User Management in NMIS8

Introduction

NMIS 8.4G introduced general availability of the full authorisation and authentication model in NMIS using internal or external authentication methods. This ensures that NMIS remains an enterprise class network management system. Once authenticated a user is looked up in the internal authorisation system to determine their role, this role is customisable and extensible.

The implementation of the authorisation, limits groups of users by roles to specific things they can view and do with NMIS, and groups of devices/nodes on which they can do it. This is commonly referred to as role based access control.

Authentication Methods

The following table lists the NMIS configuration option and the type of authentication which it works with. These settings are made in /usr/local/nmis8/conf/Config.nmis.

THESE AUTH METHODS REQUIRE OPTIONAL PERL MODULES

Each of the authentication methods require there own Perl Modules - you can install them with the cpan command and the module name e.g. *"cpan Net::LDAP" or* you can check if a module is installed with e.g. *"cpan -D Net::LDAP"*

Method	Description
apache	Apache will perform authentication and provide an authenticated user to NMIS, which will have authorisation policies applied.
htpasswd	NMIS will use the users defined in the NMIS Users file, by default /usr/local/nmis8/conf/users.dat

ldap	NMIS will use the configured LDAP server to perform authentication
	
	Requires Optional Perl Module: Net::LDAP
	
	Config:
	auth_ldap_server => 'host[:port]'
	auth_ldap_attr => '' # attributes to match to username, can be blank, then defaults to ('uid','cn')
	auth_ldap_context => 'ou=people,dc=opmantek,dc=com', # base of context to attempt to bind to

ldaps (secure)	NMIS will use the configured LDAP server to perform authentication
	
	Requires Optional Perl Modules: IO::Socket::SSL and Net::LDAPS
	
	auth_ldaps_server => 'host[:port]'
	auth_ldap_attr => '' # attributes to match to username, can be blank, then defaults to ('uid','cn')
	auth_ldap_context => 'ou=people,dc=opmantek,dc=com', # base of context to attempt to bind to

ms-ldap	NMIS will use the configured Microsoft Active Directory (LDAP) server to perform authentication
	
	Requires Optional Perl Module: Net::LDAP
	
	Config:
	auth_ms_ldap_server => 'host[:port]'
	auth_ms_ldap_dn_acc => '' # the DomainName\account to bind with
	auth_ms_ldap_dn_psw => 'password'
	auth_ms_ldap_attr => 'sAMAccountName', # attribute to match to username
	auth_ms_ldap_base => 'dc=corp,dc=opmantek,dc=com' # base to search from

ms-ldaps (secure)	NMIS will use the configured Microsoft Active Directory (LDAP) server to perform authentication
	
	Requires Optional Perl Modules: IO::Socket::SSL and Net::LDAPS
	
	Config:
	auth_ms_ldaps_server => 'host[:port]'
	auth_ms_ldap_dn_acc => '' # the DomainName\account to bind with
	auth_ms_ldap_dn_psw => 'password'
	auth_ms_ldap_attr => 'sAMAccountName', # attribute to match to username
	auth_ms_ldap_base => 'dc=corp,dc=opmantek,dc=com' # base to search from
	
	If an internal CA is used for the AD server's SSL that CA's root certificate should be imported for SSL trust.

pam	Available in NMIS versions 8.6.8G and newer. • Debian/Ubuntu: The webserver user must be able to read the `/etc/shadow` file, which can be achieved by adding the webserver user to the `shadow` group: Run `sudo adduser www-data shadow` • CentOS/RedHat: CentOS and RHEL require further configuration steps to enable PAM. 1. Create a PAM configuration file for NMIS as /etc/pam.d/nmis. You might clone /etc/pam.d/login and adjust that. Unless an 'nmis' PAM configuration file is present, the default configuration from /etc/pam.conf will reject any authentication attempts. 2. Allow web user to read `/etc/shadow` .
radius	NMIS will use the configured radius server (Cisco ACS or Steel Belted Radius for example) Requires Optional Perl Modules: Authen::Simple::RADIUS • Install from CPAN with command: ◦ sudo cpanm Authen::Simple::RADIUS --sudo Config: auth_radius_server => 'host:port' auth_radius_secret => 'secret'
tacacs	NMIS will use the configured Tacacs+ server (Cisco ACS for example) Requires Optional Perl Modules: Authen::TacacsPlus Config: auth_tacacs_server => 'host:port' auth_tacacs_secret => 'secret' # Also known as the "Key"
ConnectWise	1. Setup ConnectWise API 'auth_cw_server' => '1.2.3.4', # IP address of ConnectWise Server 'auth_cw_company_id' => 'COMPANY', # Company name in ConnectWise 'auth_cw_public_key' => 'xxxxxxXXXXXxxxxx', # ConnectWise Public Key 'auth_cw_private_key' => 'yyyyyYYYYYyyyyy', 2. Setup the system to use the auth method. 'auth_method_1' => 'connectwise', 'auth_method_2' => 'ms-ldap', 'auth_method_3' => 'htpasswd',

Configuration of the External Authentications

In the NMIS configuration you can configure multiple methods which are used for auth failure, so if ms-ldap fails, it will fail back to htpasswd for example. This means if you set auth_method_1 to be ldap and auth_method_2 to be htpasswd, and login with the default NMIS credentials (and you have not changed the password), the authentication for LDAP will fail, and then authentication with the users.dat will succeed and the user will be logged in. The limit for different auth_method variables is 3.

It is important to change your default passwords if you expect any level of security.

Default Privilege Level for Authenticated Users

When accessing NMIS, you have a choice on how to handle authenticated users who do not have authorisations defined, you can reject them, or you can allow them default access.

This is so that you do not have to define every user in the system if the authentication system is providing a reduced list of users, to have the users become an operator or guest by default and be able to see all groups of devices, the following would apply.

'auth_default_privilege' => 'guest',
'auth_default_groups' => 'all',

To prevent default authorisation, simply define them as blank, which is the default in the NMIS8 Install configuration.

Locking accounts after N failed login attempts

In NMIS versions 8.5.12G and newer you can configure optional account locking. This feature is not enabled by default as it could be abused for denial-of-service attacks.

If you set the configuration option `auth_lockout_after` to a positive number `N`, then the account in question will be locked after `N` consecutive failed login attempts. If the optional configuration item `server_admin` holds an email address, a notification email will be sent to the given administrator address.

Locked accounts can be re-enabled from the GUI: visit the System -> System Configuration -> Users page, and click on the option "reset login count" for the locked account.

From the command line re-enabling is also possibly: simply remove the file `/usr/local/nmis8/var/nmis_system/auth_failures/<accountname>.json`.

NMIS Single Sign On

NMIS 8.5 and newer support Single Sign On for NMIS installations spanning a whole organisation (or subdomain).
In version 8.6.3G we've added support for Single Sign On between NMIS and Opmantek applications, either for a single installation or spanning an organisation.

Configuring SSO: NMIS to Opmantek Applications (8.6.3G and newer)

Compatility Aspects

A number of problematic corner-cases were discovered and fixed in May 2018, which have unfortunately required certain changes that are not backwards-compatible.
The following table lists the scenarios:

NMIS	Opmantek Apps	NMIS-Opmantek SSO
before 8.6.3G	any version	not available
8.6.3 or 8.6.4	only application releases before 22.5.2018 present on your system	available but not perfectly robust in certain circumstances
8.6.5 and newer	only releases older than 22.5.2018 present	not available
8.6.5 and newer	at least one application release newer than 22.5.2018 present	available

SSO between NMIS and OMK Applications on one system

To configure NMIS to share authentication cookies with Opmantek Applications, it is necessary that you

- first change the configuration item `auth_cookie_flavour` from the default "nmis" to "omk",
- then change the configuration item `auth_web_key` to the same value as the OMK application's first `omkd_secrets` configuration.

Both the Cookie Type (or flavour) and Authentication Secret (or key) settings can be changed using the Basic Setup dialog, or the NMIS Configuration dialog (they're in section "authentication").
To gather the Opmantek application secret, you can either open `/usr/local/omk/conf/opCommon.json` with an editor (look for `omkd_secrets`), or you can ask the patch_config tool for the value of that setting, like in the following example:

```
$ /usr/local/omk/bin/patch_config.exe -r /usr/local/omk/conf/opCommon.json /omkd/omkd_secrets[0]
CHANGE_ME_askdfal2332lkwjflk
```

If you choose to set up NMIS' current authentication secret for the OMK Applications instead of the other way round, then please make sure to restart the OMK Application daemon to activate your changes.

It is quite likely that you will have to flush your browser cache once after making these adjustments, to enforce that no old cookies interfere with the authentication system.

SSO between NMIS and OMK Applications across a whole organisation

To provide SSO for all involved applications, you need to make the configuration adjustments listed in the previous section **and all of the following** changes:

- set up the same SSO Domain (using the `auth_sso_domain` configuration setting) on all involved systems for **both** NMIS and OMK Applications,
- ensure that the chosen SSO Domain meets the requirements of having two or more periods (ok: ".mydom.ain" or "suborg.myorg.com", not ok: ".com"),
- and ensure that all involved systems are accessed solely by their fully qualified domain names (FQDN) which must belong to the SSO Domain.

In NMIS the `auth_sso_domain` setting can be reached using the NMIS Configuration dialog, under section "authentication".

For the OMK Applications you have to open the configuration file `/usr/local/omk/conf/opCommon.nmis` with an editor and search for `omkd_secrets` (under `omkd`) or `auth_sso_domain` (under `authentication`). Please note that you must restart the OMK Application daemon to activate any changes.

Configuring SSO: NMIS to NMIS

In version 8.6.3G and newer you need to set the Cookie Type (`auth_cookie_flavour` configuration item) to "nmis", if you want the have SSO *only* across NMIS installations.
In older versions that is the only SSO choice.

To activate this feature, you need to

- use a fully qualified domain name (FQDN) for all your NMIS servers, e.g. nmis1.domain.com,
- and modify the NMIS configuration of **all** participating NMIS installations to contain the same SSO Domain **and** the same shared authentication key.

Here's an example of the relevant parts of the configuration file `/usr/local/nmis8/conf/Config.nmis`:

```
'auth_sso_domain' => '.domain.com',
'auth_web_key' => 'thisismysecretkey',
```

Please note that two or more periods are required in the domain name, so if your company is AJAX Cleaning and your domain name is ajaxcleaning.com this configuration would be:

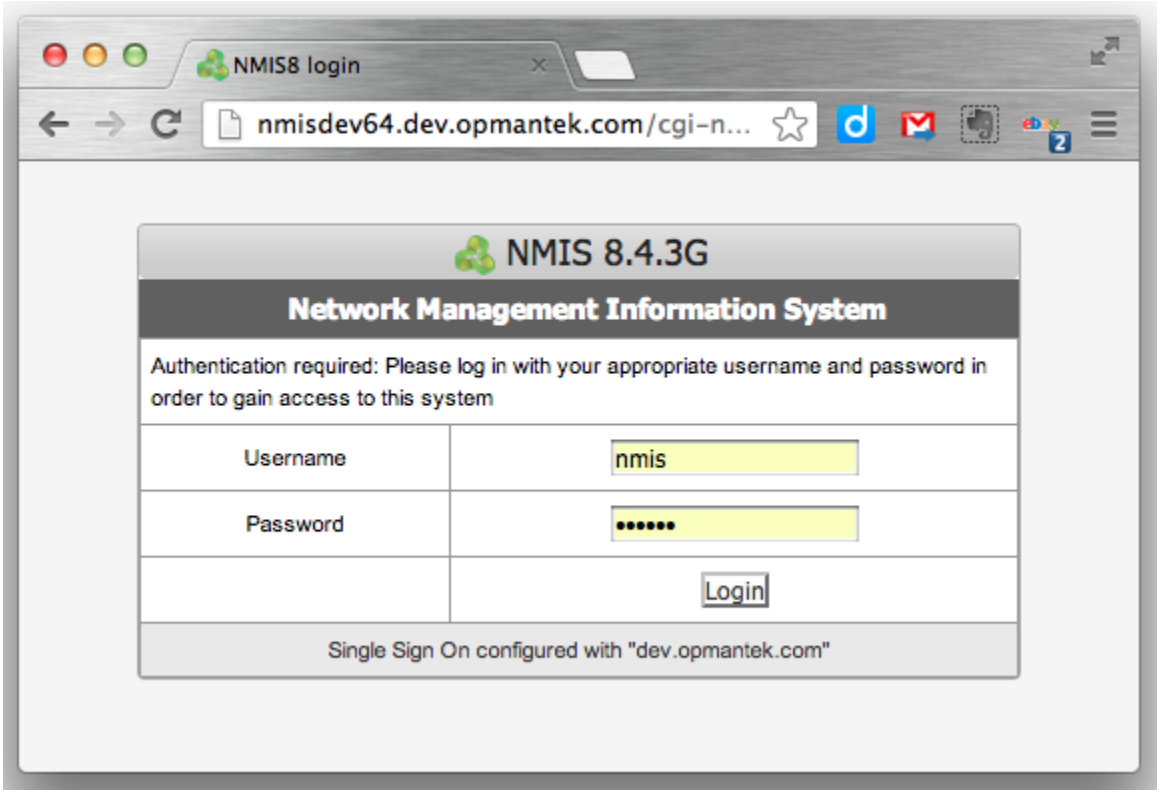
```
'auth_sso_domain' => '.ajaxcleaning.com',
```

Accessing NMIS with Single Sign On

When you are accessing the NMIS server after this is configured you must then use the FQDN. In our development labs we use dev.opmantek.com which contains two periods, so we can configure it like this:

```
'auth_sso_domain' => 'dev.opmantek.com',
```

Then when we browse to the servers, we use the full name of the server, e.g. nmisdev64.dev.opmantek.com and the login banner tells us that Single Sign On is configured with "dev.opmantek.com", I can enter the password and access the server.



If I were to use the IP address to access the server, authentication will succeed but the cookie will not be created correctly and you will be logged out again.

When NMIS Single Sign On is configured you MUST use the Full Qualified Domain Name to access the server.

User administration when using the htpasswd Method

Encryption Methods

Apache (and its `htpasswd` tool) support a number of different password hashing mechanisms. In the past `htpasswd`'s default mechanism was "crypt" but on most operating systems that has been changed to "md5" (as it resists dictionary attacks much better).

However: NMIS in versions up to and including 8.5G only supports the "crypt" mechanism. This means that you have to explicitely specify the appropriate method when you run `htpasswd`.

Adding a user for Authentication

To add a new user to NMIS8 (while it is using htpasswd, or Apache for Authentication - and assuming that Apache has not already been integrated with LDAP, Active Directory, Radius, or the like) you will need to create a new Web Based user. This is done using the following commands, using *testuser* as example:

```
cd /usr/local/nmis8/conf # adjust that if nmis8 is not installed in the default location
# for encryption method crypt:
htpasswd -d users.dat testuser
# ...then just follow the prompts
# for encryption method md5, BUT only for NMIS newer than 8.5G:
htpasswd -m users.dat testuser
```

This adds the user *testuser* for the purpose of Authentication. Now the application needs to be also told about the users' Authorisation.

User names and case

User names in many systems are not case sensitive, so NMIS will handle usernames in lower case, when adding users to Users.nmis, ensure that the name is all in LOWER CASE.

Spaces in User Names

At great expense to Opmantek, support for usernames with spaces has been added, this is in the next release to be numbered 8.3.14G or higher.

Authorisation in NMIS

NMIS 8 uses the concepts "Privileges", "Access Policy" and "Groups" to determine what resources or actions a particular user should have access to.

- The Groups setting for a user lists all the node groups this user may see (there is also the wildcard "all" with obvious meaning). Every node in NMIS belongs to exactly one group, but a user can be associated with any number of groups. Please note that group visibility checks are performed **independent** of the other authorisation mechanisms.
- The Privilege setting describes, on a very high level, the operations this user should be able to perform; these also control the visibility of certain parts of the NMIS GUI. A user account has exactly one privilege. Each privilege has a (free-form) name.
- Each privilege is translated into a single numeric access level. As a user account has exactly one privilege, it also has exactly one access level. By default NMIS uses level numbers 0 to 5.
- Particular operations and views are associated with Access Policy elements, and these list what access levels are considered sufficient for granting access.

This infrastructure is configured using three configuration files:

1. Users.nmis defines the existing users and their privileges. In the GUI this is accessibe in the System menu, under System Configuration -> Users.
2. PrivMap.nmis defines the mapping from textual privilege to numeric access level. In the GUI you'll find that under System -> System Configuration -> Privilege Map.
3. Access.nmis defines which numeric access levels shall have access to what operations and views. The GUI presents this under System -> System Configuration -> Access Policy.
   Access levels are treated independently. If a user belongs to level 3 for example, then that does not imply anything about his or her access to level 4 or level 2 operations.
   Please note that the GUI for this lists the access levels by their privilege name (inverse mapping via PrivMap), whereas the underlying configuration file uses the numeric levels exclusively.

Setting up a User's Authorisations

Login to the NMIS Portal, as an administration user, the normal URL is http://nmisserver/cgi-nmis8/nmiscgi.pl

Using the menu access "System -> System Configuration -> Users", select "add" from the top right, and then complete the form, specifying the User which matches the user added using htpasswd, specify Privilege and Groups, using "all" if all groups are permitted, multiple groups can be selected.

Related Topics

- OMK Authentication Methods