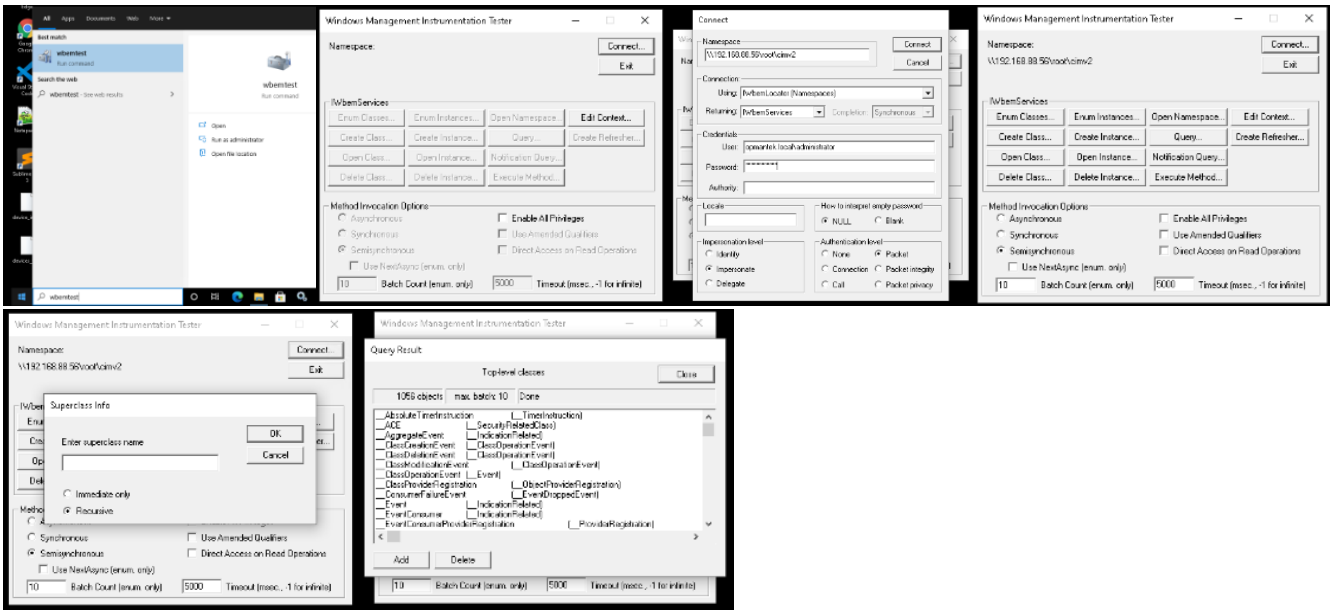
By default, Windows sends WMI data over random ports, as explained in this Microsoft knowledge base article. You need to either:
• Configure your firewalls in such a way that ***all*** WMI traffic (over random ports) is allowed. Windows Firewall includes a remote administration exception that you can enable to allow WMI traffic, as explained in this knowledge base article. For third-party firewalls, you'll need to consult your firewall documentation.
• Configure a fixed WMI port and allow traffic through that port. Setting up a fixed port is supported by Windows Vista and more recent operating systems.

Testing Remote WMI

When testing remote WMI you should disable the Windows Firewall, Antivirus, and UAC. Test as below. If it succeeds, begin by enabling one of the disabled items at a time, test and if it works, enable the next item. If any failed connections occur, it is likely the component you have just enabled. Fix that component and try again.

To test WMI remote network connections.

1. RDP to the another Windows machine (if running Open-AudIT on Linux), or if you're running Open-AudIT on Windows, the Open-AudIT server.
2. Click Start menu, then type: **wbemtest**
3. Open the utility.
4. Click **Connect...**
5. Enter the information
    a. Within namespace, type: **TARGET_DEVICE_IP**\root\cimv2 - where TARGET_DEVICE_IP is the IP of the device not being discovered by Open-AudIT.
    b. For the Username, use the format domain\username
    c. Enter the password
    d. Leave the Authority field blank.
6. Click connect
7. Click Enum Classes and select Recursive, then click OK
8. It may take several seconds to respond.
    a. If it ***does not*** work, the target PC has an issue (or something between the target PC and Open-AudIT).
    b. if it ***does*** work, please log a support ticket with Opmantek.



Testing Remote WMI #2

Open a command prompt and try the following command. Substitute your domain, username, password and target IP.

You should get a result similar to the below.

NOTE - the extra r's and for the (r) registered trademark symbols, no need to be concerned with them.

```
C:\Users\opDev>wmic /user:YOUR_DOMAIN\YOUR_USERNAME /password:YOUR_PASSWORD /node:YOUR_IP os get name
Name
Microsoftr Windows Serverr 2008 Enterprise |C:\Windows|\Device\Harddisk0\Partition1
```

If the response is: ***Description: RPC Server is unavailable***, then you have a firewall or other issue.

If the response is: ***Description: Access Denied Facility = Win32*** then the credentials that were supplied don't have Windows DCOM permissions on the Target machine.

If the response is: ***Description: Access denied Facility = WMI*** then the credentials that were supplied don't have WMI Security permissions on the Target machine.

Matching Discovery Logs to WMI issues

If you see the below, try the following fixes.

ERROR: Failed to open connection - NT_STATUS_LOGON_FAILURE

Check your credentials and that they are of a machine Administrator account.

ERROR: Failed to open connection - NT_STATUS_CONNECTION_RESET

Likely from our attempt to use SMB1, which the target Windows PC no longer accepts.

ERROR: Failed to save ADMIN$/winexesvc.exe - NT_STATUS_ACCESS_DENIED.

Are the ADMIN$ and IPC$ shares enabled? Check as below.

ERROR: UploadService failed - NT_STATUS_ACCESS_DENIED.

Are the ADMIN$ and IPC$ shares enabled? Check as below.

ERROR: Failed to install service winexesvc - NT_STATUS_ACCESS_DENIED

This most likely means the user account being used does not have sufficient rights on the target machine. To fix this issue, see the section above on this page for **UAC**.

ERROR: StartService Failed - NT_STATUS_ACCESS_DENIED

We are still investigating possible causes for this issue. It appears that the winexesvc.exe file has been copied to the target and the service registered, however it fails to start. This *may* be Antivirus related. We are unsure at this stage.

Winexe requirements (Linux only) on Windows machines

Enabled services: Workstation, Server.

"Windows Network" is running and "Printer and File Sharing" are activated.

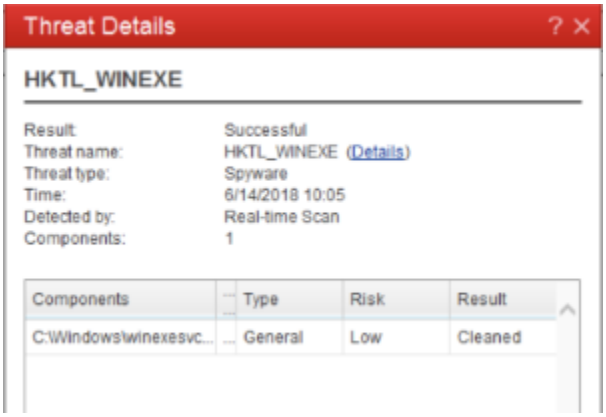
Enabled "Remote IPC" and "Remote Admin" shares. To verify it, in cmd box run command "net share", and check if there are ADMIN$ and IPC$ shares.

An account with administrative privileges and not empty password. If Windows machine is not on a domain, it is best to use **the** Administrator account (see above).

Firewall rules allowing traffic between both machines.

AntiVirus

Some antivirus programs have been known to disable DCOM and remote WMI. You might check the settings of your antivirus program and disable them for testing. We recently had a report of Trend AV specifically blocking calls to winexesvc when auditing Windows computers.



Windows Firewall

To enable remote PCs to be audited, either the local (on the target machines) firewall (likely the Windows Firewall) must be disabled or access allowed for the WMI service.

For Windows Vista, 7, 8, 2008 and 2012, enter the following commands:

```
netsh firewall set service type=remoteadmin mode=enable
netsh advfirewall firewall set rule group="windows management instrumentation (wmi)" new enable=yes
netsh advfirewall firewall set rule group="remote administration" new enable=yes
netsh advfirewall firewall set rule name="File and Printer Sharing (Echo Request - ICMPv4-In)" new enable=yes
```

Linux

On Linux, Open-AudIT uses SSH as it's primary method of auditing. SNMP is also supported (and detailed below).

The user used to audit a Linux host should be root or have sudo access. Some distributions do not allow sudo over an SSH session without a terminal. On these distributions, root can be used.

Using a user account that has no sudo access (and is not root) will result in an audit without all possible attributes retrieved.

Sudo / Root is required for:

dmidecode
(system - uuid, serial, form_factor),
(bios - version, smversion, serial),
(processor - socket),
(memory - all)

netstat
(netstat - program name where not owned by user running script)

NOTE - Running as different users will generate a different list of environment variables.

SSHGuard

If you find your target machine is not correctly being audited, check to make sure SSHGuard has not been triggered. Below you can see the IP 192.168.1.179 has been blocked by SSHGuard (see last line).

```
root@desktop:/# iptables -L -n -v
Chain INPUT (policy ACCEPT 77217 packets, 48M bytes)
 pkts bytes target     prot opt in     out     source               destination
  35M   43G sshguard   all  --  *      *       0.0.0.0/0            0.0.0.0/0

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 77202 packets, 48M bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain sshguard (1 references)
 pkts bytes target     prot opt in     out     source               destination
   77  6368 DROP       all  --  *      *       192.168.1.179        0.0.0.0/0
```

To enable your Open-AudIT server, edit the file /etc/sshguard/whitelist and restart the sshguard service.

```
sudo nano /etc/sshguard/whitelist

sudo service sshguard restart
```

ESX

On VMware ESX, Open-AudIT uses SSH as it's primary method of auditing. SNMP is also supported (and detailed below).

OSX

On OSX, Open-AudIT uses SSH as it's primary method of auditing. SNMP is also supported (and detailed below).

The OSX audit script should be run by root or using sudo access.

AIX

On AIX, Open-AudIT uses SSH as it's primary method of auditing. SNMP is also supported (and detailed below).

The AIX audit script should be run by root.

Solaris

On Solaris, Open-AudIT uses SSH as it's primary method of auditing. SNMP is also supported (and detailed below).

The Solaris audit script should be run by root.

SNMP

SNMP v1, v2c and v3 are supported. Read access is required.