

Troubleshooting device connectivity with NMIS

- [Troubleshooting with Ping](#)
- [Troubleshooting SNMP](#)
 - [Check for opened port.](#)
 - [Test SNMP Response](#)
- [Troubleshooting WMI](#)
 - [Test WMI availability and credentials](#)

For NMIS to be able to obtain information from a network device, basically, it must comply with 2 conditions.

1. The device itself must be reachable or available on the network.
2. The device must be capable to provide management information via SNMP or WMI

Troubleshooting with Ping

Ping is used to check the availability of a host in the network, it sends out an echo request (ICMP message) and waits for a reply.

```
$ ping <ip_address>
```

```
root@vpnServer:~# ping 192.168.1.200
PING 192.168.1.200 (192.168.1.200) 56(84) bytes of data.
From 192.168.1.30 icmp_seq=1 Destination Host Unreachable
From 192.168.1.30 icmp_seq=2 Destination Host Unreachable
From 192.168.1.30 icmp_seq=3 Destination Host Unreachable
From 192.168.1.30 icmp_seq=4 Destination Host Unreachable
From 192.168.1.30 icmp_seq=5 Destination Host Unreachable
From 192.168.1.30 icmp_seq=6 Destination Host Unreachable
^C
--- 192.168.1.200 ping statistics ---
8 packets transmitted, 0 received, +6 errors, 100% packet loss, time 6999ms
pipe 4
root@vpnServer:~# ping 192.168.1.100
PING 192.168.1.100 (192.168.1.100) 56(84) bytes of data.
64 bytes from 192.168.1.100: icmp_seq=1 ttl=64 time=12.5 ms
64 bytes from 192.168.1.100: icmp_seq=2 ttl=64 time=5.49 ms
64 bytes from 192.168.1.100: icmp_seq=3 ttl=64 time=5.49 ms
64 bytes from 192.168.1.100: icmp_seq=4 ttl=64 time=7.49 ms
64 bytes from 192.168.1.100: icmp_seq=5 ttl=64 time=6.91 ms
64 bytes from 192.168.1.100: icmp_seq=6 ttl=64 time=6.21 ms
64 bytes from 192.168.1.100: icmp_seq=7 ttl=64 time=6.40 ms
^C
--- 192.168.1.100 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6009ms
rtt min/avg/max/mdev = 5.490/7.220/12.526/2.268 ms
root@vpnServer:~# _
```

A device that is not reachable won't be able to respond to the ping and a "Destination Host Unreachable" will be returned instead.

If this is the case:

- Proper physical connection should be verified.
- Firewall Policies should be reviewed to verify that ICMP messages are allowed.



Firewalls can be configured to block packets from ping. If a remote host does not respond to ping requests, it is possible that it is up and running normally, but ignoring ping requests.

Troubleshooting SNMP

In order to be able to obtain information from the monitored device via SNMP, we must verify that the device is correctly configured for this purpose. There is a checklist with the most common items to check on the device side:

- Is SNMP enabled on the device?, some devices allow us to perform pre-configuration of SNMP without enabling it.
- if using SNMP v1 or v2, check if the device is using the correct community string.
- if using SNMP v3, check if the device is using the correct username, privpass and authpass, it is also recommended to check for the correct SNMP Authentication and Privacy Protocols.
- Firewall Policies should be reviewed to verify that SNMP messages are allowed.

Check for opened port.

With the device correctly configured, we can troubleshoot the SNMP connection to the device from the server side, to do this, SSH to your NMIS server, it is important to do this from the NMIS server itself because it ensures that any access control you have from Firewalls or other security controls is part of the testing.

First we check if the port used by SNMP is open, usually it is port UDP 161. It is possible to test it with NMAP or any other port testing tool.

```
$ nmap -sU -p 161 <ip_address>
```

```
root@idontknow:/# nmap -sU -p 161 192.168.8.161
Starting Nmap 7.40 ( https://nmap.org ) at 2019-08-28 15:11 AEST
Nmap scan report for 192.168.8.161
Host is up (0.00032s latency).
PORT      STATE SERVICE
161/udp    closed snmp
MAC Address: 00:0C:29:15:29:1C (VMware)
Nmap done: 1 IP address (1 host up) scanned in 0.58 seconds

root@idontknow:/# nmap -sU -p 161 192.168.8.57
Starting Nmap 7.40 ( https://nmap.org ) at 2019-08-28 15:11 AEST
Nmap scan report for 192.168.8.57
Host is up (0.00018s latency).
PORT      STATE SERVICE
161/udp    open  snmp
Nmap done: 1 IP address (1 host up) scanned in 0.31 seconds

root@idontknow:/#
```

If the port is closed, NMIS won't be able to get any information from the device using SNMP, if this is the case:

- Check for Firewall configuration, port UDP 161 must be allowed.
- Check SNMP configuration, it may be the case that the port used for SNMP purposes it's not the default.

Test SNMP Response

Once the opened port has been verified, it is time to check if we get a response from the device via SNMP using SNMPwalk.

We have a guide detailing how to do this, in the following Link for SNMP v1 and v2: [Testing SNMP Connectivity from the NMIS Server with snmpwalk](#) and for SNMP v3: [Using SNMPv3 with NMIS for Secure Network Management](#)

Troubleshooting WMI

To collect WMI data from a device, NMIS has to use a WMI access tool. There is a checklist with the most common items to check on the device side:

- Is the WMI service running?
- Network and firewalls must be configured to let WMI accesses pass.
- WMI accesses are generally negotiated to use dynamic ports (following up on an initial conversation on TCP port 135)

Newer versions of Windows that are fully patched may run into connection/access issues with the version of wmic that ships with NMIS. An error like the following indicates this problem is occurring:

```
librpc/rpc/dcerpc_util.c:1290:dcerpc_pipe_auth_recv()] Failed to bind to
uuid 4d9f4ab8-7d1c-11cf-861e-0020af6e7c57 - NT_STATUS_NET_WRITE_FAULT
[librpc/rpc/dcerpc_connect.c:790:dcerpc_pipe_connect_b_recv()] failed NT
status (c0000022) in dcerpc_pipe_connect_b_recv
[wmi/wmic.c:196:main()] ERROR: Login to remote object.
NTSTATUS: NT_STATUS_ACCESS_DENIED - Access denied
```

A replacement wmic application is available which implements the required security protocols to allow wmi to continue working.

```
#download wmic.py from https://github.com/simply42/check_wmi_plus_wmic_dropin
#copy wmic.py onto box
scp wmic.py user@box:~/

# make sure python, pip and the scripts deps are available, apt instructions would be very similar
sudo yum install python3
sudo pip3 install --upgrade pip
sudo pip3 install impacket

#move old wmic out of the way and put in the new one
sudo mv /usr/local/nmis9/bin/wmic /usr/local/nmis9/bin/wmic-orig-2023-05-23
sudo mv ~/wmic.py /usr/local/nmis9/bin/
sudo ln -s /usr/local/nmis9/bin/wmic.py /usr/local/nmis9/bin/wmic
sudo chown nmis:nmis /usr/local/nmis9/bin/wmic.py
sudo chmod u+x /usr/local/nmis9/bin/wmic.py

# verify wmic runs:
sudo /usr/local/nmis9/bin/wmic
#usage: wmic [-h] [-U USERNAME] [-A AUTHFILE] [-v] [-n NAMESPACE]
#          [-d DELIMITER]
#          host query
#wmic: error: the following arguments are required: host, query
```

Test WMI availability and credentials

The test can be perform using the wmic program found in: /usr/local/nmis8/bin and the credential for the device.

```
$ /usr/local/nmis8/bin/wmic -U somewmiuser --password='somewmipassword' //testserver "select Caption,
Manufacturer,Model,Name from Win32_ComputerSystem"

CLASS: Win32_ComputerSystem
Caption|Manufacturer|Model|Name
TestServer|VMware, Inc.|VMware Virtual Platform|TestServer
```

Additional information can be found here:

- [How NMIS interfaces with WMI-based devices](#)
- [Using WMI to query and monitor Windows devices](#)