



# **SNMP Setup Guide Eli-v6.4.84**

---

## **Bridgeworks**

Unit 1, Aero Centre, Ampress Lane,  
Ampress Park, Lymington,  
Hampshire SO41 8QF  
Tel: +44 (0) 1590 615 444  
Email: [support@4bridgeworks.com](mailto:support@4bridgeworks.com)

---

# Table of Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Overview . . . . .	3
1.2	Types of SNMP . . . . .	3
1.2.1	SNMPv1 . . . . .	3
1.2.2	SNMPv2c . . . . .	3
1.2.2.1	Using SNMPv2c . . . . .	4
1.2.3	SNMPv3 . . . . .	4
<b>2</b>	<b>Definitions</b>	<b>5</b>
<b>3</b>	<b>Requirements</b>	<b>6</b>
3.1	Net-SNMP . . . . .	6
3.1.1	Using Net-SNMP on Windows . . . . .	6
<b>4</b>	<b>SNMPv2c</b>	<b>7</b>
4.1	Enabling SNMP Agent . . . . .	7
4.2	Querying the Agent . . . . .	8
4.2.1	Net-SNMP CLI . . . . .	8
4.2.2	SNMPb GUI . . . . .	9
<b>5</b>	<b>SNMPv3</b>	<b>17</b>
5.1	Enabling SNMP Agent: noAuthNoPriv mode . . . . .	17
5.1.1	Querying the Agent . . . . .	19
5.1.1.1	Net-SNMP CLI . . . . .	19
5.1.1.2	SNMPb GUI . . . . .	20
5.2	Enabling authentication: authNoPriv Mode . . . . .	29
5.2.1	Querying the Agent . . . . .	30
5.2.1.1	Net-SNMP CLI . . . . .	30
5.2.1.2	SNMPb GUI . . . . .	31
5.3	Enabling privacy: authPriv Mode . . . . .	33

---

5.3.1	Querying the Agent	34
5.3.1.1	Net-SNMP CLI	34
5.3.1.2	SNMPb GUI	35
<b>6</b>	<b>SNMP Traps</b>	<b>37</b>
6.1	Types of Traps	37
6.2	Adding a trap sink	37
6.3	Receiving a trap	40
6.3.1	Net-SNMP CLI	40
6.3.2	SNMPb GUI	42
<b>7</b>	<b>MIBs</b>	<b>44</b>
7.1	SNMP-MIBs-Downloader	44
7.2	Bridgeworks MIBs	45
<b>8</b>	<b>Useful Links</b>	<b>46</b>

---

# 1 Introduction

## 1.1 Overview

This guide is intended to instruct a user through setting up the SNMP agent entity on a Bridgeworks product. It will cover enabling and implementing settings, along with examples of connecting to the SNMP agent entity from the *Net-SNMP* command line tools and the *SNMPb* GUI browser software.

## 1.2 Types of SNMP

SNMP (Simple Network Management Protocol) is a high level protocol primarily used to monitor nodes in a network. One or more *Manager* entities will request information from one or more *Agent* entities. Unsolicited messages can be sent by an *Agent* to a *Manager*. Messages that the manager did not expect are known as: *Traps*, *Notifications*, or *Events*.



Note: Traps is the most consistently used term across all major SNMP versions, as such it will be used for this guide.

There are currently 3 significant versions of SNMP:

- SNMPv1
- SNMPv2c
- SNMPv3

### 1.2.1 SNMPv1

The first and most basic implementation of the SNMP protocol. This version is out of date and generally not suitable for any environment.

SNMPv1 contains no privacy or authentication systems. All SNMPv1 traffic is plain-text and easily observed by any third party with access to the network.

SNMPv1 employs basic access control through the use of the *community name*. This feature is similar to password controlled access, but lack of authentication or privacy means that this only offers minimal security.

This version is not officially supported.

### 1.2.2 SNMPv2c

SNMPv2c is a functionality upgrade built on top of SNMPv1, with an expanded command set and improved back end syntax. A significant improvement is the ability for agents to send a trap message that requests the manager confirm its receipt. This new message type is known as an *inform*.

---

SNMPv2c retains the basic access control used by SNMPv1. The *community name* feature is similar to password controlled access, but lack of authentication or privacy means that this only offers minimal security.

### 1.2.2.1 Using SNMPv2c

Because of the vulnerabilities associated with SNMPv2c it is good practice to assess whether it is suitable for use in your network.

General criteria for using SNMPv2c:

- All manager and agent entities are located on an internal network and do not route SNMP traffic outside of this domain.
- All agent entities are providing read only information that does not contain any private information.
- If an agent entity does have writeable data then those elements should not be capable of causing breaches of security. For instance, having a writeable reset request entry would cause inconvenience, but not leak private information.

There are practices for securing SNMPv2c communication externally, such as private network connections for exclusive SNMP traffic, but these fall outside the scope of this guide.

### 1.2.3 SNMPv3

This is the recommended version of SNMP to use. It requires more setup time and is more complicated to implement than previous SNMP versions, but adds much more security through authentication and privacy systems.

SNMPv3 is identical to SNMPv2c from a functionality perspective, message contents and unsolicited messages all work identically between the two versions. The most significant development of SNMPv3 was the hardening of the framework by building hashing based authentication systems and encryption cipher protocols directly into the package.

With this added security SNMPv3 becomes a secure standalone system that can be deployed in networks that would otherwise have been too vulnerable for any sensitive information to traverse.

---

## 2 Definitions

**Entity** SNMPv3 moves over from Agent and Manager to simply SNMP 'entity'. An entity is any system that contains the SNMP engine and some sort of SNMP applications. The SNMP engine is common across all entities, whilst applications specialise the entity and may vary between entities based on individual requirements of the system.

**Agent** The SNMPv1/2c name for an SNMP system that only responds to queries and sends out traps. It does not query other SNMP entities.

**Manager** The SNMPv1/2c name for an SNMP system that sends queries and receives responses from agents. In addition it listens for and receives traps from agents. Typically the manager can be extended with a variety of MIB files to add compatibility with more agents as needed by the user.

**Agent entity** This guide uses Agent entity to mean an SNMP entity that can only fulfill the role of responding to requests and sending traps. It does not perform any management style querying and does not relay any information.

**Manager entity** This guide uses Manager entity to mean an SNMP entity that fulfills the role of querying agents, or agent entities, and then processes the response. It also listens for and receives trap information.

**MIB** Management Information Base - These are structured database entries that contain information about the hierarchical layout of the Objects in the database. Where applicable, they also contain specific information about how to describe the value the object links to.

**OID** Object Identifier - These are addresses to a specific entry in the hierarchical database tree. They are the literal address that is interpreted by MIB objects.

**NoAuthNoPriv** "No Authorisation No Privacy" An unprotected SNMPv3 connection, functionally identical to an SNMPv2c connection.

**AuthNoPriv** "Authorisation No Privacy" An authenticated SNMPv3 connection. Uses hashing to prove the contents of the message have not been tampered with and the message comes from an entity that has a trusted authentication password. Does not provide any privacy.

**AuthPriv** "Authorisation and Privacy" A secure SNMPv3 connection. In addition to the authentication (see *AuthNoPriv*), the SNMP messages are now encrypted to make them private.

---

## 3 Requirements

To establish and test an SNMP connection the user requires a single Bridgeworks unit and a single system capable of running an SNMP manager.

The Net-SNMP command line tools and SNMPb GUI manager software used in this guide are available for both Windows and Linux. Mac operating systems are outside of the scope of this guide.

It is assumed that you have your unit running with a network connection and have access to the GUI on a web browser.



Note: The Bridgeworks unit does not act as a Manager.

### 3.1 Net-SNMP

Net-SNMP is an open source project that provides complete support for all major SNMP versions.

See source downloads at:	<a href="http://www.net-snmp.org/download.html">http://www.net-snmp.org/download.html</a>
Pre-compiled Linux tarballs and Windows executables available from:	<a href="https://sourceforge.net/projects/net-snmp/files/">https://sourceforge.net/projects/net-snmp/files/</a>



Note: Compiling from source code is not in the scope of this guide.

#### 3.1.1 Using Net-SNMP on Windows

To use these tools in Windows you should be comfortable using a command prompt or powershell terminal. The installer executable will prompt for a location to install to. The command line tools are unpacked to this location.

The bin folder this installer creates should be linked in your PATH environment system variable, otherwise you need to navigate your terminal to its location and run the commands locally.



Important: At time of writing the Net-SNMP download page (<http://www.net-snmp.org/download.html>) states that pre-compiled Windows binaries will fail to install on systems with OpenSSL 1.0 or greater.

---

# 4 SNMPv2c

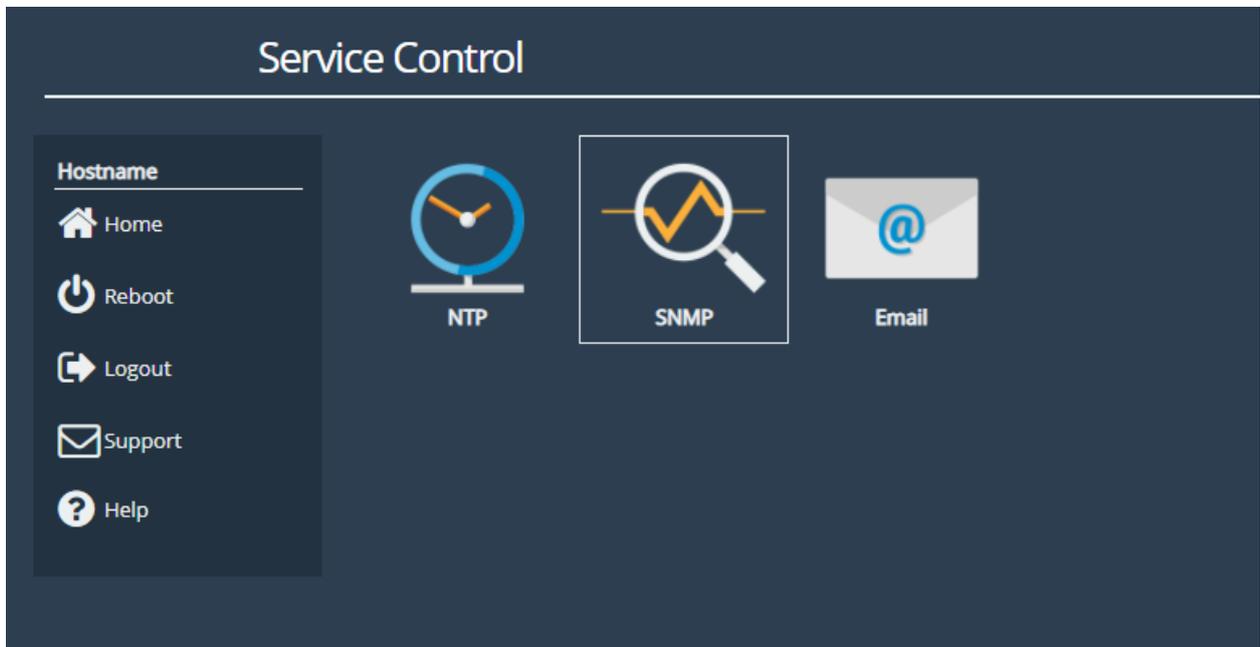
## 4.1 Enabling SNMP Agent

Log in to the target Bridgeworks unit.

Left click on the *Service Control* icon.



Then left click on the *SNMP* icon.



From this page check the box left of the *SNMP v2c Agent* title.

A configuration form for the "SNMP v2c Agent". It has a checkbox labeled "SNMP v2c Agent" which is checked. Below it is a "Community Name:" label followed by a text input field containing the word "public". At the bottom right is a "Save" button.

Provide a *Community Name* and left click on the *Save* icon at the bottom right of the *SNMP v2c Agent* section.



At this stage the SNMP Agent is now running and accessible.

## 4.2 Querying the Agent

### 4.2.1 Net-SNMP CLI

To receive SNMP information from the SNMP Agent, Net-SNMP users can simply run `snmpwalk` to get the entire tree.

Be sure to get the correct community string, otherwise, the agent will fail silently as per the example below.

```
C:\usr\bin>.\snmpwalk.exe -v2c -c demo 10.10.64.24
Timeout: No Response from 10.10.64.24
```

A `snmpwalk` will have a lot of response information, this example does not show the full response.

'...' denotes gaps where multiple lines of the SNMP response were cut.

```
C:\usr\bin>.\snmpwalk.exe -v2c -m ALL -c bridgeworks 10.10.64.24
...
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (44901882) 5 days, 4:43:38.82
SNMPv2-MIB::sysContact.0 = STRING: support@4bridgeworks.com
SNMPv2-MIB::sysName.0 = STRING: bw-pr400-snmp
SNMPv2-MIB::sysLocation.0 = STRING: Bridgeworks Unit
IF-MIB::ifNumber.0 = INTEGER: 2
IF-MIB::ifIndex.1 = INTEGER: 1
IF-MIB::ifIndex.2 = INTEGER: 2
IF-MIB::ifDescr.1 = STRING: lo
IF-MIB::ifDescr.2 = STRING: port1
IF-MIB::ifType.1 = INTEGER: softwareLoopback(24)
IF-MIB::ifType.2 = INTEGER: ethernetCsmacd(6)
IF-MIB::ifMtu.1 = INTEGER: 65536
IF-MIB::ifMtu.2 = INTEGER: 1500
IF-MIB::ifSpeed.1 = Gauge32: 10000000
IF-MIB::ifSpeed.2 = Gauge32: 1000000000
...
TCP-MIB::tcpConnectionState.ipv4."10.10.64.24".443.ipv4."10.0.80.136".62933 =
INTEGER: established(5)
...
TCP-MIB::tcpConnectionProcess.ipv4."10.10.64.24".443.ipv4."10.0.80.136".62933 =
```

---

```
Gauge32: 403
...
IF-MIB::ifPromiscuousMode.1 = INTEGER: false(2)
Timeout: No Response from 10.10.64.24
```

```
C:\usr\bin>
```

In the response above the SNMP manager has interpreted the OID (Object Identifiers) values and replaced them with relevant names for each entry.

These names are from MIBs (Management Information Bases), which contain descriptive information about their respective OID value. The `-m ALL` in the command includes the MIBs.

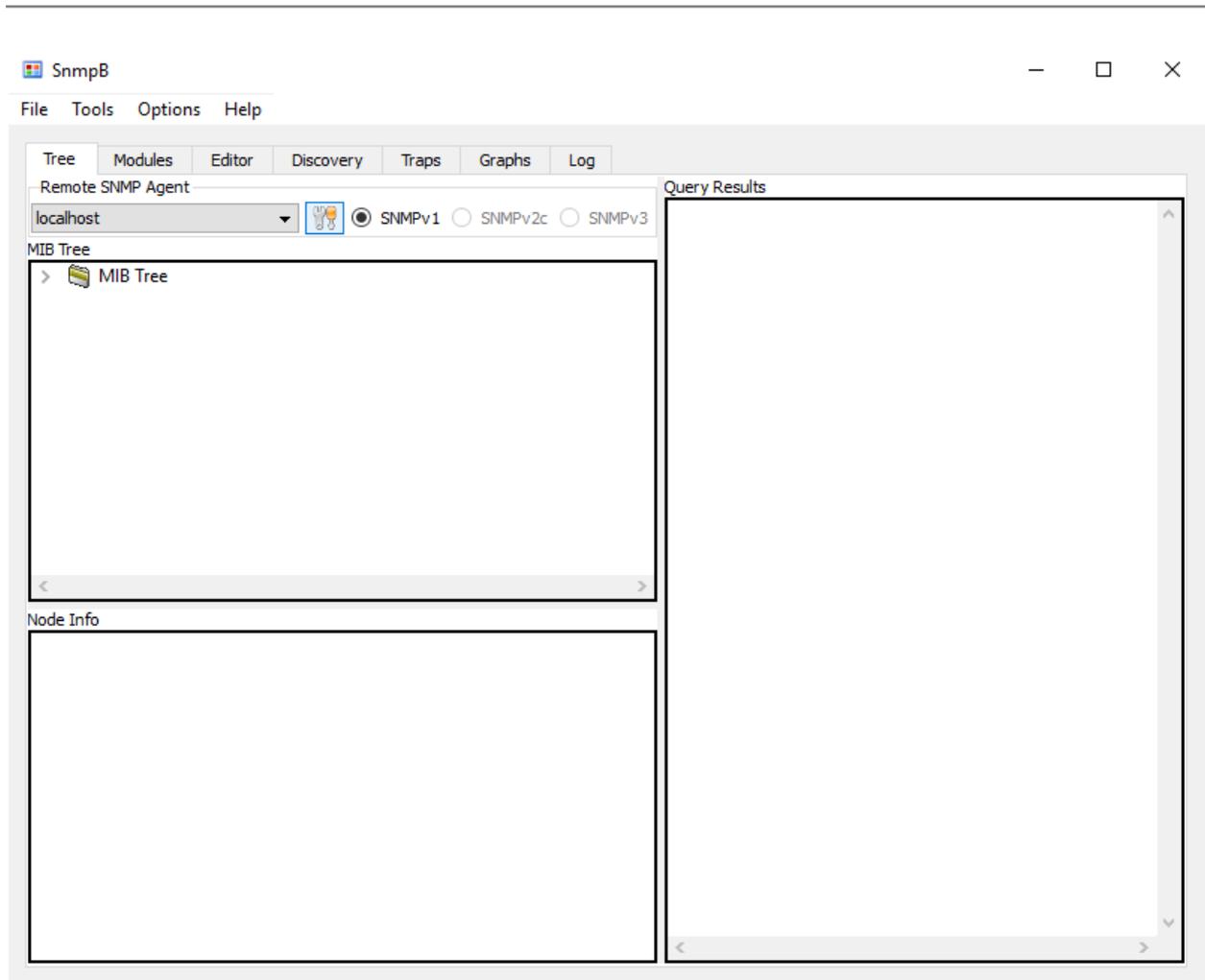


Note: For further use of `snmpwalk` and other Net-SNMP commands such as: `get`, `snmpget`, `snmpgetnext`, and `snmpbulkget`, see the Net-SNMP wiki page (<http://www.net-snmp.org/wiki/>).

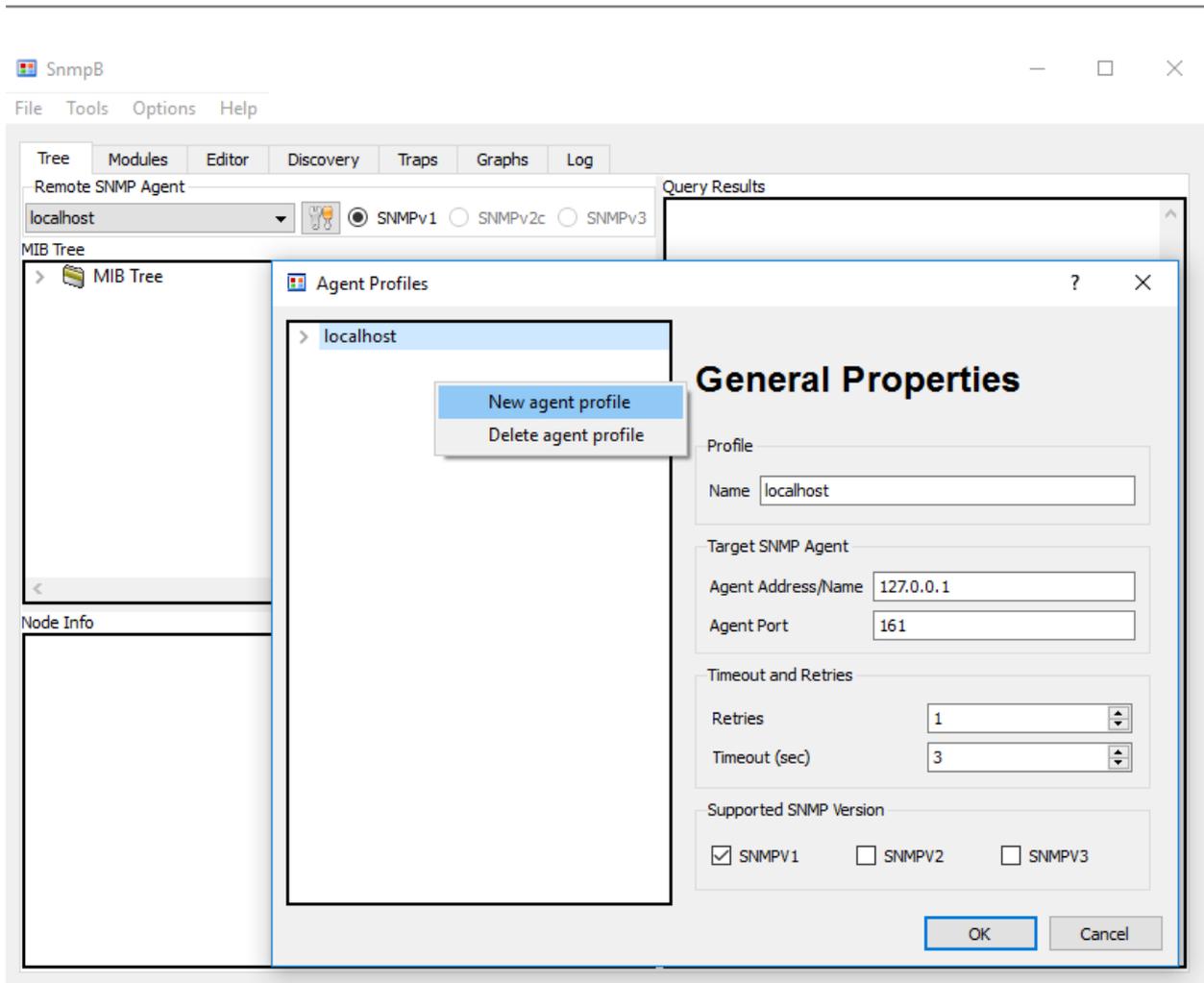
## 4.2.2 SNMPb GUI

With SNMPb there is more overhead in setting up the initial connection than an equivalent command line query.

With SNMPb open, left click the settings icon in the *Remote SNMP Agent* section.

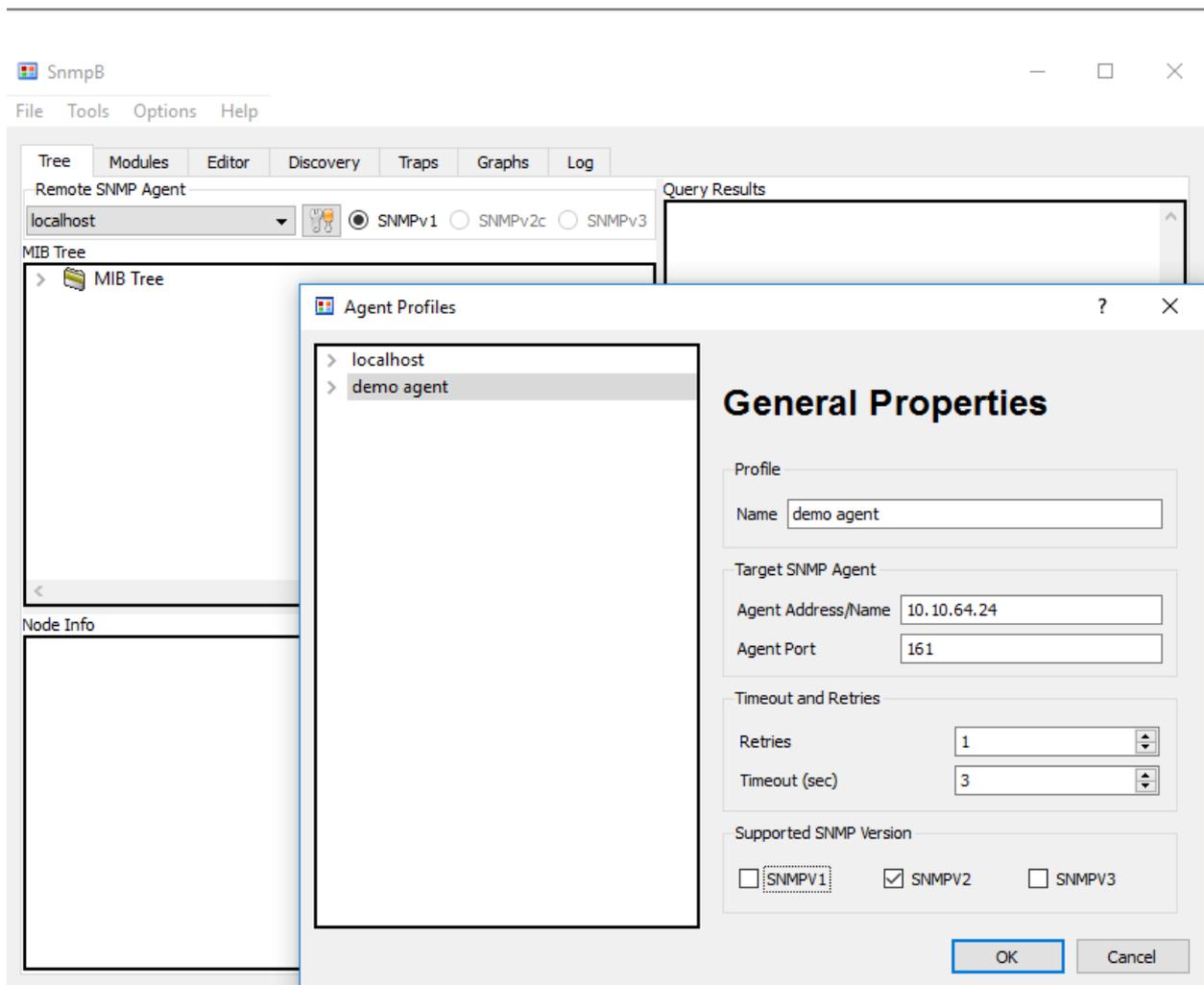


In the new window right click in the profile selection pane on the left. Then left click on *New agent profile*.



Now enter information relevant to your setup. The *Name* entry in this section is for descriptive purposes only.

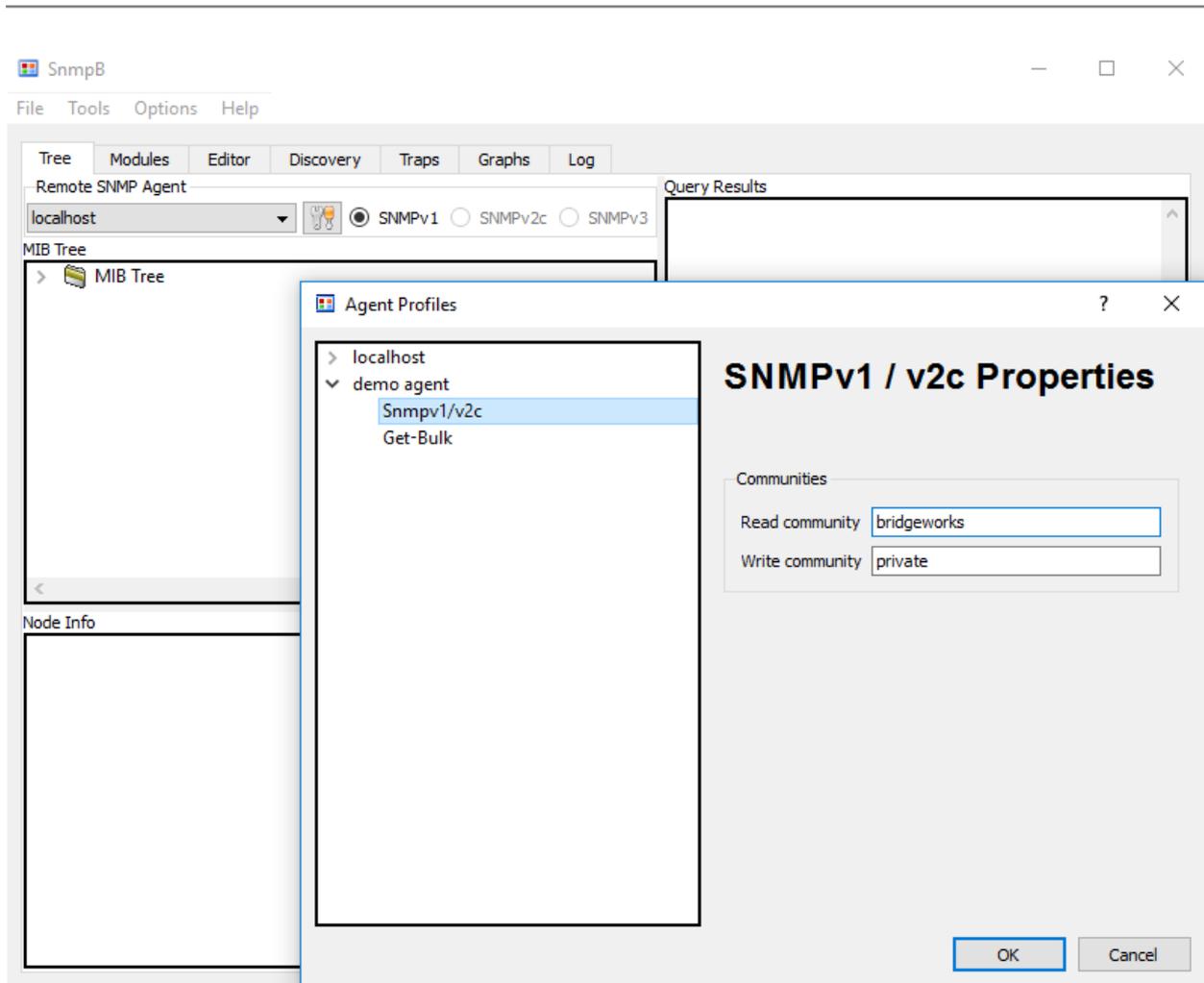
In this example only the *Name*, *Agent Address/Name*, and *Supported SNMP Version* entries needed to be set. All other values were left as default.



Next left click the arrow to the left of the new profile you are setting up. This will expose more detailed settings.

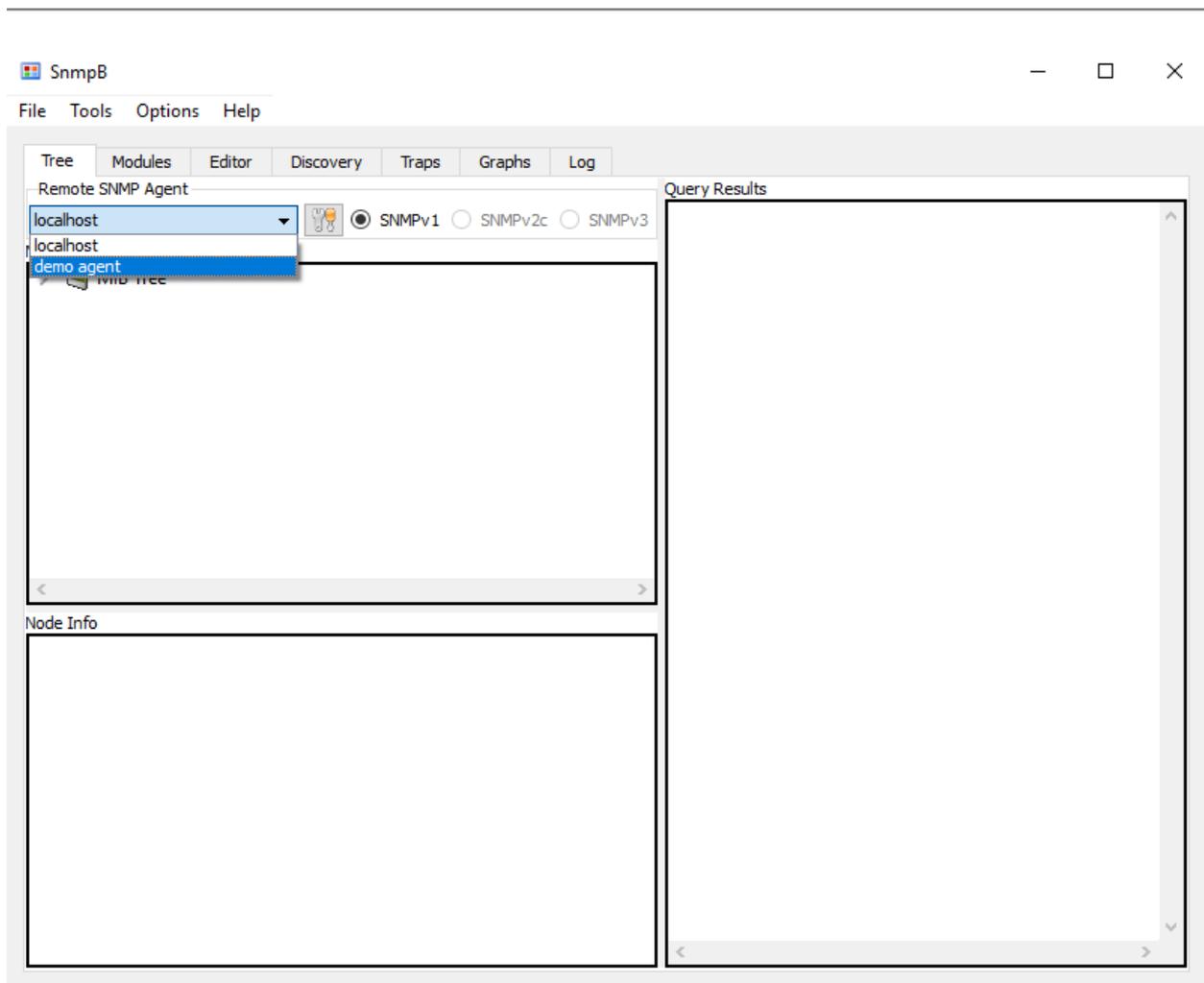
In this instance we only need to change the *Read community* entry to match the *Community Name* we set on the *SNMP* page.

Left click on *SNMPv1/v2c*, then edit the *Read community* entry on the right to match the string entered into the *SNMP* page.



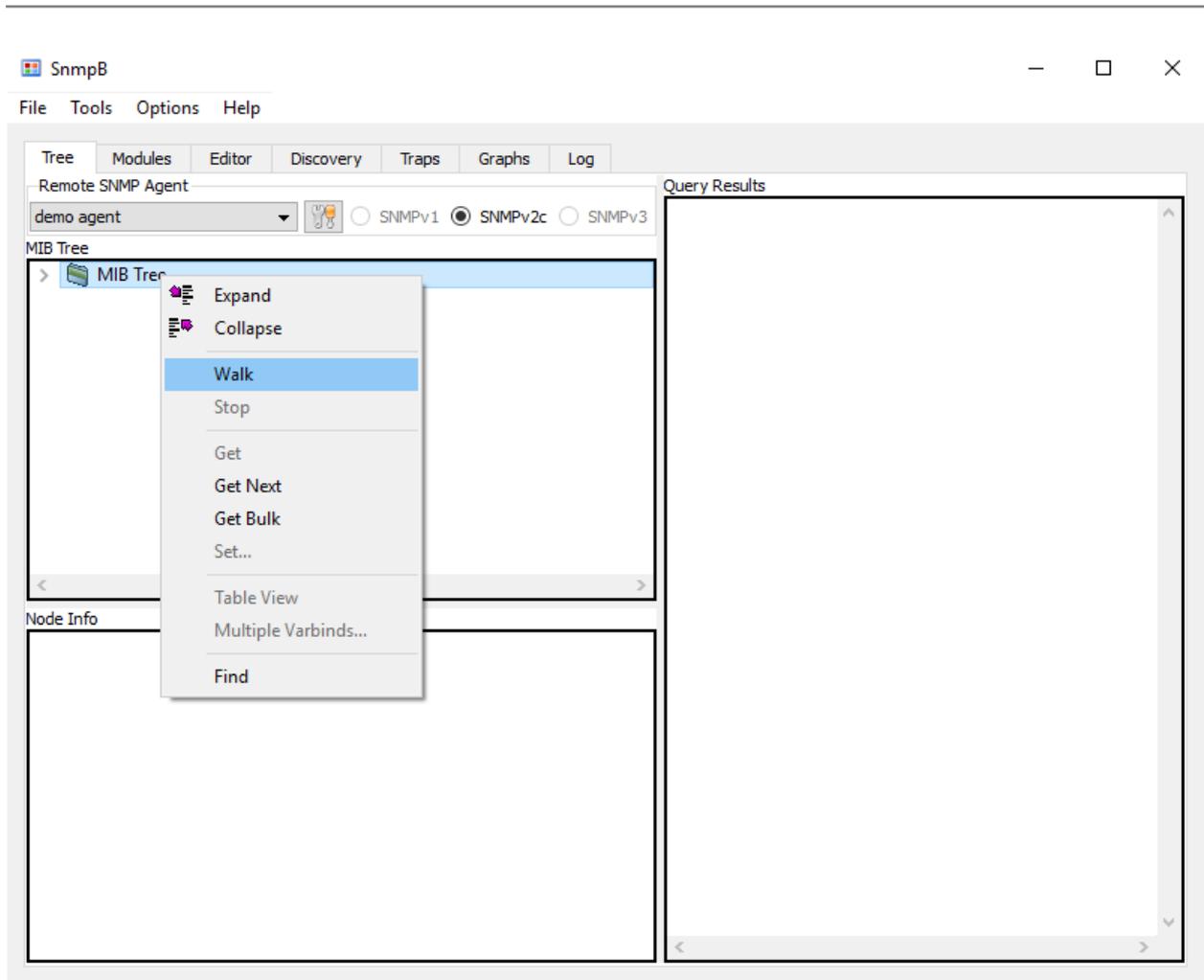
Note: The advanced settings entries change depending on the *Supported SNMP Version* selection; be sure to check the correct versions for your use *before* moving to this section.

Now left click *OK* to add this profile. At this stage you can left click the dropdown in *Remote SNMP Agent* and select your newly created profile.

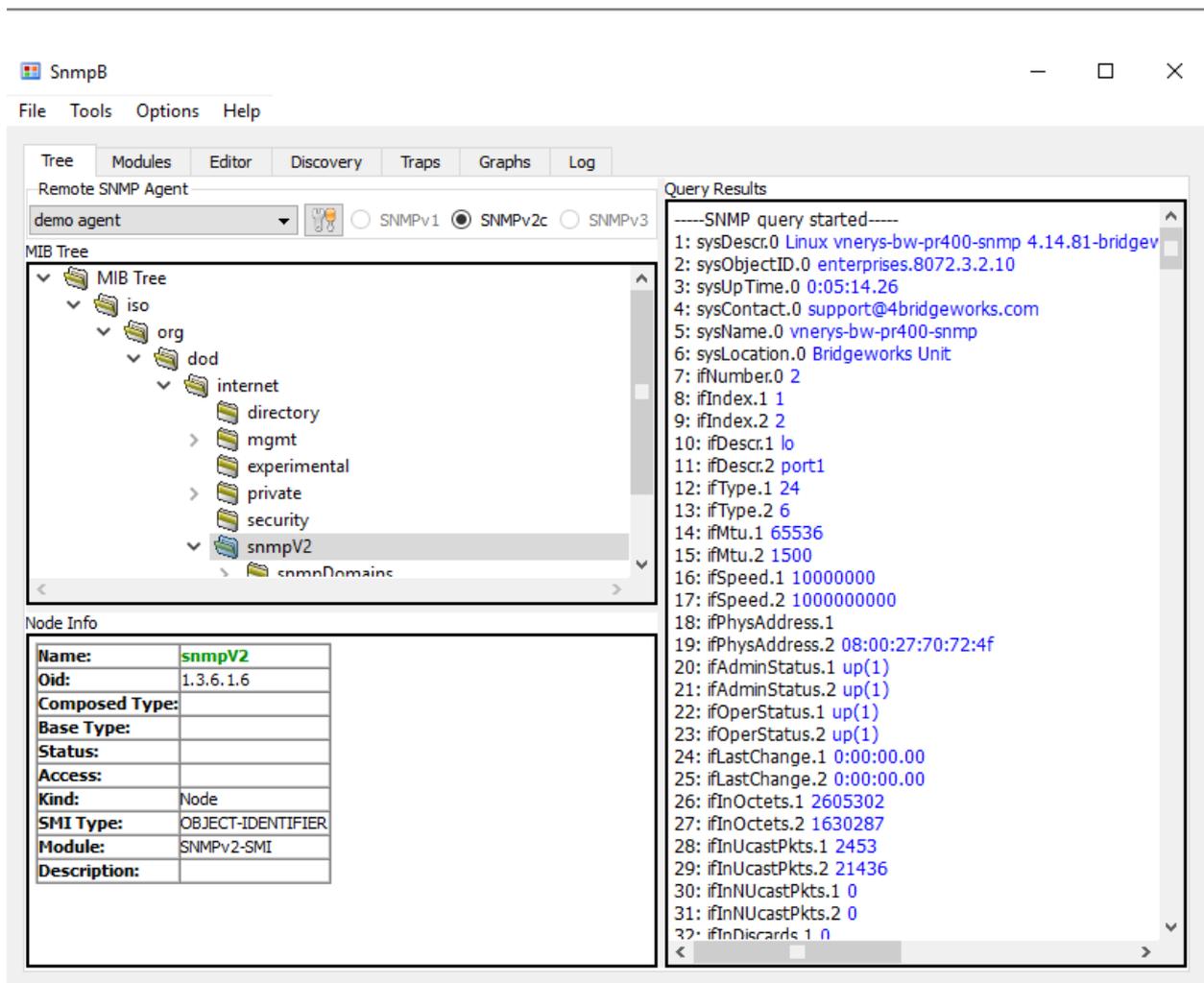


Now right click on the *MIB Tree* top level selection in the *MIB Tree* section.

From the context menu left click on *Walk* to perform a complete scan of all SNMP values.



The scan can take a while to complete depending on connection and number of SNMP entries. The viewer on the right side of the window will show output in realtime as the response is received.



In SNMPb the *MIB Tree* on the left can be expanded to show the MIB hierarchy of the available MIBs on your system. This tree may not be a complete reference to the information being received on the right side of the window.

SNMPb is packaged with a large number of MIBs that are automatically imported and used to interpret names in place of the OID numbers for each entry. If SNMPb cannot find a match for the number it won't substitute a name into the viewer.

Additional MIBs can be loaded into SNMPb using the *Modules* tab at the top of the window. If the needed MIB is not available in SNMPb then the user will need to seek additional MIB files from the manufacturer of the product running the SNMP agent.

---

# 5 SNMPv3

In this example SNMPv3 will be configured in stages:

- 1: NoAuthNoPriv** Replicates an unprotected SNMPv2c connection. There will be no strong security; as such this mode should not be used where sensitive data or vulnerable systems are involved.
- 2: AuthNoPriv** Adds authentication to the SNMP messages. This is to ensure the message content has not been tampered with. The SNMP messages are still plain-text and viewable by third parties, but with password protected digests the message cannot be easily manipulated or spoofed.
- 3: AuthPriv** Enables full privacy of the SNMP messages. At this point the messages are both: hashed for authenticity to ensure they have not been manipulated, and encrypted to stop third parties on the network from easily reading any content. This is the recommended mode for SNMP communication where possible.

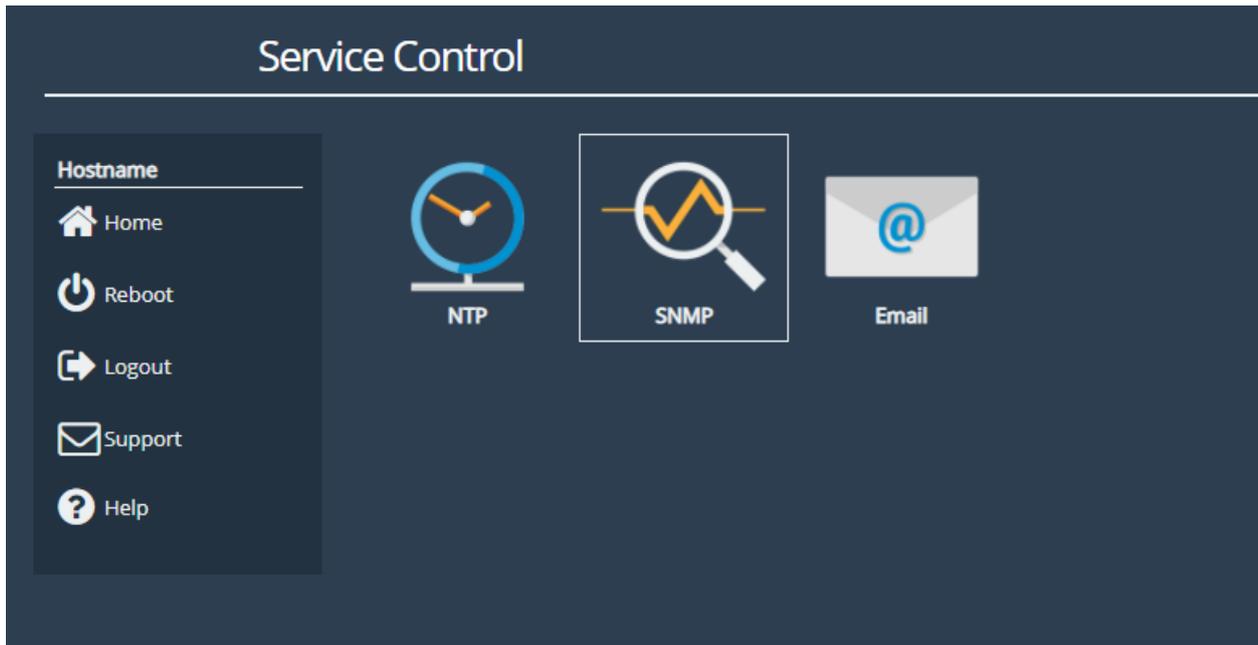
## 5.1 Enabling SNMP Agent: noAuthNoPriv mode

Log in to the target Bridgeworks unit.

Left click on the *Service Control* icon.



Then left click on the *SNMP* icon.



From this page check the box left of the *SNMP v3 Agent* title.

This is a configuration form for the 'SNMP v3 Agent'. At the top left, there is an unchecked checkbox next to the title 'SNMP v3 Agent'. Below the title are five input fields: 'Username' (empty), 'Auth Type' (dropdown menu showing 'SHA1'), 'Auth Password' (empty), 'Privacy Type' (dropdown menu showing 'AES-128'), and 'Privacy Password' (empty). A 'Save' button is located at the bottom right of the form.

Provide a *Username*, set *Auth Type* to *None* and left click on the *Save* icon at the bottom right of the *SNMP v3 Agent* section.

This is the same configuration form as above, but now the checkbox for 'SNMP v3 Agent' is checked. The 'Username' field contains the text 'bridgeworks'. The 'Auth Type' dropdown menu is now set to 'None'. The 'Privacy Type' dropdown menu is also set to 'None'. The 'Save' button remains at the bottom right.

At this stage the SNMP Agent is now running. It is now accessible without any security restrictions beyond a plain-text username.

---

## 5.1.1 Querying the Agent

### 5.1.1.1 Net-SNMP CLI

To receive SNMP information from the SNMP Agent Net-SNMP users can simply run `snmpwalk` to get the entire tree.

Be sure to get the correct username, otherwise, the agent will fail as per the example below.

```
> snmpwalk -v3 -m ALL -u badUsername 10.10.64.24
snmpwalk: Unknown user name
```

An `snmpwalk` will have a lot of response information, this example does not show the full response.

'...' denotes gaps where multiple lines of the SNMP response were cut.

```
> snmpwalk -v3 -m ALL -u bridgeworks 10.10.64.24
...
SNMPv2-MIB::sysContact.0 = STRING: support@4bridgeworks.com
SNMPv2-MIB::sysName.0 = STRING: bw-pr400-snmp
SNMPv2-MIB::sysLocation.0 = STRING: Bridgeworks Unit
IF-MIB::ifNumber.0 = INTEGER: 2
IF-MIB::ifIndex.1 = INTEGER: 1
IF-MIB::ifIndex.2 = INTEGER: 2
IF-MIB::ifDescr.1 = STRING: lo
IF-MIB::ifDescr.2 = STRING: port1
IF-MIB::ifType.1 = INTEGER: softwareLoopback(24)
IF-MIB::ifType.2 = INTEGER: ethernetCsmacd(6)
IF-MIB::ifMtu.1 = INTEGER: 65536
IF-MIB::ifMtu.2 = INTEGER: 1500
IF-MIB::ifSpeed.1 = Gauge32: 10000000
IF-MIB::ifSpeed.2 = Gauge32: 1000000000
...
TCP-MIB::tcpConnectionState.ipv4."10.10.64.24".443.ipv4."10.0.80.136".62933 =
INTEGER: established(5)
...
TCP-MIB::tcpConnectionProcess.ipv4."10.10.64.24".443.ipv4."10.0.80.136".62933 =
Gauge32: 403
...
IF-MIB::ifPromiscuousMode.1 = INTEGER: false(2)
```

In the response above the SNMP manager has interpreted the OID (Object Identifier) values and replaced them with relevant names for each entry. This has been manually enabled with the `-m ALL` option. See specific command options for your `snmpwalk` command.

These names are from MIBs (Management Information Bases), which contain descriptive information about their respective OID value.



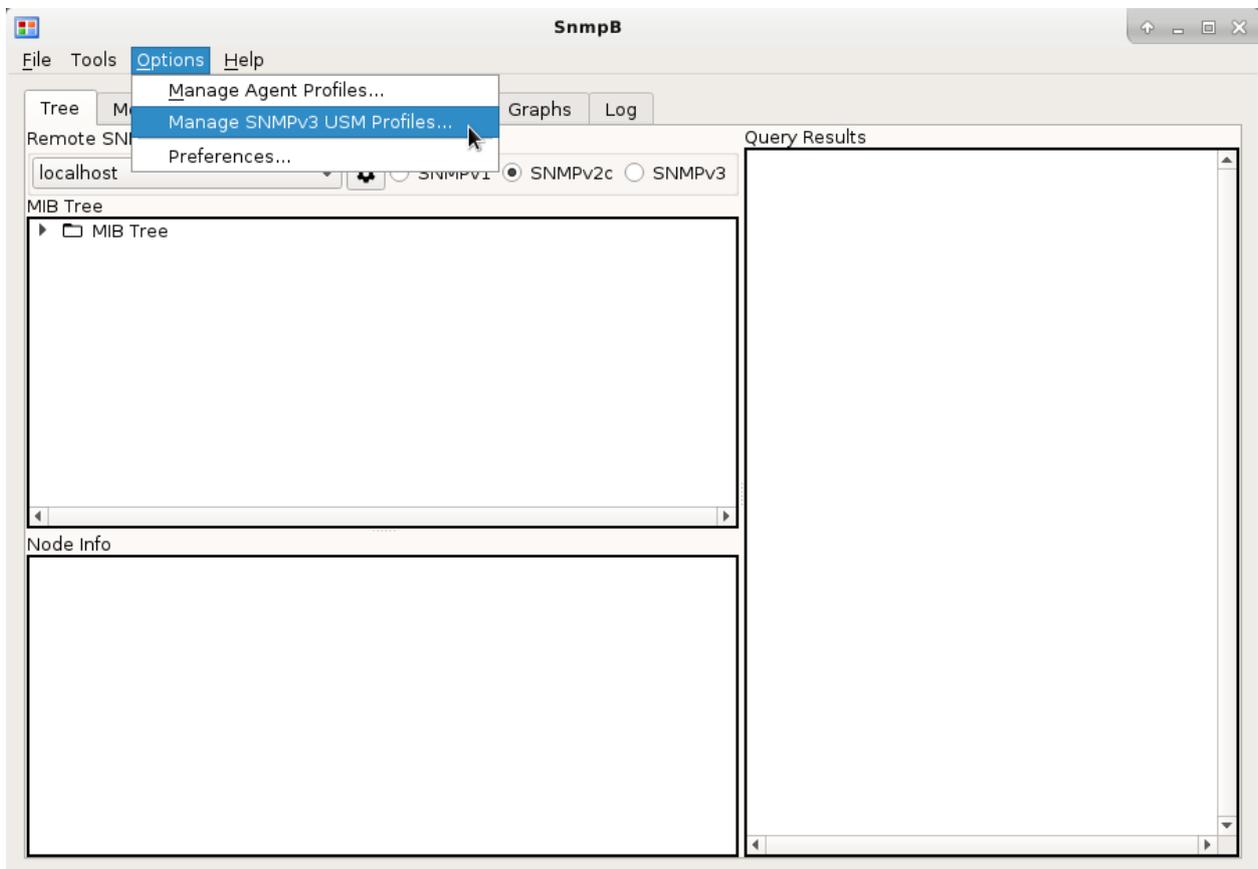
Note: For further use of `snmpwalk` and other Net-SNMP commands such as: `get`, `snmpget`, `snmpgetnext`, and `snmpbulkget`, see the Net-SNMP wiki page (<http://www.net-snmp.org/wiki/>).

### 5.1.1.2 SNMPb GUI

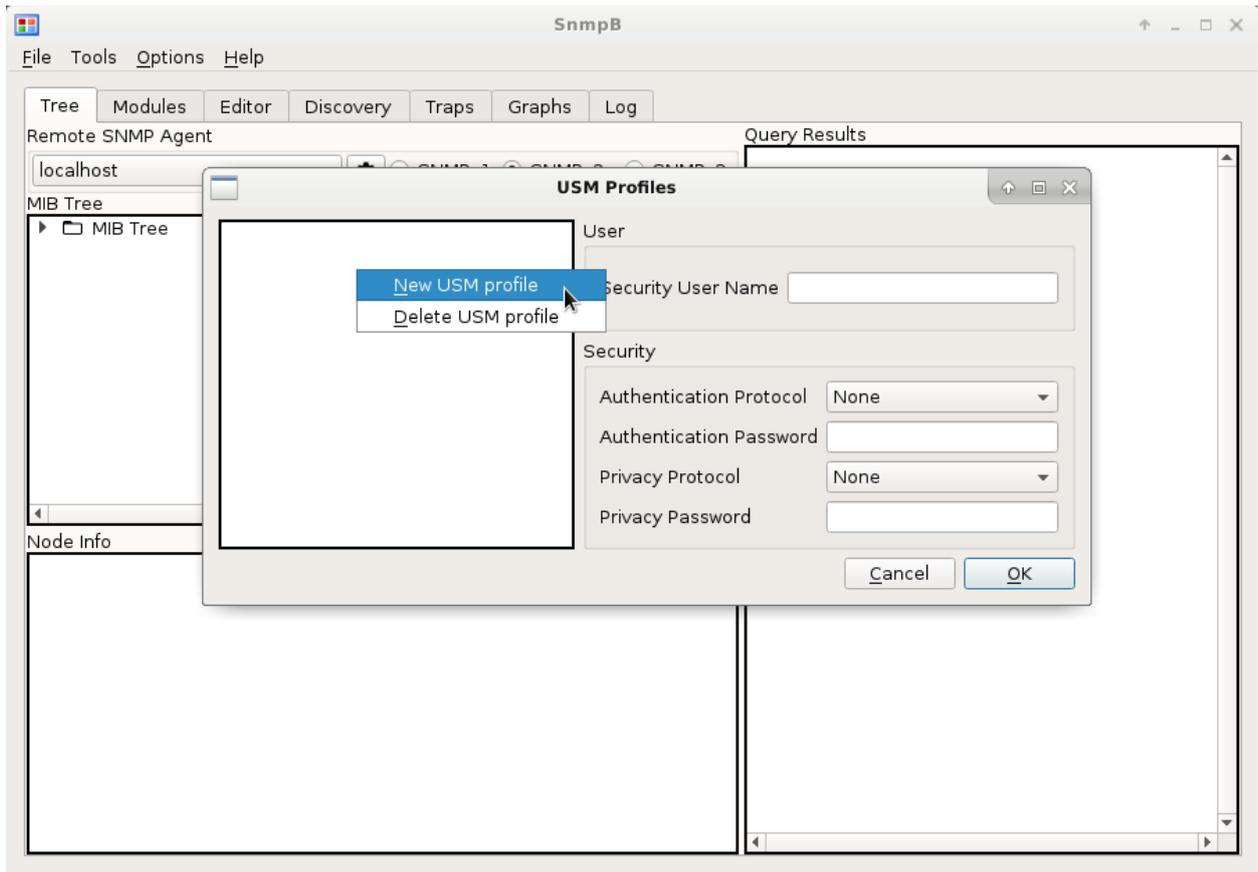
With SNMPb there is more overhead in setting up the initial connection than an equivalent command line query.

Unlike SNMPv2c, SNMPv3 requires a security profile. With SNMPb we need to add an *SNMPv3 USM Profile*.

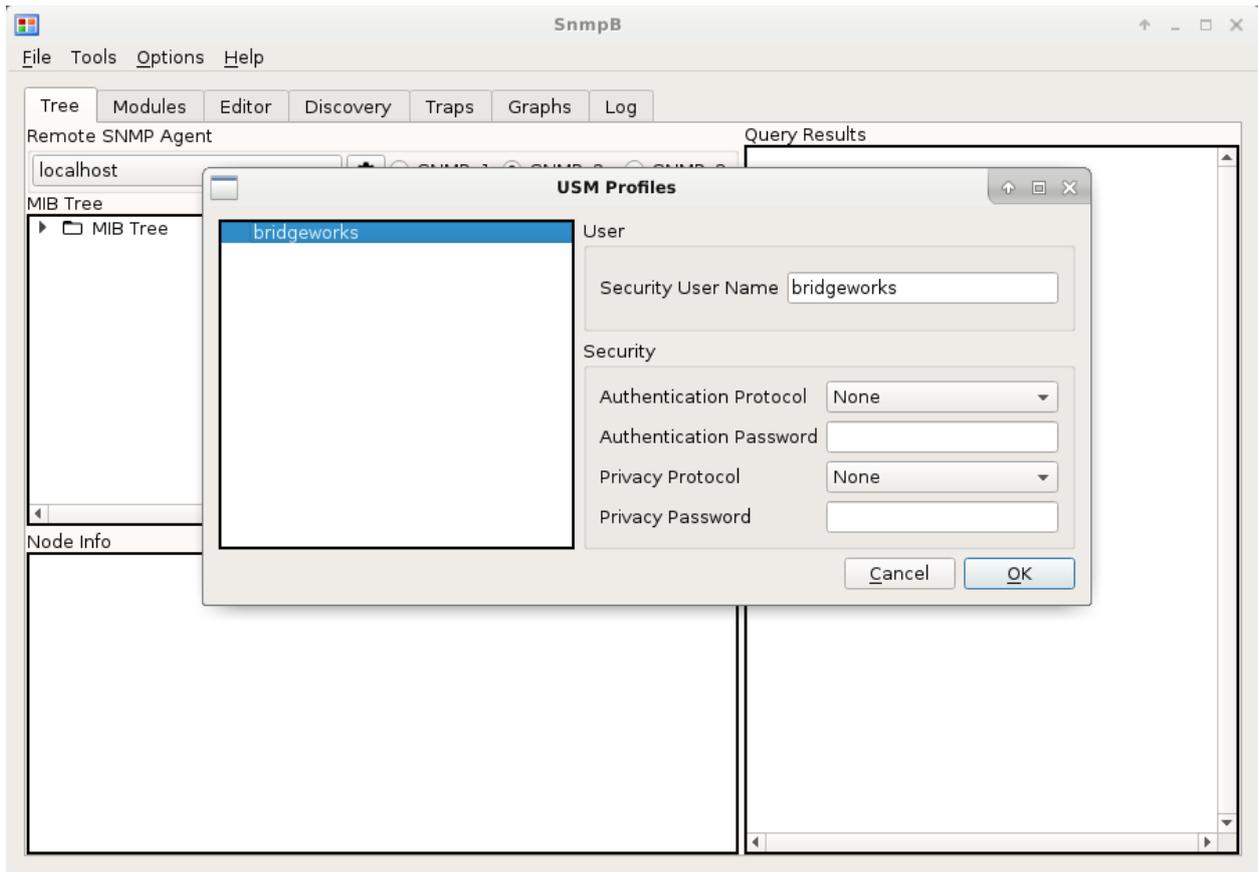
With SNMPb open, left click on *options* and then left click to select *Manage SNMPv3 USM Profiles*.



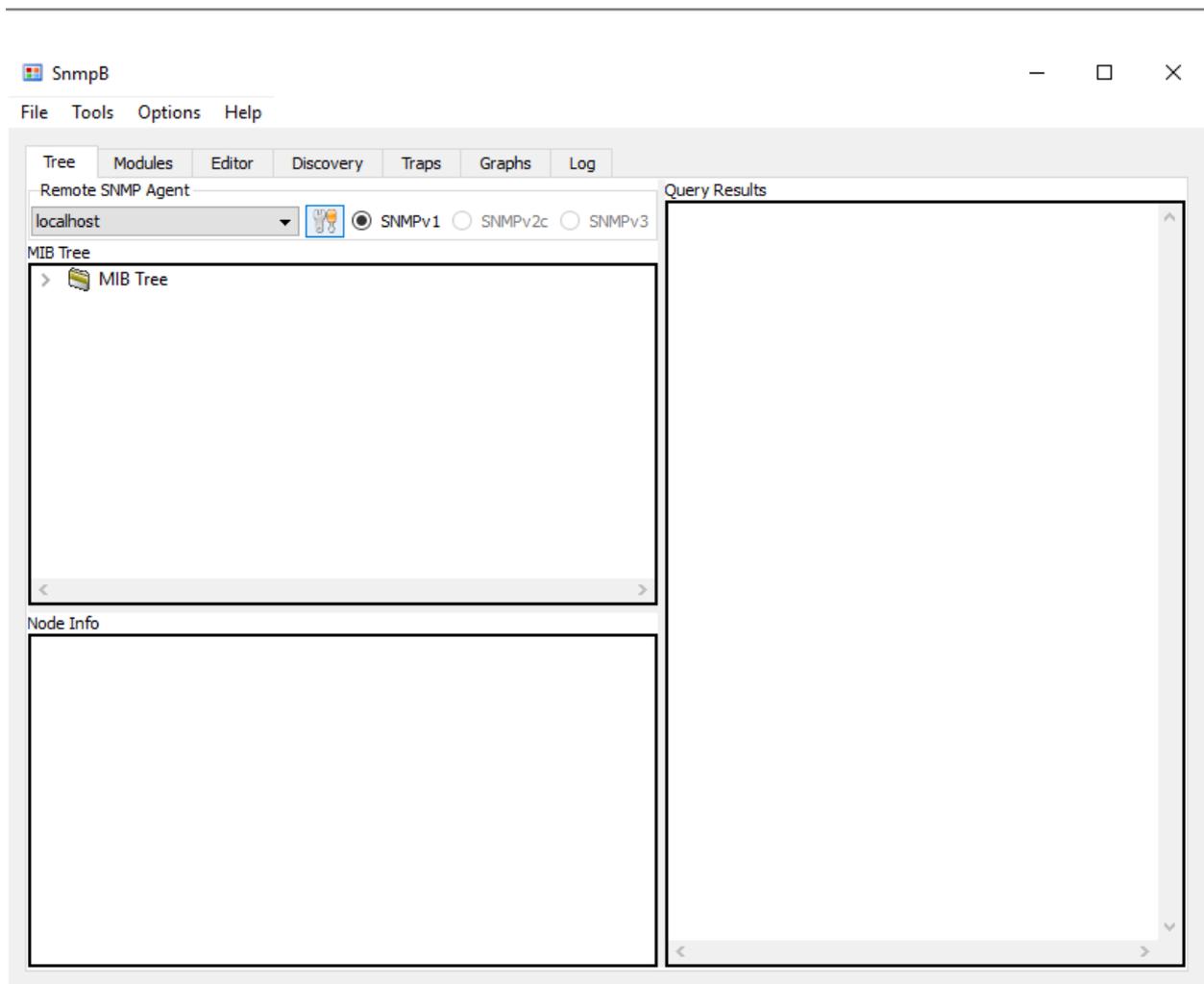
In the new window, right click in the USM profile selection pane on the left. From the context menu, left click on the *New USM profile*.



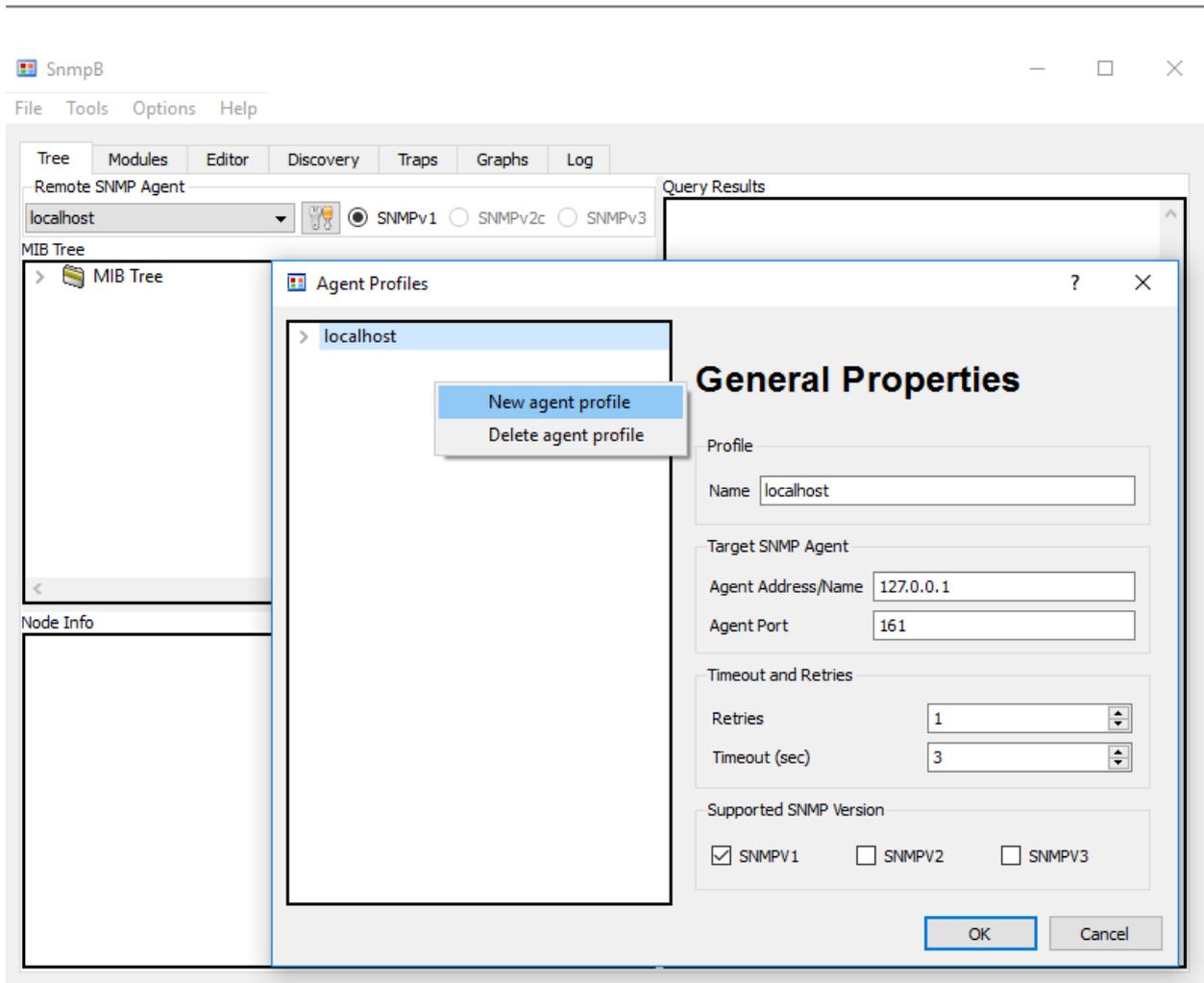
Fill out the username for this profile, it should be the same username you entered into the SNMP page.



Now left click on *OK*. At this stage we have an SNMPv3 profile; now the SNMP Agent can be added. Back on the main SnmpB window, left click the settings icon in the *Remote SNMP Agent* section.

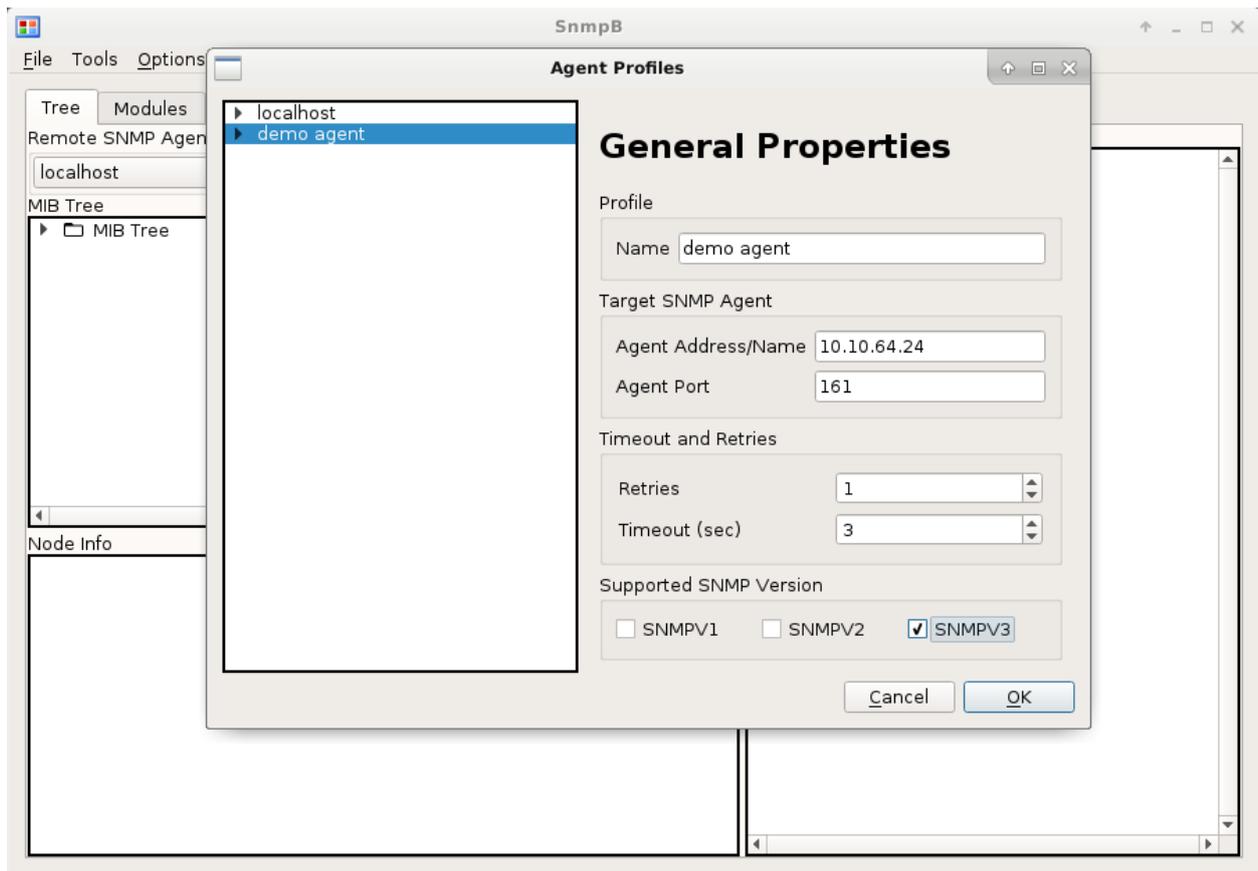


In the new window right click in the profile selection pane on the left. Then left click on *New agent profile*.



Now enter information relevant to your setup. The *Name* entry in this section is for descriptive purposes only.

In this example only the *Name*, *Agent Address/Name*, and *Supported SNMP Version* entries needed to be set. All other values were left as default.

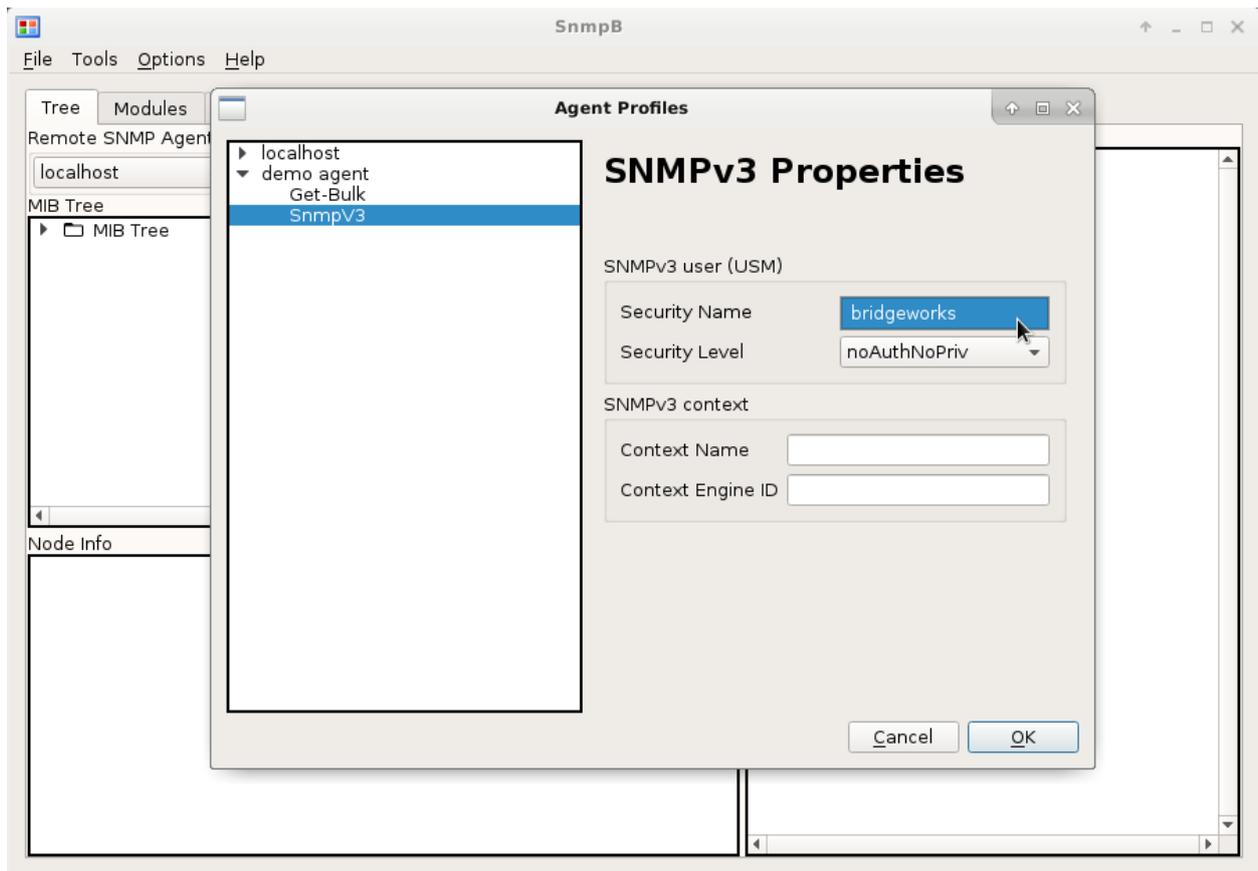


Next left click the arrow to the left of the new profile you are setting up. This will expose more detailed settings.

In this instance we need to attach the SNMPv3 USM Profile to the Agent.

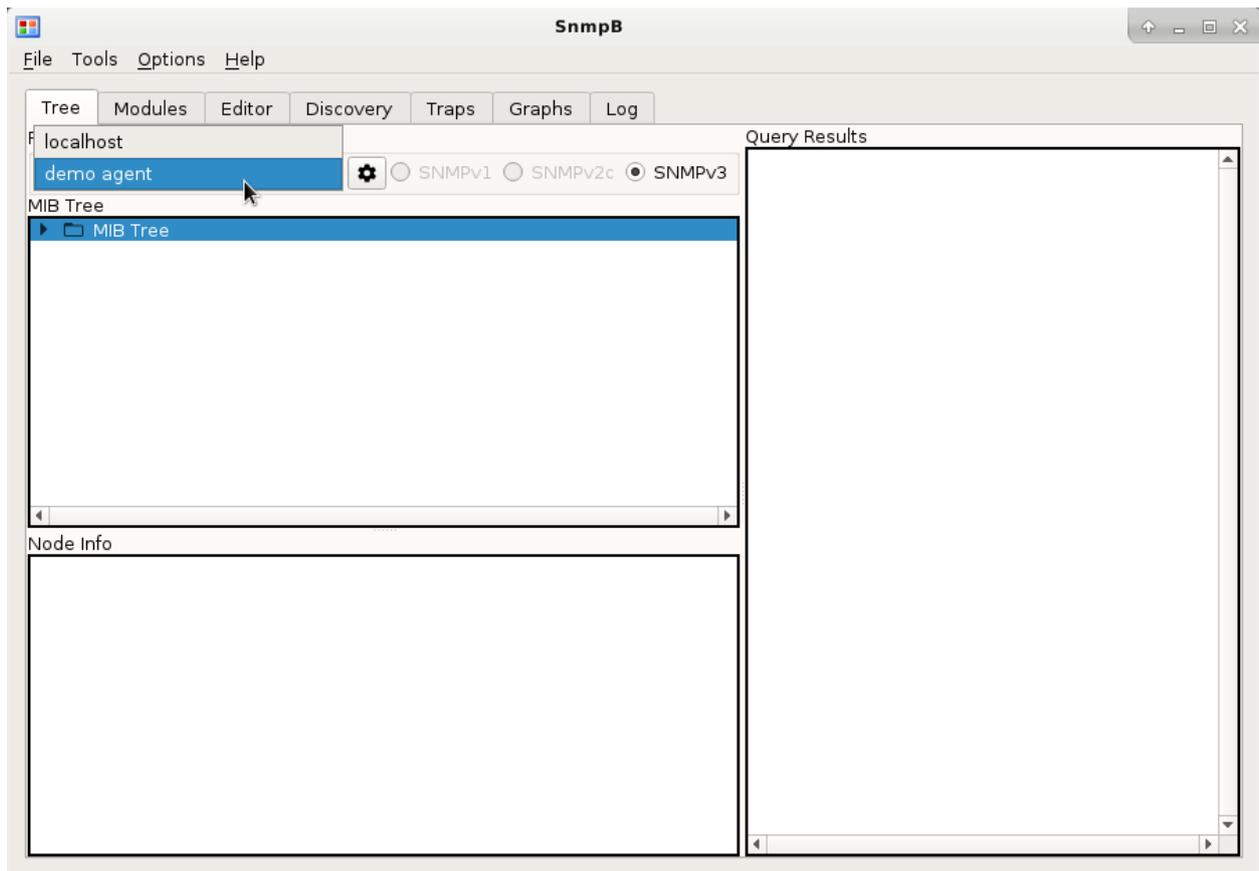
Left click *SNMPv3*.

In the settings displayed on the right of the window, left click on the dropdown right of the *Security Name* label, then from the dropdown left click to select the name of the USM profile you created earlier. The name should match the name used when enabling SNMPv3 on the SNMP page.



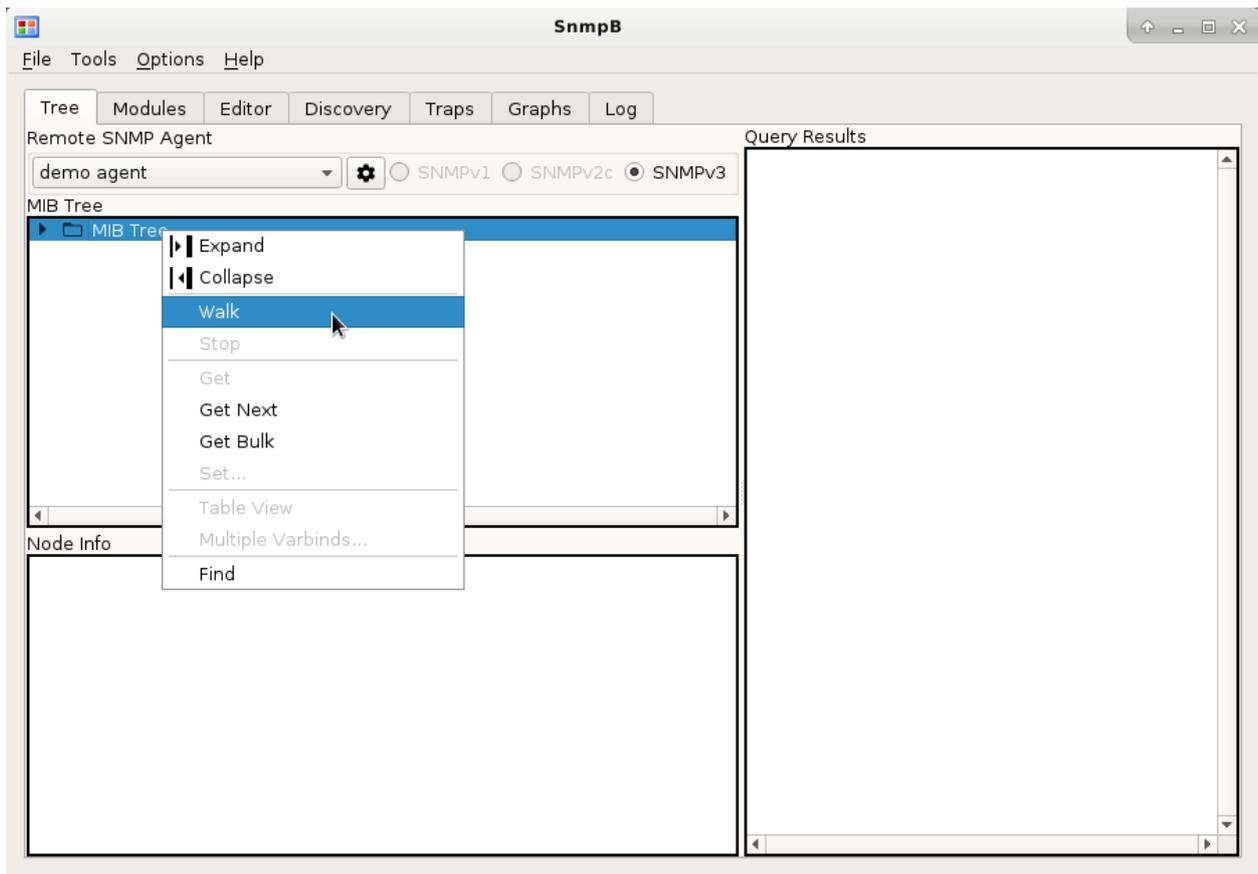
Note: The advanced settings entries change depending on the *Supported SNMP Version* selection; be sure to check the correct versions for your use *before* moving to this section.

Now left click **OK** to add this profile. At this stage you can left click the dropdown in *Remote SNMP Agent* and select your newly created profile.

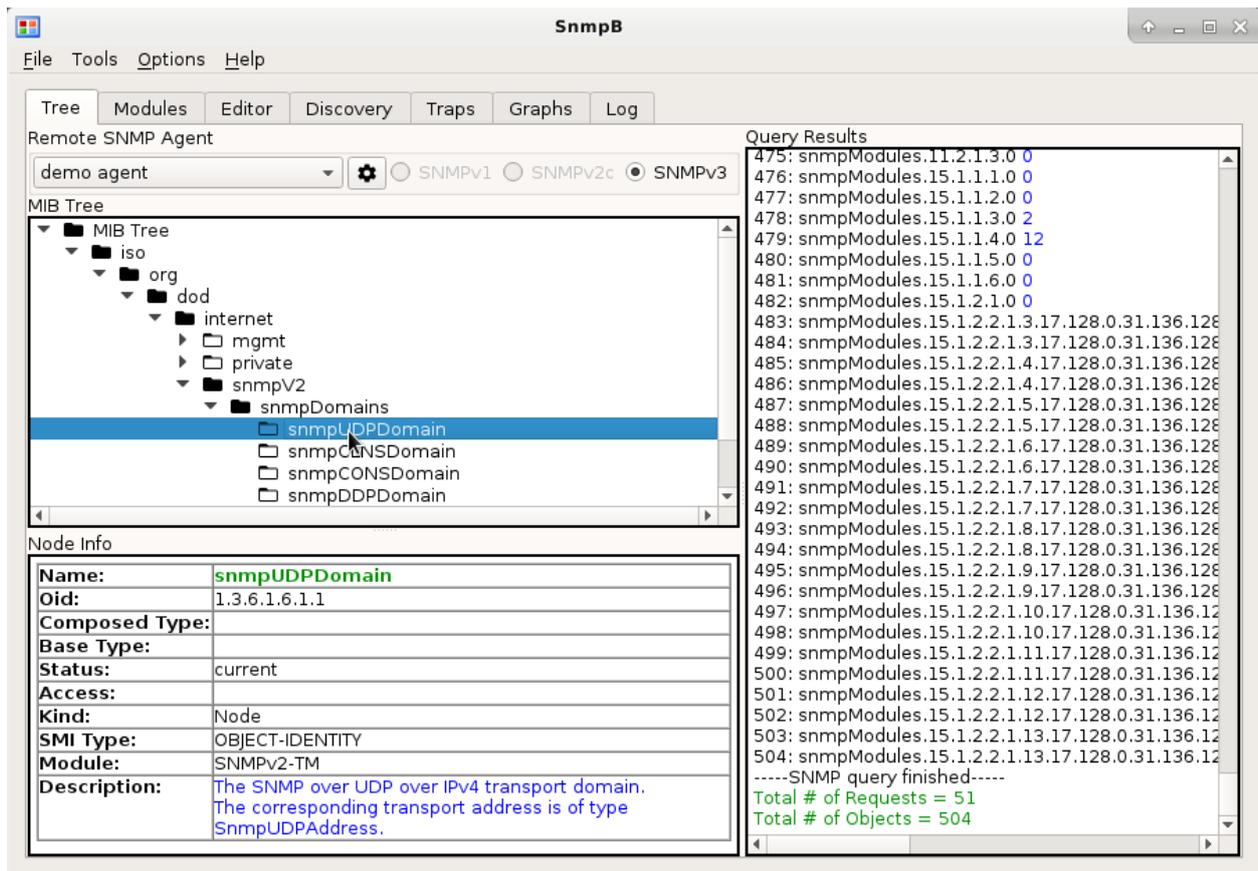


Now right click on the *MIB Tree* top level selection in the *MIB Tree* section.

From the context menu left click on *Walk* to perform a complete scan of all SNMP values.



The scan can take a while to complete depending on connection and number of SNMP entries. The viewer on the right side of the window will show output in realtime as the response is received.



In SNMPb the *MIB Tree* on the left can be expanded to show the MIB hierarchy of the available MIBs on your system. This tree may not be a complete reference to the information being received on the right side of the window.

SNMPb is packaged with a large number of MIBs that are automatically imported and used to interpret names in place of the OID numbers for each entry. If SNMPb cannot find a match for the number it won't substitute a name into the viewer.

Additional MIBs can be loaded into SNMPb using the *Modules* tab at the top of the window. If the needed MIB is not available in SNMPb then the user will need to seek additional MIB files from the manufacturer of the product running the SNMP agent.

## 5.2 Enabling authentication: authNoPriv Mode

With SNMPv3 NoAuthNoPriv running correctly, the next available step to harden the SNMP connection is to add authentication.

Authentication is a way to ensure that received message was sent by a system that has the correct hashing password, and that the message was not tampered with. This security does not hide the information being sent to or from the SNMP agent, and as such does not protect sensitive data from being viewed.

SNMPv3 makes use of message digests to verify the integrity of the message. Additional use of a password in the hashing function used to digest the message also increases the likelihood that the message is from an authorised SNMP entity.

---

The SNMP entities will share the same password for their hashing. When a message and message digest come in to an entity it will digest the message internally and compare the value with the value from the digest sent with the message. If the two digest values match it will consider the message authentic and process it.

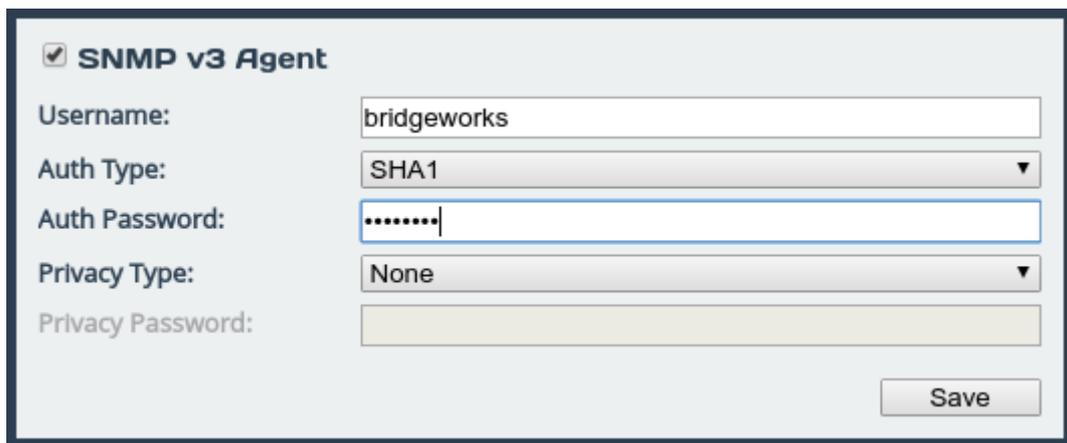
SNMPv3 supports two types of hash based authentication:

- MD5** This hashing function produces a 128 bit hash value from the message input.
- SHA** SHA (SHA1 specifically for SNMPv3) produces a 160 bit hash value from the message input.

In this example the SHA hashing type is used. SHA is chosen over MD5 as SHA is the more secure of the two methods.

In this example the authentication settings are enabled on the SNMPv3 configuration previously set up. (See Section 5.1: [Enabling SNMP Agent: noAuthNoPriv mode](#) for settings.)

On the SNMP page, left click the dropdown for the *Auth Type*, and left click to select *SHA*. Then enter an *Auth Password*. “demoAuth” is used in this guide.



**SNMP v3 Agent**

Username:

Auth Type:

Auth Password:

Privacy Type:

Privacy Password:

## 5.2.1 Querying the Agent

### 5.2.1.1 Net-SNMP CLI

The Net-SNMP command previously used now needs to be modified to include the *Auth Type* and present the correct *Auth Password*.

If an incorrect authentication is given SNMP will specify the failure.

Incorrect:

```
> snmpwalk -v3 -u bridgeworks -m ALL -a SHA -A 2Short -l authNoPriv 10.10.64.24
Error: passphrase chosen is below the length requirements of the USM (min=8).
snmpwalk: (The supplied password length is too short.)
Error generating a key (Ku) from the supplied authentication pass phrase.
```

```
> snmpwalk -v3 -u bridgeworks -m ALL -a SHA -A aBadAuth -l authNoPriv 10.10.64.24
```

---

Error in packet.  
Reason: authorizationError (access denied to that object)

```
> snmpwalk -v3 -u bridgeworks -m ALL -a SHA -A demoAuth 10.10.64.24  
Error in packet.  
Reason: authorizationError (access denied to that object)
```

Correct:

```
> snmpwalk -v3 -u bridgeworks -m ALL -a SHA -A demoAuth -l authNoPriv 10.10.64.24  
...
```

The final line in the example above is the only correct entry due to the correct hash config and the addition of `-l authNoPriv` to specify that no privacy has been set.

The Net-SNMP commands should now work; showing identical output to the SNMPv3 noAuthNoPriv section previously in this guide. (See Section [5.1.1.1: Net-SNMP CLI](#).)

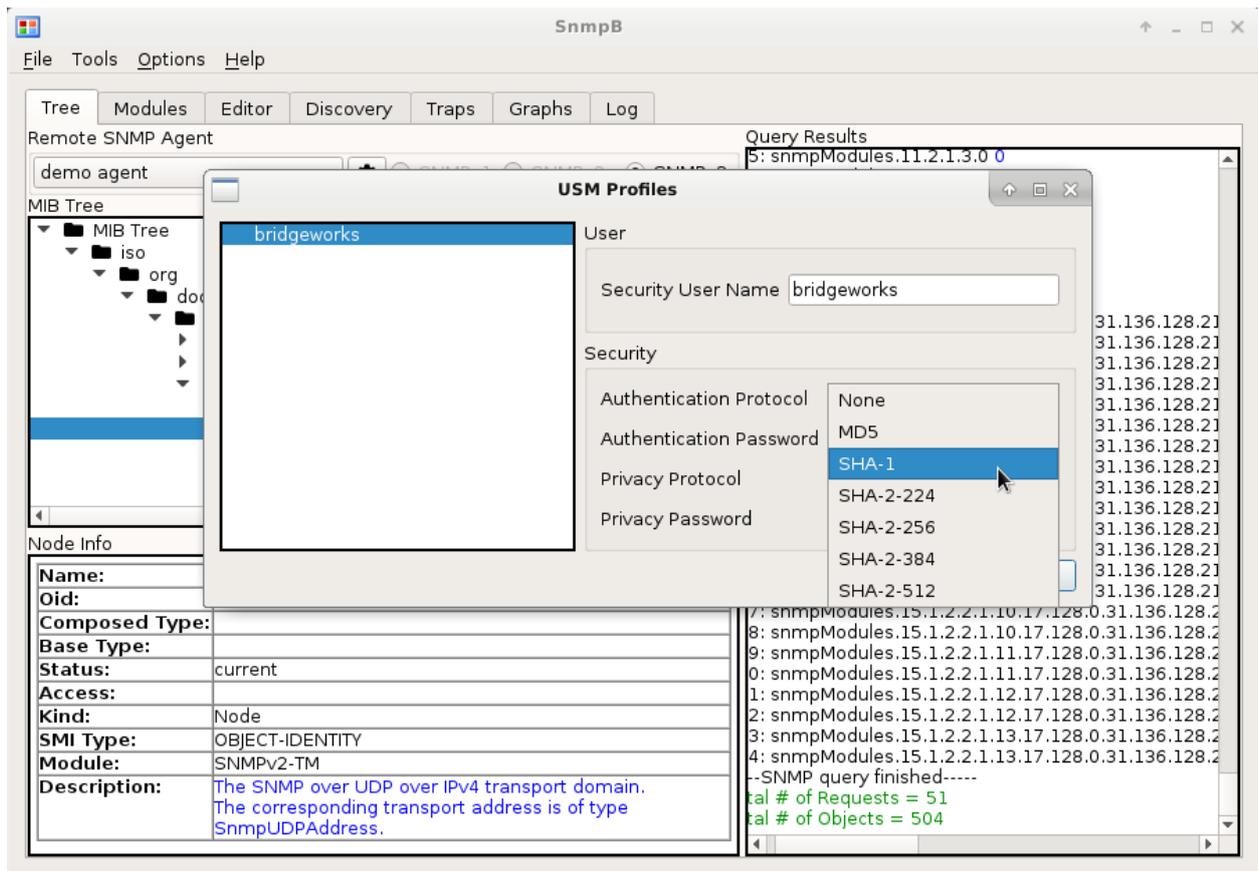
### 5.2.1.2 SNMPb GUI

To query the SNMP agent entities the SNMPb software will now need to be reconfigured to add the authentication password. Both the USM profile and the SNMP agent profile added previously need to be changed.

(See Section [5.1.1.2: SNMPb GUI](#) for accessing the USM profile and Agent profile management.)

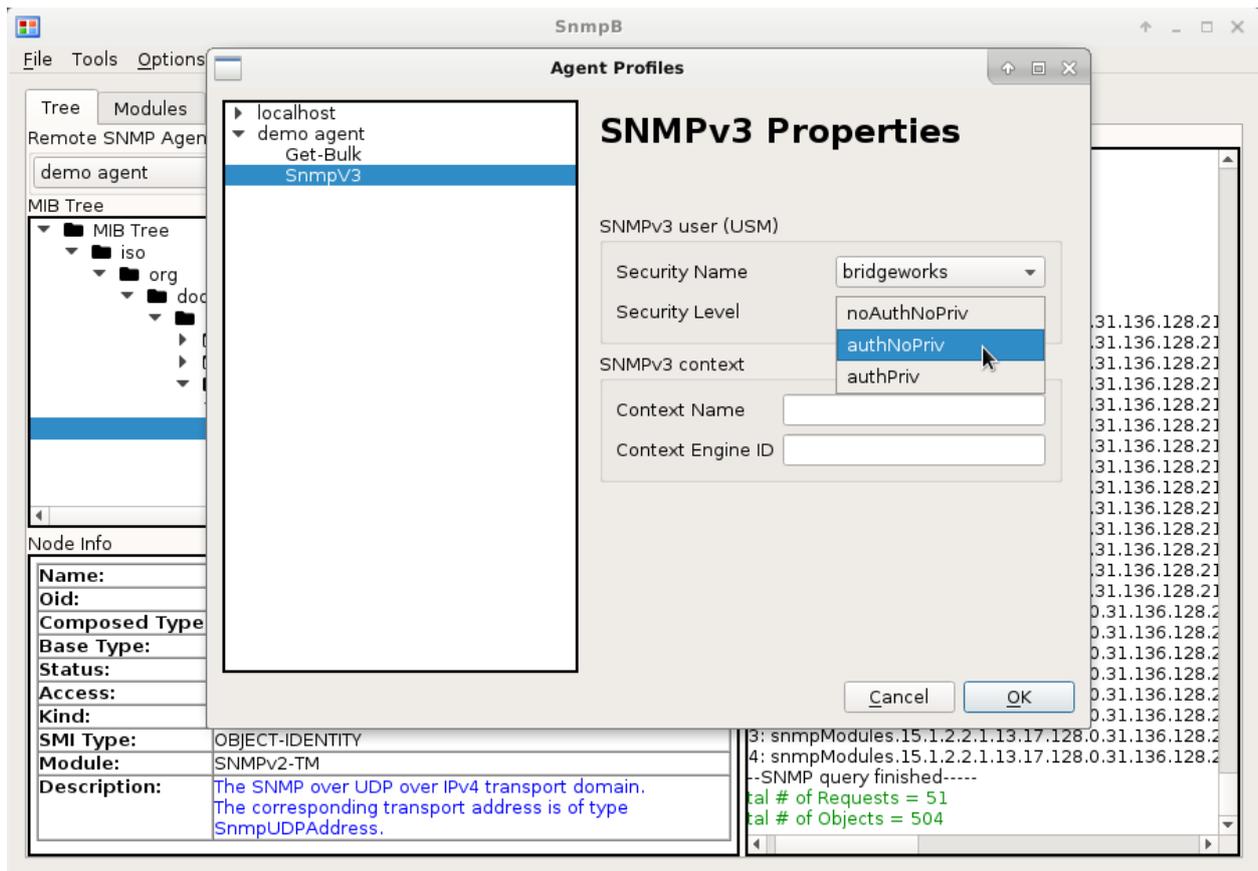
In the *USM Profiles* section, accessed through *Options* then *Manage SNMPv3 USM Profiles*, left click to select the intended profile. Once the profile has been selected, change the dropdown for *Authentication Protocol* to *SHA-1*.

Then add the authentication password to the *Authentication Password* entry. Left click *OK* to apply the settings.



In the *Agent Profiles* section, accessed through *Options* then *Manage Agent Profiles*, left click the arrow left of the intended profile to expand the advanced settings. From the advanced settings, left click *SNMPv3*.

The *SNMPv3 Properties* view on the right should now appear. In this view, left click the dropdown entry for *Security Level*, then left click to select *authNoPriv*. Then left click *OK* to save the settings.



An SNMPb walk command should now work; showing identical output to the SNMPv3 noAuthNoPriv section previously in this guide. (See Section 5.1.1.2: [SNMPb GUI](#).)

### 5.3 Enabling privacy: authPriv Mode

At this point the SNMPv3 example laid out in this guide now has SHA authentication enabled. SNMP messages between entities are now more trusted. The final step available in SNMPv3 is to enable privacy.

Privacy in this context means to make the SNMP messages hard to read by anything that does not possess the privacy key. Where authentication aims to make sure SNMP entities don't get malicious requests, privacy aims to hide the information sent between SNMP entities so sensitive information is not easily read maliciously.

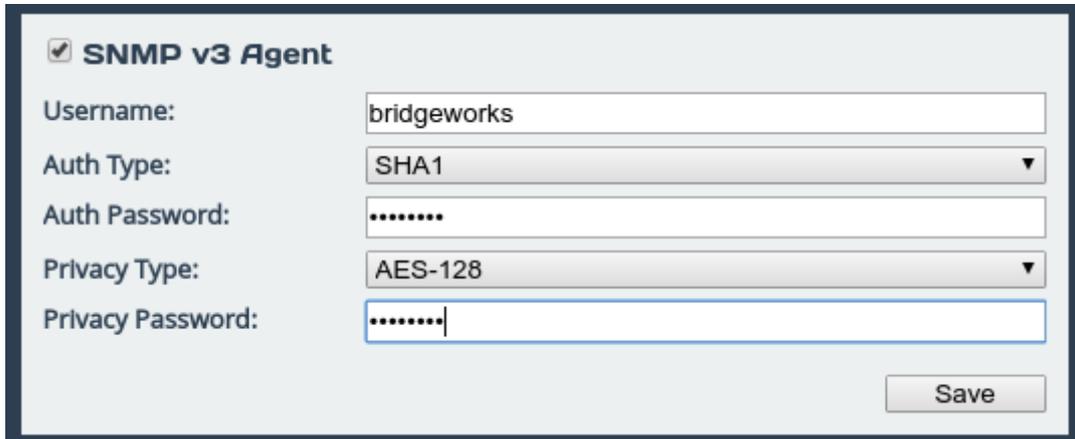
SNMPv3 offers two block ciphers for privacy:

- DES** The *Data Encryption Standard*. This cipher is no longer an officially recognised standard. Brute force attacks and rainbow table cracking have proven to break DES very quickly.
- AES** The *Advanced Encryption Standard* is an officially recognised standard endorsed by NIST (the National Institute of Standards and Technology).

In this example the AES cipher is used. AES is significantly more secure than DES and should be preferred where possible.

In this example the privacy settings are enabled on the SNMPv3 configuration previously set up. (See Section 5.2: [Enabling authentication: authNoPriv Mode.](#))

In the SNMP page, left click the dropdown for the *Privacy Type*, and left click to select *SHA*. Then enter a *Privacy Password*. “demoPriv” is used in this guide.



### 5.3.1 Querying the Agent

#### 5.3.1.1 Net-SNMP CLI

The Net-SNMP command previously used for the authNoPriv needs to now include the correct *Privacy type* and specify the *Privacy Password*.

Incorrect:

```
> snmpwalk -v3 -u bridgeworks -l authPriv -x AES -X demoPri -a SHA -A demoAuth
10.10.64.24
```

```
Error: passphrase chosen is below the length requirements of the USM (min=8).
snmpwalk: (The supplied password length is too short.)
Error generating a key (Ku) from the supplied privacy pass phrase.
```

```
> snmpwalk -v3 -u bridgeworks -l authPriv -x AES -X demoPrivV -a SHA -A demoAuth
10.10.64.24
```

Timeout: No Response from 10.10.64.24

Correct:

```
> snmpwalk -v3 -u bridgeworks -m ALL -a SHA -A demoAuth -x AES -X demoPriv -l
authPriv 10.10.64.24
...
```

The Net-SNMP commands should now work; showing identical output to the SNMPv3 noAuthNoPriv section previously in this guide. (See Section 5.1.1.1: [Net-SNMP CLI.](#))

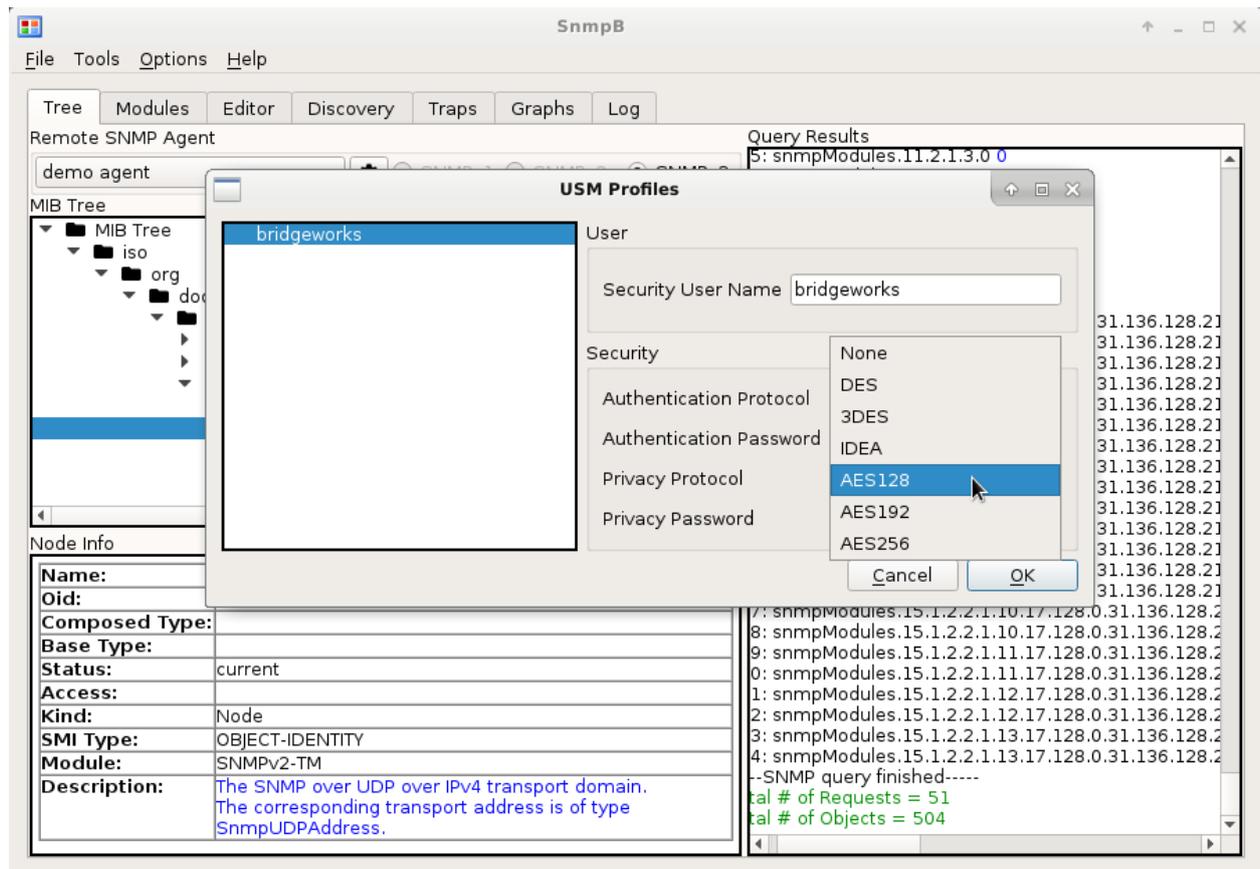
### 5.3.1.2 SNMPb GUI

To query the SNMP agent entities the SNMPb software will now need to be reconfigured to add the privacy type and password. Both the USM profile and the SNMP agent profile added previously need to be changed.

(See Section 5.1.1.2: [SNMPb GUI](#) for accessing the USM profile and Agent profile management.)

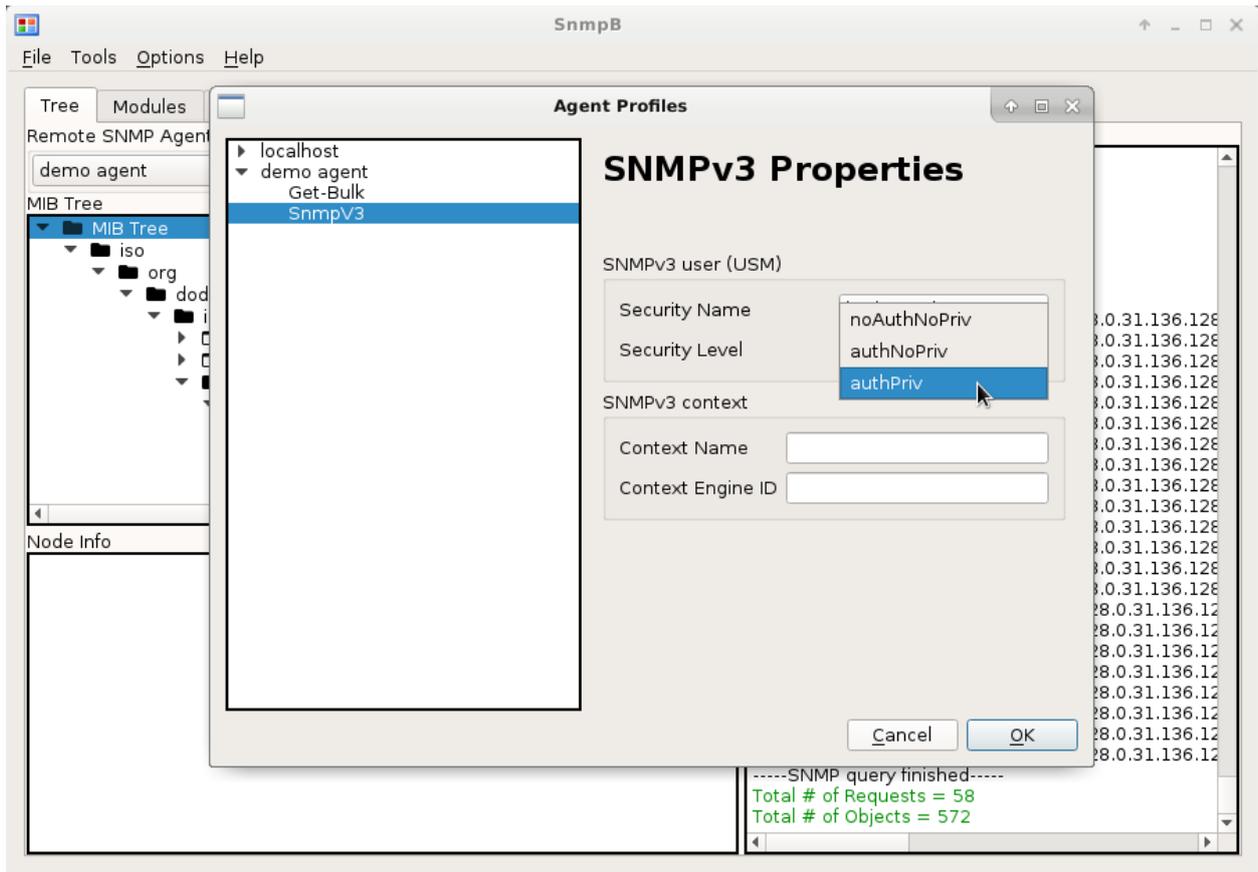
In the *USM Profiles* section, accessed through *Options* then *Manage SNMPv3 USM Profiles*, left click to select the intended profile. Once the profile has been selected, change the dropdown entry for *Privacy Protocol* to *AES-128*.

Then add the privacy password to the *Privacy Password* entry. Left click *OK* to apply the settings.



In the *Agent Profiles* section, accessed through *Options* then *Manage Agent Profiles*, left click the arrow left of the intended profile to expand the advanced settings. From the advanced settings, left click *SNMPv3*.

The *SNMPv3 Properties* view on the right should now appear. In this view, left click the dropdown entry for *Security Level*, then left click to select *authPriv*. Then left click *OK* to save the settings.



An SNMPb walk command should now work; showing identical output to the SNMPv3 noAuthNoPriv section previously in this guide. (See Section 5.1.1.2: [SNMPb GUI](#).)

---

# 6 SNMP Traps

## 6.1 Types of Traps

This guide shows an SNMPv3 trap sink configuration. While an SNMPv3 setup will not work with older versions of SNMP, this guide can be followed when deploying an older version SNMP trap sink. When deploying to an older version of SNMP some elements that are referenced in this guide will not be available for configuring.

Below is a quick summary of different trap versions and types:

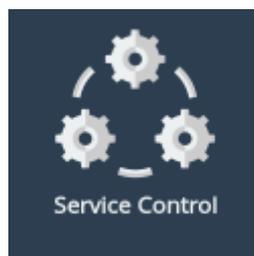
- **v1 - Trap** - The most basic trap. This trap is an unsolicited message sent by an Agent to a Master. Trap messages can be self contained messages, or contain other SNMP objects to add more info.
- **v2 - Trap** - v2 traps are functionally the same as v1.
- **v2 - Inform** - v2 adds the *inform* type, this is a trap that requires the receiving master entity to confirm the receipt of the message.
- **v3 - Trap** - v3 traps gain the same authentication and privacy options available to SNMPv3. It also replaces the community string with the *Engine ID*, the intent is to add more description and context to the access control system.
- **v3 - Inform** - v3 informs gain the same benefits of the v3 traps.

## 6.2 Adding a trap sink

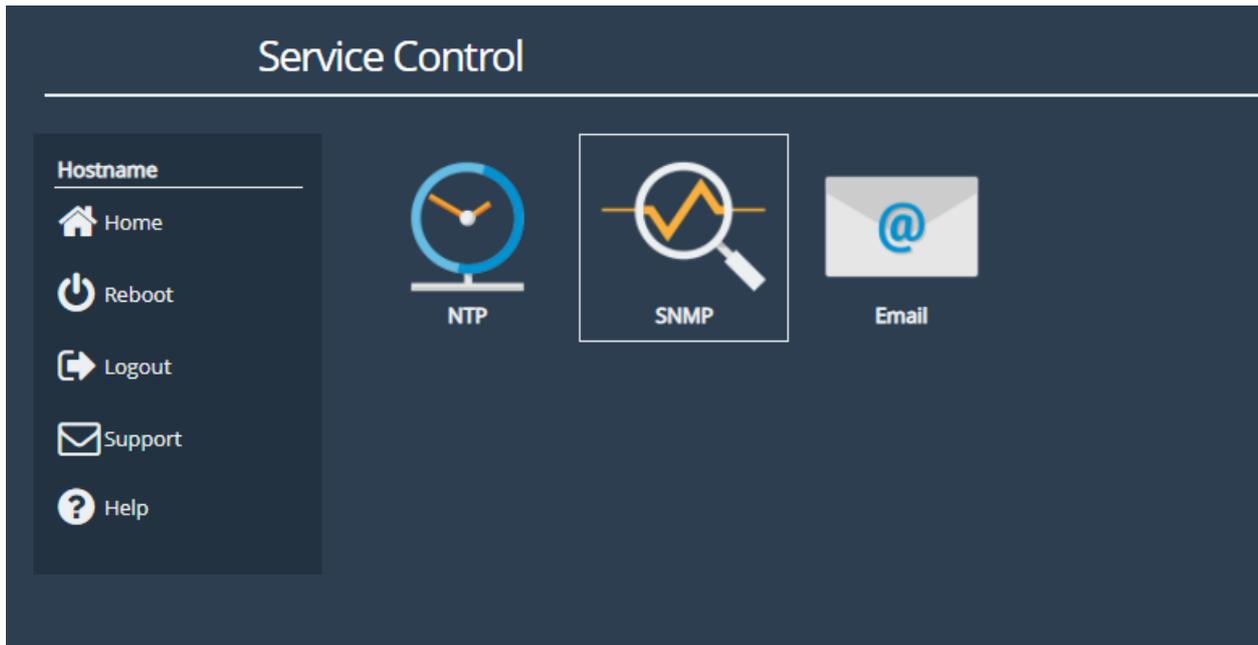
In this example the unit has been set up to run as an SNMPv3 authPriv agent entity.

Log in to the target Bridgeworks unit.

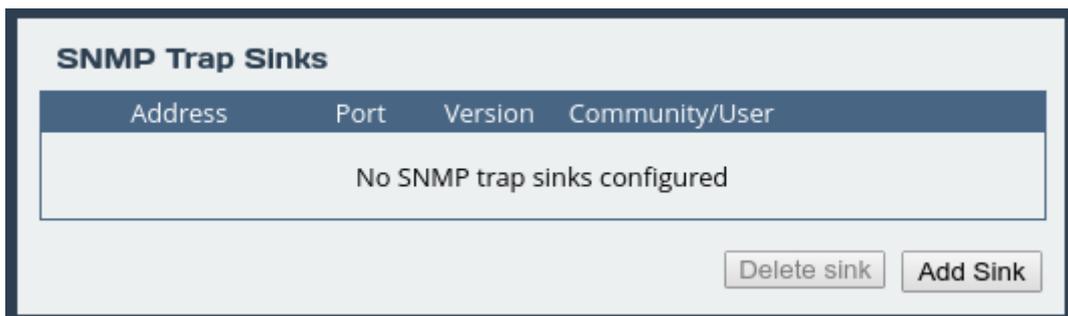
Left click on the *Service Control* icon.



Then left click on the *SNMP* icon.



The *SNMP Trap Sinks* section of the page shows the trap sinks that are currently known of by the system.



Left click on *Add Sink* to bring up the *Add SNMP Trap Sink* form. By default this form will be displaying the entries for a *v3 Inform* style event. Be sure to change the dropdown options for *Version* before adding specific settings, the form will present different options depending on the version selected.

### Add SNMP Trap Sink

Address:

Port:

Version:

Type:

Username:

Engine ID:

Auth Type:

Auth Password:

Privacy Type:

Privacy Password:

Fill out the fields you need, this will depend on the *SNMP mode* you are using. In this example the *authPriv* mode is being used, so both *Authentication* and *Privacy* are set up.

In this example the same user from the SNMP setup chapters is being used, so the Authentication and Privacy passwords are also reused. (See Section 5.2: [Enabling authentication: authNoPriv Mode](#) and Section 5.3: [Enabling privacy: authPriv Mode](#) for enabling these features.)



**Important:** If the trap sink uses both authentication and privacy but only a single password, it is likely that the single password is being used for both authentication and privacy. In this case the same password must be entered into both fields.

**Add SNMP Trap Sink**

Address: 10.10.72.5

Port: 162

Version: v3

Type: Inform

Username: bridgeworks

Engine ID:

Auth Type: SHA1

Auth Password: .....

Privacy Type: AES-128

Privacy Password: .....

Cancel Add Sink

Left click *Add Sink* to save the trap configuration. Confirm that the new addition is present in the *SNMP Trap Sinks* section.

**SNMP Trap Sinks**

Address	Port	Version	Community/User	
10.10.72.5	162	v3	bridgeworks	<a href="#">More Info</a>

Delete sink Add Sink

At this stage the system will send traps to the destination address previously specified. An easy way to test is to disable and re-enable the SNMP Agent. In this example the *Enable SNMPv3* and *Enable SNMPv1/2c Agent* checkboxes would be unchecked and we'd delete the only SNMP trap sink. Our SNMP manager entity should receive an *nsNotifyShutdown* ( oid object 1.3.6.1.4.1.8072.4.0.2 ) trap.

Navigate back into the SNMP settings and re-add the SNMP trap sink to then receive a *coldStart* trap.

## 6.3 Receiving a trap

### 6.3.1 Net-SNMP CLI

The example here is not a complete Net-SNMP manager entity setup and is only used to prove working receipt of an inform message.

For this example the master SNMP entity is running on CentOS.

---

The *Net-SNMP* package needs to be installed.

```
> yum install net-snmp
```

In addition, a valid `snmptrapd.conf` needs to be present.

```
#
# Example snmptrapd.conf file for bridgeworks demo
#

authCommunity log,execute,net bridgeworks

# Specific SNMPv3 user addition:
#
# bridgeworks = the username of the user to create
# SHA = authentication type
# demoAuth = authentication password
# AES = privacy type
# demoPriv = privacy password
createUser bridgeworks SHA demoAuth AES demoPriv

# Give the user permissions and restrict to minimum of 'priv'
# priv = authPriv mode (authentication and privacy enabled)
authUser log,execute,net bridgeworks priv
```

The `snmptrapd` process can now be started, the previously created `snmptrapd.conf` file may need to be explicitly referenced if it is not in a default SNMP config path. In this example the file is stored in the home directory of the root user.

```
> snmptrapd -m ALL -c /root/.snmp/snmptrapd.conf
```

The output from `snmptrapd` by default goes to the system log, so running a `tail -f` will watch the logs constantly. When a trap is received a summary of it's information and the agent entity address will be printed.

```
> tail -f /var/log/messages | grep snmptrapd
```

```
Feb 19 15:44:23 localhost snmptrapd[12564]: NET-SNMP version 5.7.2
```

```
Feb 19 15:45:03 localhost snmptrapd[12564]: hostname.test.domain [UDP: [10.10.64
.24]:34610->[10.10.10.86]:162]: Trap , DISMAN-EVENT-MIB::sysUpTimeInstance = Tim
eticks: (255449) 0:42:34.49, SNMPv2-MIB::snmpTrapOID.0 = OID: NET-SNMP-AGENT-MIB
::nsNotifyRestart, SNMPv2-MIB::snmpTrapEnterprise.0 = OID: NET-SNMP-MIB::netSnmp
NotificationPrefix
```

```
Feb 19 15:45:09 localhost snmptrapd[12564]: hostname.test.domain [UDP: [10.10.64
.24]:34610->[10.10.10.86]:162]: Trap , DISMAN-EVENT-MIB::sysUpTimeInstance = Tim
```

---

```
eticks: (256120) 0:42:41.20, SNMPv2-MIB::snmpTrapOID.0 = OID: NET-SNMP-AGENT-MIB::nsNotifyShutdown, SNMPv2-MIB::snmpTrapEnterprise.0 = OID: NET-SNMP-MIB::netSmpNotificationPrefix
```

```
Feb 19 15:45:12 localhost snmptrapd[12564]: hostname.test.domain [UDP: [10.10.64.24]:43339->[10.10.10.86]:162]: Trap , DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (8) 0:00:00.08, SNMPv2-MIB::snmpTrapOID.0 = OID: SNMPv2-MIB::coldStart, SNMPv2-MIB::snmpTrapEnterprise.0 = OID: NET-SNMP-MIB::netSmpAgentOIDs.10
```

In the terminal example above there are 4 log lines output from the `tail` command.

- Log 1 This line is printed when `snmptrapd` starts.
- Log 2 is the `nsNotifyRestart` trap. This trap was sent when the new trap was added to the unit, it means that the SNMP service on the agent entity was reloaded. The reload in this case was to add the new trap to the SNMP service.
- Log 3 is the `nsNotifyShutdown` trap. This trap was sent when the SNMP service was disabled on the unit. It is sent when you remove the last trap sink while both SNMP versions are disabled or when the unit is turned off.
- Log 4 is the `coldStart` trap. This trap was sent when the SNMP service was re-enabled on the unit. This trap will be sent to a trap sink if it is added as the only trap sink and both SNMP versions are disabled or when the unit turns on.

`Snmptrapd` is now running in the background; be sure to kill the process when testing has been completed or new settings need to be enacted.

### 6.3.2 SNMPb GUI

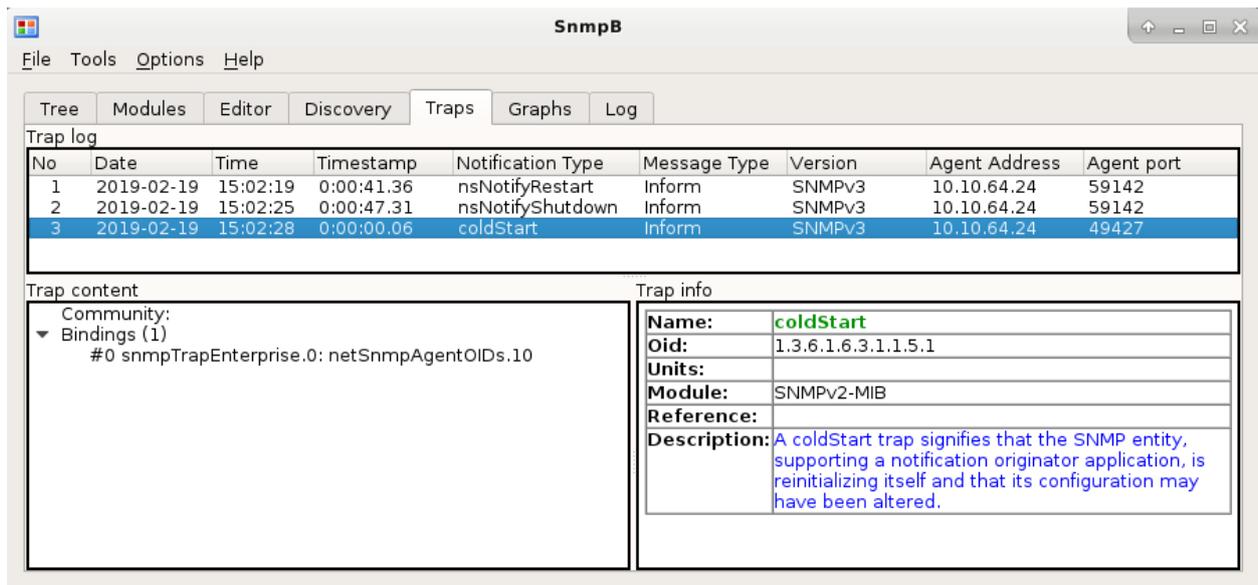
Throughout this guide the steps required for `SNMPb` to receive `SNMPv3` traps have all been put into place.

`SNMPv3` traps using `authPriv` mode require a valid `USM` account on both entities involved in sending and receiving the trap message.

See: Section [5.1.1.2: SNMPb GUI](#) for adding a `USM` profile, Section [5.2: Enabling authentication: authNoPriv Mode](#) for adding authentication, and Section [5.3: Enabling privacy: authPriv Mode](#) for adding privacy.

After the `USM` profile has been correctly set up, left click on the `Traps` tab in the `SNMPb` window.

Any trap messages that reach the system should appear in this log.



In the screenshot above there are 3 notable trap logs.

- **Entry 1 is the *nsNotifyRestart* trap.** This trap was sent when the new trap was added to the unit, it means that the SNMP service on the agent entity was reloaded. The reload in this case was to add the new trap to the SNMP service.
- **Entry 2 is the *nsNotifyShutdown* trap.** This trap was sent when the SNMP service was disabled on the unit. It is sent only when you remove the last trap sink while both SNMP versions are disabled or when the unit is turned off.
- **Entry 3 is the *coldStart* trap.** This trap was sent when the SNMP service was re-enabled on the unit. This trap will be sent to a trap sink if it is added as the only trap sink and both SNMP versions are disabled or when the unit turns on.

---

# 7 MIBs

MIBs have been mentioned previously in this guide. They map OIDs to human readable names. This chapter tells you how to install *snmp-mibs-downloader* to get the commonly used MIBs, and download MIBs specific to Bridgeworks products, then use them with Net-SNMP.

## 7.1 SNMP-MIBs-Downloader

Data common to all network devices is described in core MIBs which were published over time in various RFCs (Request For Comment). *Snmp-mibs-downloader* is a package that contains all of these. On Ubuntu systems it can be installed with:

```
sudo apt install snmp-mibs-downloader
```

If this doesn't work you may need to activate the non-free repository. This can be done by editing *etc/apt/sources.list* to have *multiverse* at the end of all lines starting with *deb*. Then retry the command above. Below is an example *sources.list* line for ubuntu.

```
deb http://archive.ubuntu.com/ubuntu <OSversion>-security main multiverse universe
```

After it's installed, running any SNMP command with `-m ALL` in should result in OIDs being replaced by names.

---

## 7.2 Bridgeworks MIBs

Bridgeworks have been allocated the Private Enterprise Number (PEN) 49599 and use the corresponding OID, 1.3.6.1.4.1.49599, and its subtree for our own data. This structure is documented in the Bridgeworks MIBs. These MIBs are needed to translate information into human readable form.

The first step is to download the Bridgeworks MIBs from the unit. Log in and navigate to the *SNMP* page via *Service Control*. Click the *Click Here to Download* button under *Download MIB Files* to download our MIB files in a zip archive.

Once extracted, all .mib files need to be placed in Net-SNMP's default MIB location so they'll be included when you put `-m ALL` in your queries. If you're unsure where the default MIB location is, run `snmpget -h`. MIBs' default location is in *General Options* on the line below `-M DIR[:...]`.

You can check if all MIBs are included by running a `snmpget` to your unit at the `sysObjectID` OID. Below is an example of this, you'll have to substitute in the management IP address and protocol details of your unit.

```
snmpget -m ALL -v 2c -c public 10.10.72.18 1.3.6.1.2.1.1.2.0
```

If the result resembles the output below, you have successfully installed both *SNMP-MIBs-downloader* and *Bridgeworks MIBs*. *SNMP-MIBs-downloader* has translated the OID you fetched as the `sysObjectID`, and *Bridgework's MIB* has found which product code the OID stored there corresponds to.

```
SNMPv2-MIB::sysObjectID.0 = OID: BW-PRODUCTS-MIB::bridgeworksVPR460
```

---

## 8 Useful Links

The following section contains links to other guides and FAQs. Support is available through our website: <https://support.4bridgeworks.com/>

The following resources are available online:

- [User Manuals](#)
- [Installation Guides](#)
- [General FAQ](#)
- [AWS FAQ](#)

If your question is not answered in our documentation, please [submit a ticket](#) through our website.