NetIQ Advanced Authentication Framework

How To

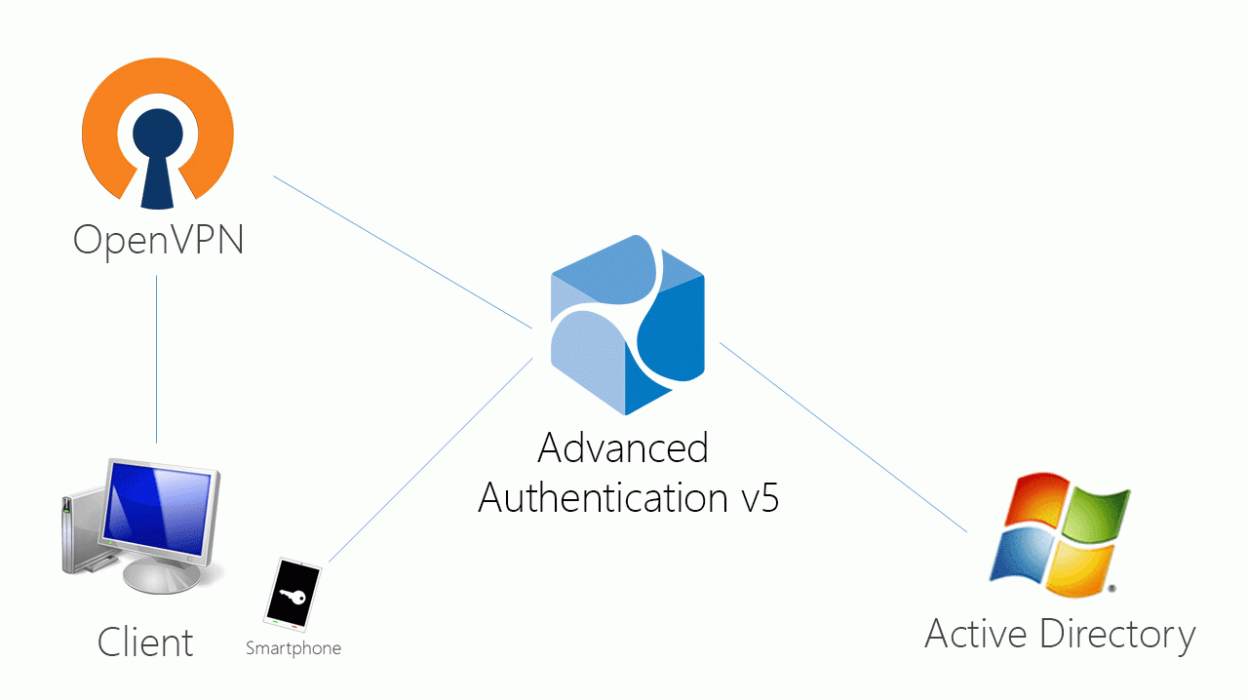
Version 5.1.0

Table of Contents

How to configure advanced authentication in OpenVPN

These instructions will help you to configure integration of NetIQ Advanced Authentication Framework Appliance Edition with the OpenVPN virtual appliance to refuse non-secure passwords in OpenVPN connection.

The advanced authentication in OpenVPN is represented on the following diagram.



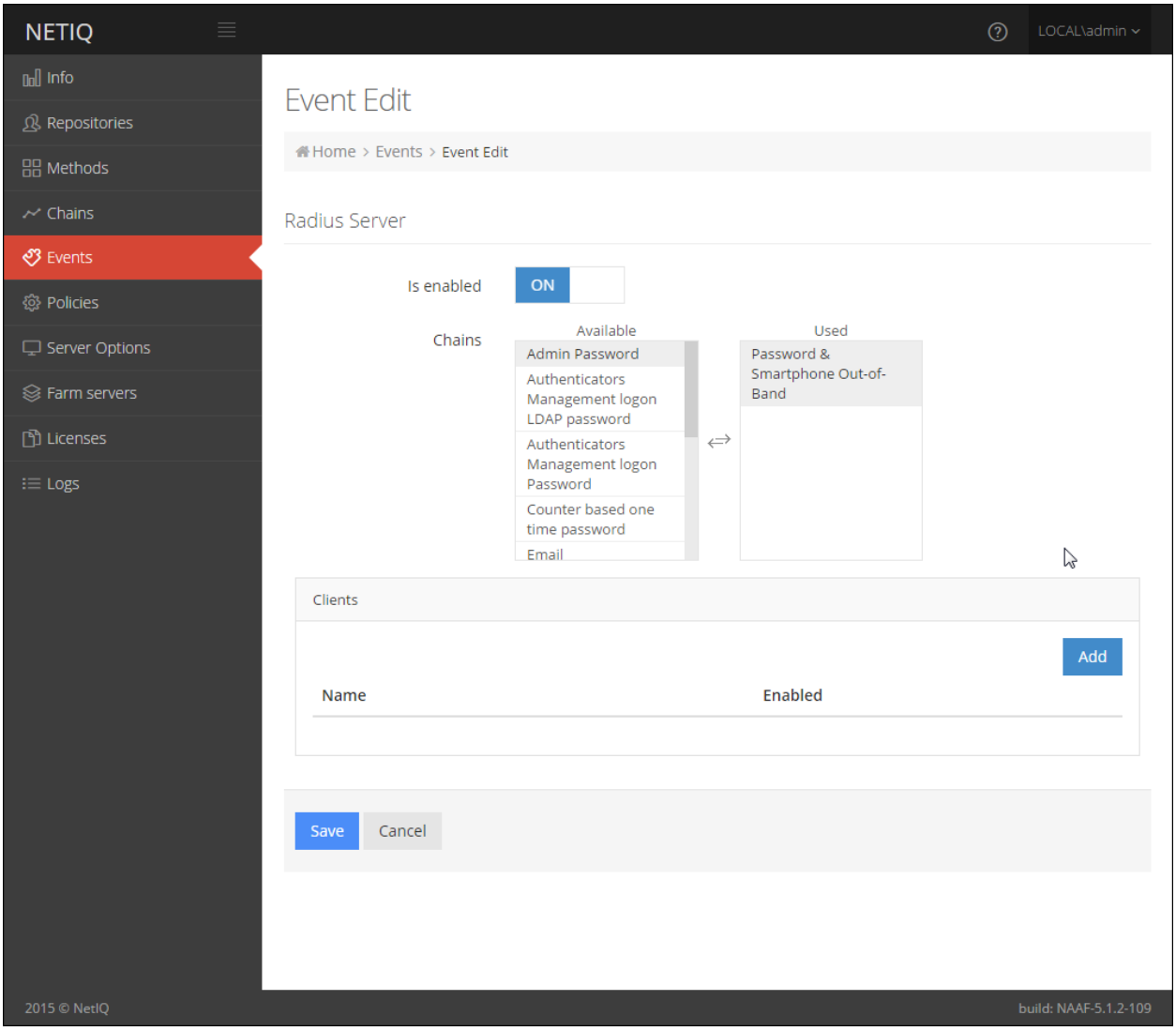
To get started, ensure that you have:

- OpenVPN v2 appliance (version 2.0.10 was used to prepare these instructions)
- NetIQ v5 appliance (version 5.1.1 was used to prepare these instructions) with the already configured repository

Configure the NetIQ RADIUS server:

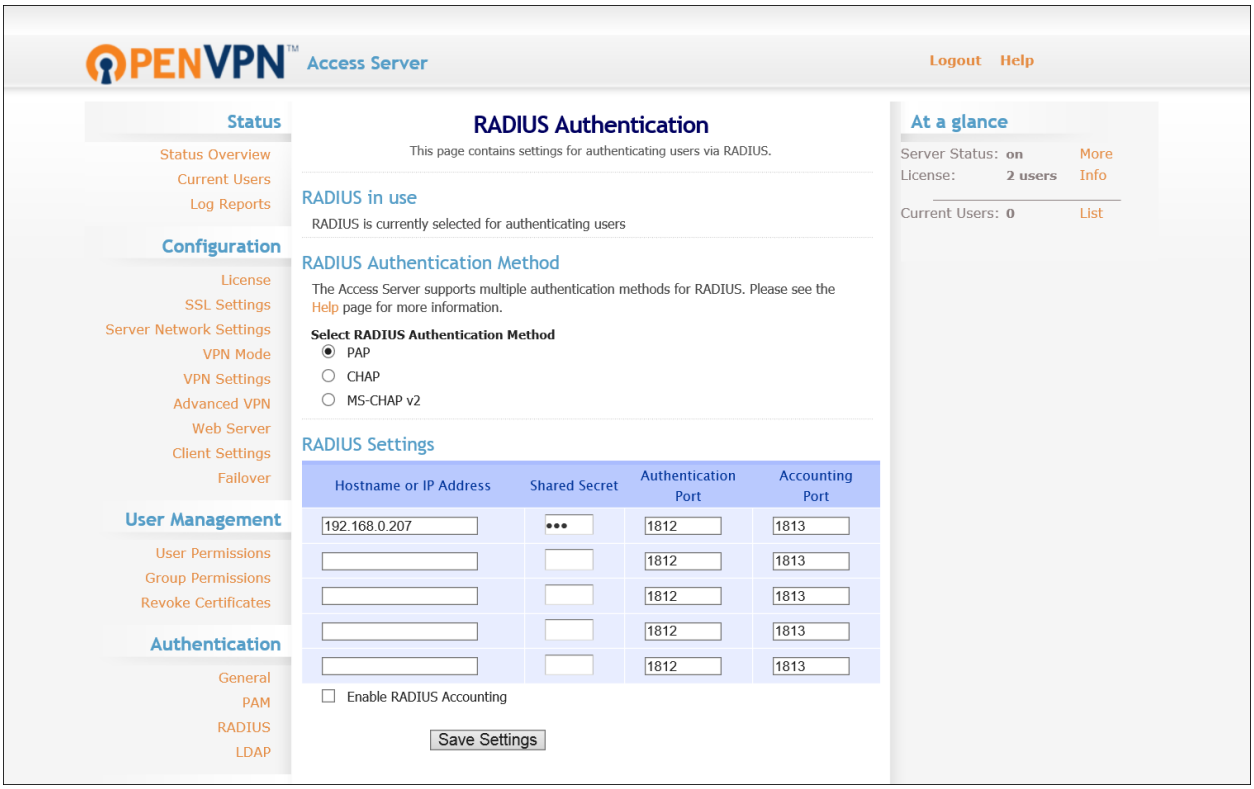
1. Open the NetIQ Admin Interface.
2. Go to the **Events** section.
3. Open properties of the **Radius Server** event.
4. Set the **Radius Server** event to the **ON** mode.
5. Select one or more chains from the list of **Used** chains (make sure that they are enabled and set to the users group in the **Chains** section).

6. Add a **Client**, enter an IP address of the OpenVPN appliance, specify a secret, confirm it and set the **Enabled** option.
7. Click the **Save** button in the **Client** string. Click the **Save** button at the bottom of the **Events** view to save changes.



Configure the OpenVPN appliance:

1. Open the **OpenVPN Access Server** site.
2. Go to the **Authentication** - **RADIUS** section.
3. Enable the **RADIUS** authentication.
4. Select **PAP** authentication method.
5. Add an IP address of the NetIQ v5 appliance and enter the secret.



If you have one **Used** chain selected in the **Radius Server** settings, to connect to OpenVPN, please enter the <repository name>\<username> or only <username> if you have set the default repo name in **Policies - Login options** section of the NetIQ v5 appliance.

If you have multiple **Used** chains selected, to connect to OpenVPN, in the username field after the entered <username> and space you need to enter a **Short name** of the necessary chain (the **Short name** can be selected in **Chains** section of the NetIQ v5 appliance).

Please note that some of the available authentication methods require correct time on the OpenVPN appliance. You can sync the time of the OpenVPN appliance using the following commands:

```
/etc/init.d/ntp stop
/usr/sbin/ntpdate pool.ntp.org
```

How to configure load balancer for NAAF cluster

Load balancer can be installed and configured via third party software. Below is an example of how to install and configure nginx as load balancer on Ubuntu 14.

Target configuration:

	Hostname	***IP address***	***Role***	***Operation System***
Domain controller	win-dc	192.168.1.42	AD DS, DNS	Windows Server 2008 R2
NAAF 5.1 master	naafmaster	192.168.1.43	NAAF Master server	NAAF 5.1.2
NAAF 5.1 slave	naafslave	192.168.1.41	NAAF Slave server	NAAF 5.1.2
Load balancer	loadbalancer	192.168.1.40	Nginx load balancer	Ubuntu 14

Before starting the configuration, please make sure that the following requirements are fulfilled:

- Repository is configured in NAAF appliance.
- Both NAAF servers are installed and configured as Master and Slave.
- Appropriate entries are added to DNS.
- Ubuntu 14 is installed.

To configure Load Balancer for NAAF cluster, it is required to install nginx on Ubunto 14 and configure it.

Installing nginx on Ubunto 14

To install nginx on Ubunto 14, follow the steps:

1. Open the following source list:
   - `sudo nano /etc/apt/sources.list`
2. Add necessary entries:
   - `deb http://nginx.org/packages/ubuntu/ trusty nginx`
   - `deb-src http://nginx.org/packages/ubuntu/ trusty nginx`
3. Update repository and install nginx:
   - `apt-get update`
   - `apt-get install nginx`
4. Start nginx and make sure that web server is working:
   - `sudo service nginx restart`
5. Open your browser and go to web server http://192.168.1.40 or http://loadbalancer.

Configuring nginx

The following load balancing mechanisms/methods are supported in nginx:

- **round-robin** - requests to the application servers that are distributed in a round-robin fashion
- **least-connected** - next request assigned to the server with the least number of active connections
- **ip-hash** - a hash-function that is used to determine what server should be selected for the next request (based on the client's IP address)

This article describes only round-robin configuration. To configure nginx, follow the steps:

1. Backup original configuration file: `sudo cp /etc/nginx/nginx.conf /etc/nginx/nginx.conf_original`.
2. Open the **nginx.conf** file and replace with following:

```
user nginx;
error_log /var/log/nginx/error.log warn; # error log location
pid /var/run/nginx.pid; # process id file

# limit number of open sockets. Debian default max is 1024, ensure
nginx not open all the sockets.
worker_processes 1;
events {
     worker_connections 900; # 512 is default
}
# worker_processes auto; # ssl needs CPU

http {
     include /etc/nginx/mime.types;
```

```
    default_type application/octet-stream;

    log_format main '$remote_addr - $remote_user [$time_local] "$re-
quest" '
                '$status $body_bytes_sent "$http_referer" '
                '"$http_user_agent" "$http_x_forwarded_for"';

    access_log /var/log/nginx/access.log main; # access log location

    sendfile on;
    # keepalive default is 75
    # keepalive_timeout 10;

    gzip on;
    gzip_static on;
    gzip_comp_level 5;
    gzip_disable msie6;
    gzip_min_length 1000;
    gzip_proxied expired no-cache no-store private auth;
    gzip_vary on;
    gzip_types text/plain text/css application/json applic-
ation/javascript
          text/xml application/xml application/rss+xml applic-
ation/atom+xml;

    ssl_certificate /etc/nginx/cert.pem;
    ssl_certificate_key /etc/nginx/cert.pem;
    ssl_session_cache shared:SSL:2m; # 1m stores 4000 sessions,
default expire 5 min
    ssl_protocols TLSv1 TLSv1.1 TLSv1.2; # disable TLSv3 - POODLE
vulnerability

    resolver 192.168.1.42 valid=300s ipv6=off; # ip address of DNS
    resolver_timeout 10s;
    upstream web {
          #server naafmaster.authasas.local:443 resolve;
          #server naafslave.authasas.local:443 resolve;
          server 192.168.1.43:443;
          server 192.168.1.41:443;
          }

    server {
          #listen 80;
          listen 443 ssl;
          location / {
                proxy_pass https://web;
                proxy_set_header HOST $host;
                proxy_set_header X-Forwarded-Proto $scheme;
```

```
                proxy_set_header X-Real-IP $remote_addr;
                proxy_set_header X-Forwarded-For $proxy_add_x_for-
warded_for;

            }
        }
}
```

3. Copy certificate from any NAAF server in cluster from the directory **/etc/nginx/cert.pem** to the same directory on load balancer.
4. Go to https://loadbalancer/admin page and make sure that connection was redirected to NAAF cluster.

Nginx can be installed and configured on any Linux supported by nginx.

Additional information on nginx configuration can be found at http://nginx.org/en/docs/.