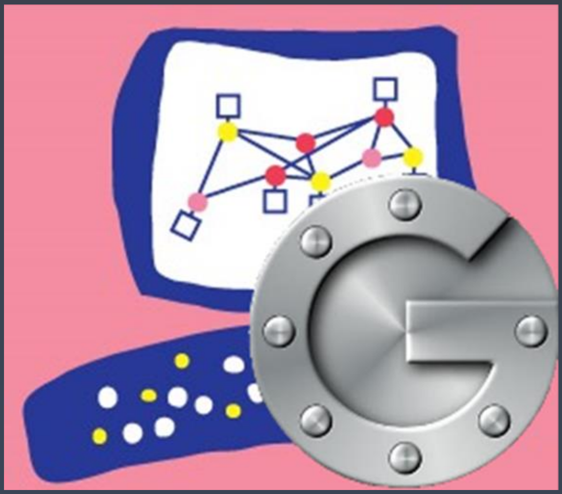
Higher Intelligence

Information Technology Infrastructure & Security Services



CHECKPOINT MFA WITH GOOGLE AUTHENTICATOR

Higher Intelligence
Information Technology Infrastructure & Security Services

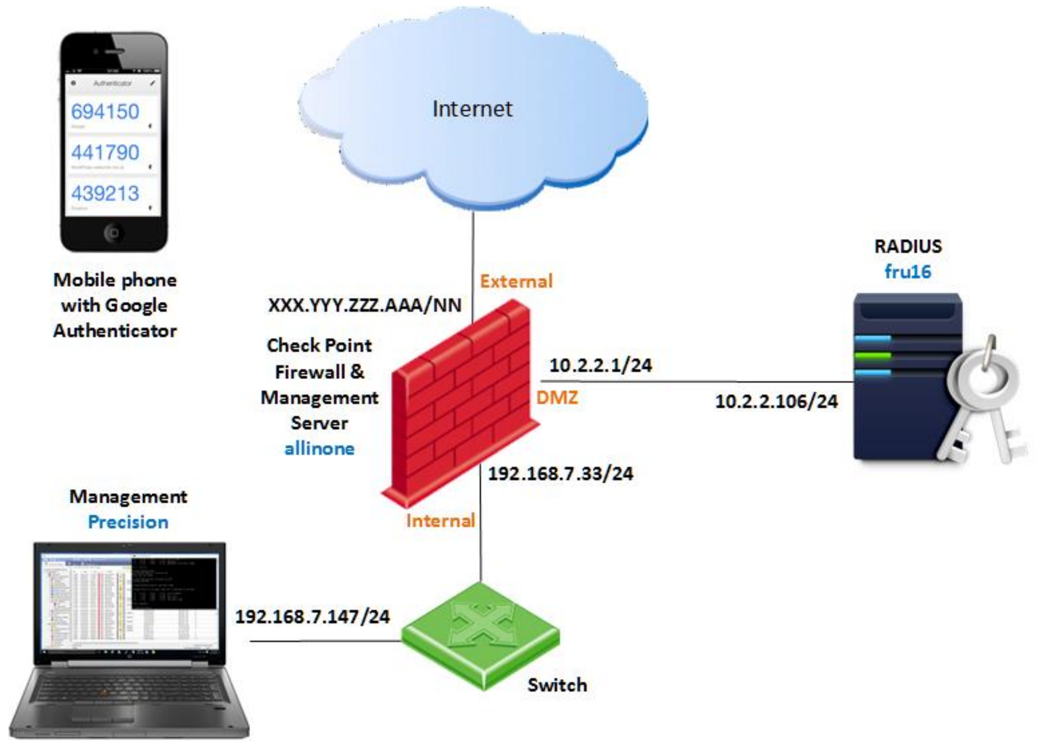
Vladimir Yakovlev 101010@higherintelligence.com

Most of us are familiar with Google Authenticator application and are already using it for a multitude of services. I would like to describe the procedures that will make it possible to authenticate administrators and users to Check Point Gaia Web UI and SSH as well as Smart Console applications and VPN using this application.

This setup requires creation and maintenance of local accounts on RADIUS server. To simplify user onboarding, QR codes will be generated for mobile Google Authenticator application registration.

This guide is created with beginners in mind and does not require extensive Linux experience.

Network diagram, hosts and IP addresses



IP addresses, host names and zone designations shown reflect those used in a lab environment for creation of this document. Change those according to your requirements.

Conventions:

```
# Green on grey denotes existing text in files or prompts.
Blue on grey indicates modified entries in files, commands and responses to the
prompts made by you. Commands should be followed by pressing Enter
```

```
# Code blocks are entities to be copied and pasted as directed
```

Highlighted text signifies notes of importance

Ctrl+x ; **Ctrl+w** ; **y** ; **n** ; **Enter** , etc., represent key combinations, single key presses or sequences.

When making changes in configuration files, comment out the lines you are changing and create modified entries under those. Create a commented-out delimiter above and below the changed entries to allow for easier searches later, i.e.:

```
#----------------------  changed-by-admin1  ------------------------
#original entry, commented-out
Modified entry
#--------------------------  end-change  ---------------------------
```

You can subsequently search configuration files for changes using “changed-by-admin1” pattern.

Setting up Ubuntu server

We'll be using the most widely deployed RADIUS server in the world, FreeRADIUS, running on Ubuntu distribution of Linux.

Install Ubuntu 16.04 LTS server with default settings on a platform of your choice. Do not encrypt user directories!!!

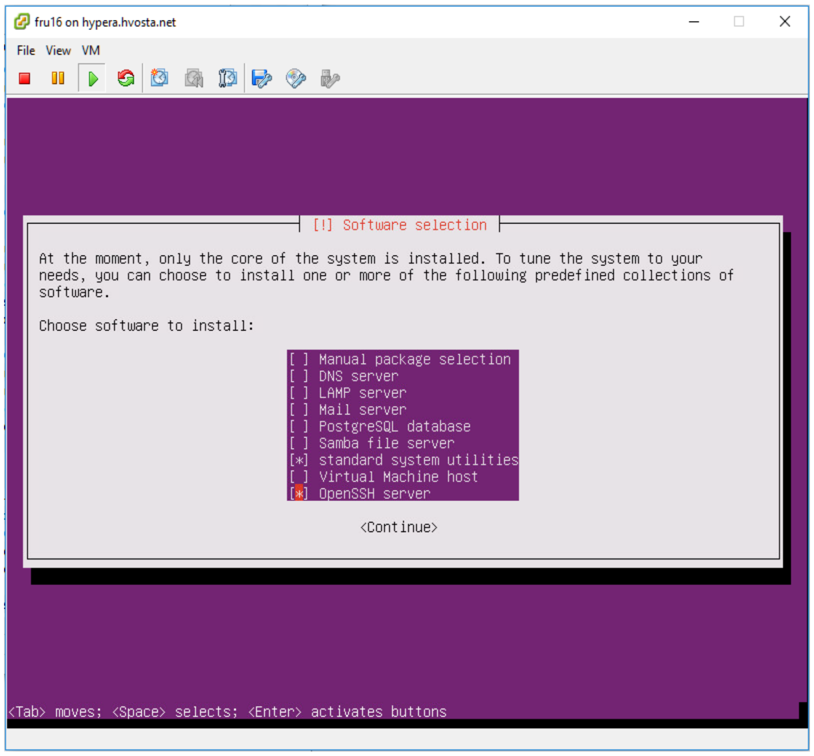
You will need to provide the Ubuntu server with Internet access, at least during initial configuration. Once configuration is completed, you may restrict its access to the Internet completely if you have internal Time Server(s). If you are relying on time-based tokens, your RADIUS server should retain capability to keep accurate time.

Depending on installation method and target, you may be prompted with configuration options during setup. If so, enter your hostname, user name, password and network settings as you go through the installation process. If some of the options shown in the table were not available during the installation, follow instructions below to make necessary changes.

Parameters (blue values used in the lab), note your own in the empty cells

Hostname (fru16)		Address (10.2.2.106)	
User (fradmin)		Netmask (255.255.255.0)	
Password (FR@dmin)		Network (10.2.2.0)	
		Broadcast (10.2.2.255)	
		Gateway (10.2.2.1)	
Time Zone	America/New_York	Domain (-//-)	
DNS-Resolvers	8.8.8.8 single space 8.8.8.4	DNS-Resolvers	

If prompted with "Software Selection" options during installation, include "Standard System Utilities" and "OpenSSH Server" and skip the "Installing OpenSSH" section below.



```
[!] Software selection

At the moment, only the core of the system is installed. To tune the system to your
needs, you can choose to install one or more of the following predefined collections of
software.

Choose software to install:

                    [ ] Manual package selection
                    [ ] DNS server
                    [ ] LAMP server
                    [ ] Mail server
                    [ ] PostgreSQL database
                    [ ] Samba file server
                    [*] standard system utilities
                    [ ] Virtual Machine host
                    [*] OpenSSH server

                              <Continue>

<Tab> moves; <Space> selects; <Enter> activates buttons
```

If you are proficient in Linux administration, install the following packages:

ntp, build-essential, libpam0g-dev, freeradius, git, libqrencode3 and libpam-google-authenticator.

Set the correct Time Zone, reboot the server and skip to page 6, Configuring Initial Firewall Settings. Use your own choice of editor; otherwise, follow the document.

Logon using credentials defined during installation.

Type:

```
sudo su
```

and press **Enter** . When prompted, use the password you have chosen during setup to elevate your privileges.

Configuring Network Settings for Ubuntu Server

Type (or copy and paste):

```
nano /etc/network/interfaces
```

Inside the Nano editor, navigate to the `iface` line located under `# The primary network interface`, comment it out and paste "**iface ens33 inet static**" below, (your interface name may differ, but will be in identical location):

```
# The primary network interface
auto ens33
#iface ens33 inet dhcp
iface ens33 inet static
```

Copy and paste, if you can, or type these lines under the "iface ens33 inet static", to configure IP address, gateway and DNS resolvers. Substitute values for address, mask, gateway and DNS servers with those pertinent to your network:

```
address 10.2.2.106
        netmask 255.255.255.0
        network 10.2.2.0
        broadcast 10.2.2.255
        gateway 10.2.2.1
        # dns-* options are implemented by the resolvconf package, if installed
        dns-nameservers 8.8.8.8 8.8.4.4
```

(Note the space between 8.8.8.8 and 8.8.4.4).

Press **Ctrl+x**; **y** and **Enter** to save changes and exit Nano.

Type:

```
sudo reboot now
```

to restart your Ubuntu server.

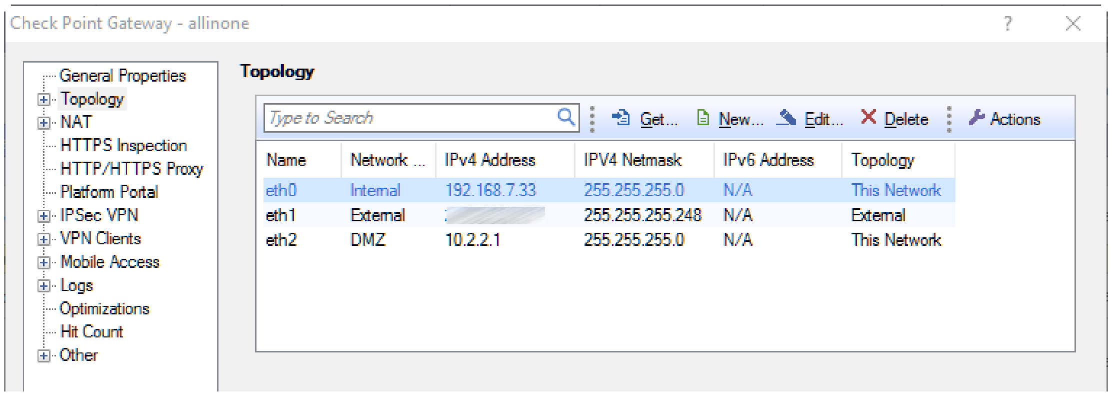
We will continue configuration of the Ubuntu and FreeRADIUS once initial configuration of our firewall is completed.

Configuring Initial Firewall Settings

At this point in the process, we must configure the firewall to allow for the following actions:

1. Access Ubuntu server from your management workstation
2. Allow Ubuntu to resolve names
3. Allow Ubuntu to sync time with NTP servers or pools
4. Permit Internet access from Ubuntu for installation and updates of additional software packages

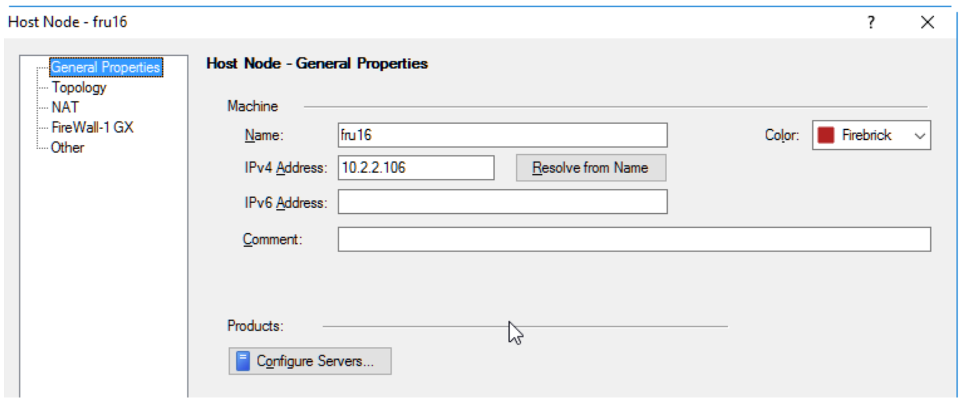
Our Check Point firewall topology, reflecting network diagram on Page 2, should look like this:



Topology

Name	Network ...	IPv4 Address	IPV4 Netmask	IPv6 Address	Topology
eth0	Internal	192.168.7.33	255.255.255.0	N/A	This Network
eth1	External	[redacted]	255.255.255.248	N/A	External
eth2	DMZ	10.2.2.1	255.255.255.0	N/A	This Network

Let's create the host object representing our Ubuntu server and name it "**fru16**", (abbreviated FreeRADIUS Ubuntu 16):



Host Node - General Properties

Machine

Name: fru16

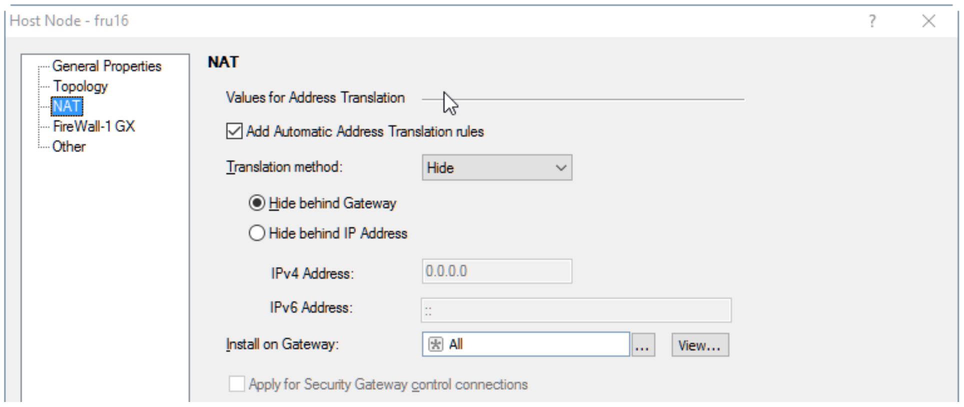
IPv4 Address: 10.2.2.106

IPv6 Address:

Comment:

Color: Firebrick

And its NAT properties set to "Add Automatic Address Translation rules" with Translation method set to "Hide".



Host Node - fru16

General Properties
Topology
NAT
FireWall-1 GX
Other

NAT

Values for Address Translation

☑ Add Automatic Address Translation rules

Translation method: Hide

● Hide behind Gateway
○ Hide behind IP Address

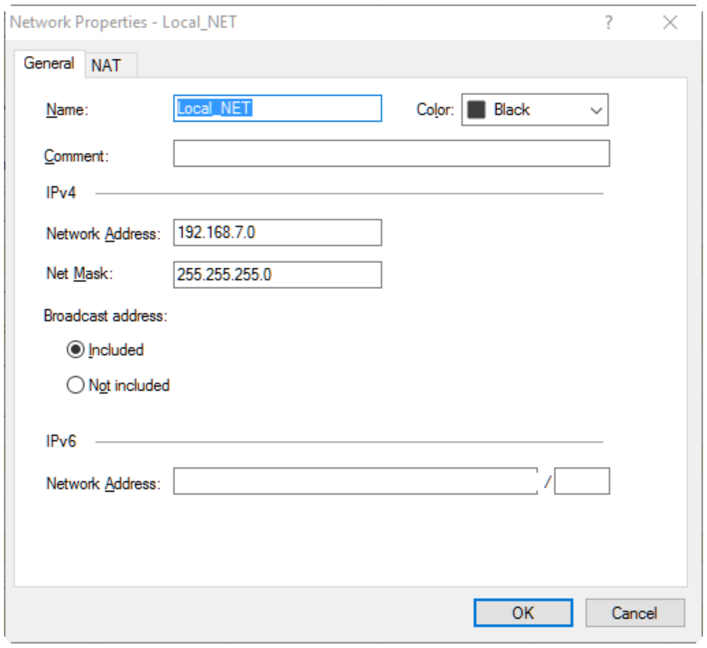
IPv4 Address: 0.0.0.0
IPv6 Address: ::

Install on Gateway: All

☐ Apply for Security Gateway control connections

Note, that in the future you may have to change the Translation method to "Static", if you'll decide to incorporate Google Authentication for users defined in FreeRADIUS server for external services or devices.

The object representing our Internal network and its NAT properties should be defined as well:



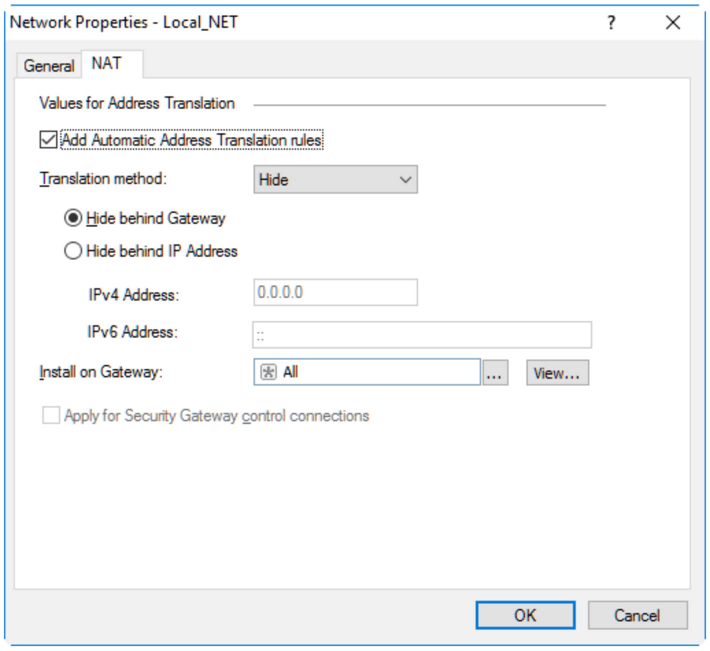
Network Properties - Local_NET

General | NAT

Name: Local_NET
Color: Black
Comment:

IPv4
Network Address: 192.168.7.0
Net Mask: 255.255.255.0
Broadcast address:
● Included
○ Not included

IPv6
Network Address: /



Network Properties - Local_NET

General | NAT

Values for Address Translation

☑ Add Automatic Address Translation rules

Translation method: Hide

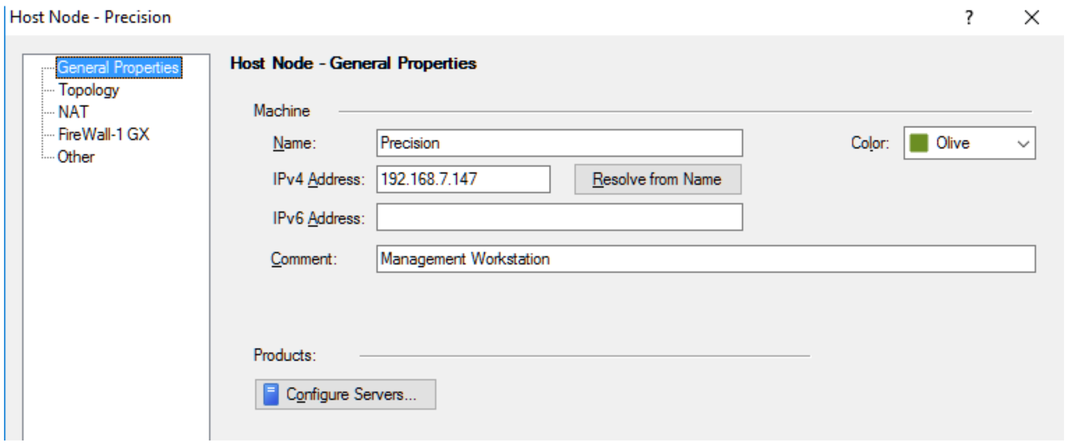
● Hide behind Gateway
○ Hide behind IP Address

IPv4 Address: 0.0.0.0
IPv6 Address: ::

Install on Gateway: All

☐ Apply for Security Gateway control connections

As well as the object representing our management workstation, (do not set NAT properties):



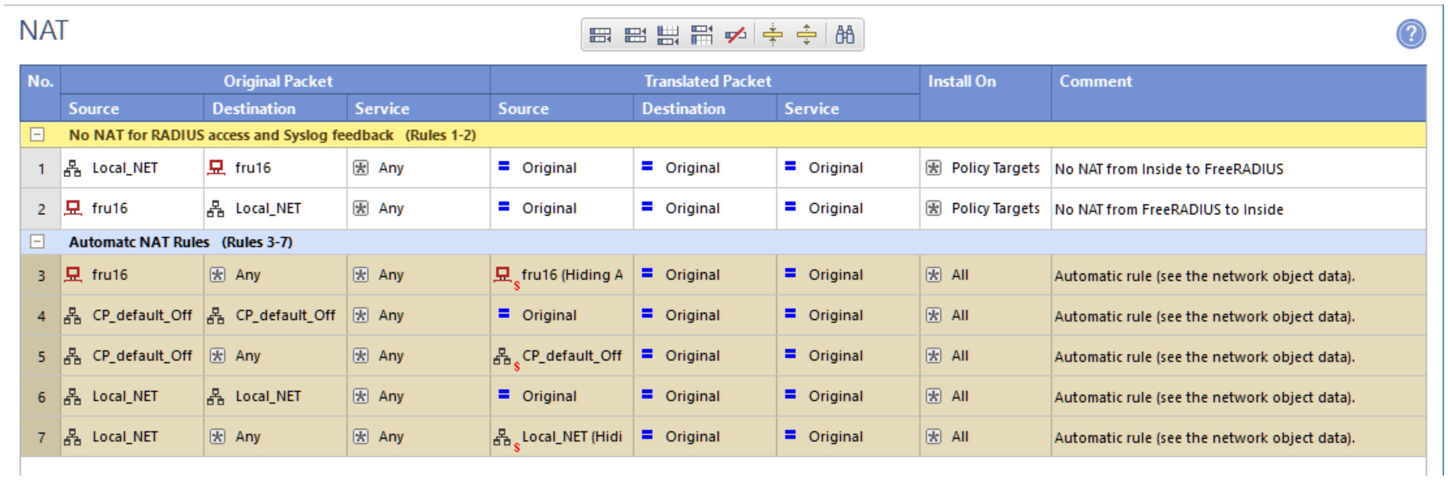
Host Node - General Properties

Machine

Name:	Precision
IPv4 Address:	192.168.7.147
IPv6 Address:	
Comment:	Management Workstation
Color:	Olive

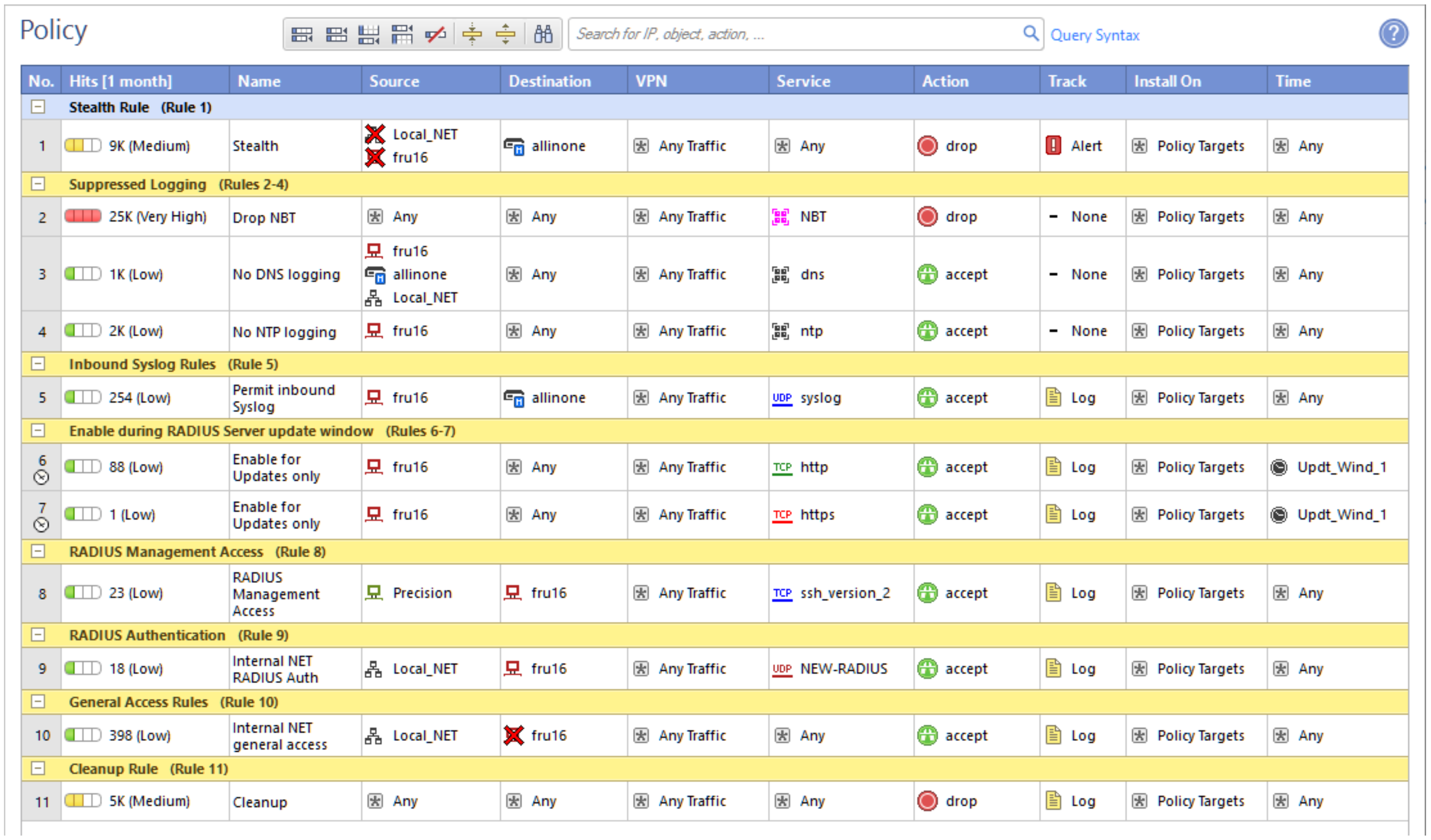
Products:

In SmartDashboard, “Firewall/NAT”, create the top two new rules shown here:



No.	Original Packet			Translated Packet			Install On	Comment
	Source	Destination	Service	Source	Destination	Service		
	No NAT for RADIUS access and Syslog feedback (Rules 1-2)							
1	Local_NET	fru16	Any	Original	Original	Original	Policy Targets	No NAT from Inside to FreeRADIUS
2	fru16	Local_NET	Any	Original	Original	Original	Policy Targets	No NAT from FreeRADIUS to Inside
	Automatc NAT Rules (Rules 3-7)							
3	fru16	Any	Any	fru16 (Hiding A	Original	Original	All	Automatic rule (see the network object data).
4	CP_default_Off	CP_default_Off	Any	Original	Original	Original	All	Automatic rule (see the network object data).
5	CP_default_Off	Any	Any	CP_default_Off	Original	Original	All	Automatic rule (see the network object data).
6	Local_NET	Local_NET	Any	Original	Original	Original	All	Automatic rule (see the network object data).
7	Local_NET	Any	Any	Local_NET (Hidi	Original	Original	All	Automatic rule (see the network object data).

And in “Firewall/Policy”, re-create the rules shown below, defining new “Time” object to limit your FreeRADIUS server’s ability to establish outbound connections to the Internet outside of its designated maintenance window:


Policy

Search for IP, object, action, ...

Query Syntax

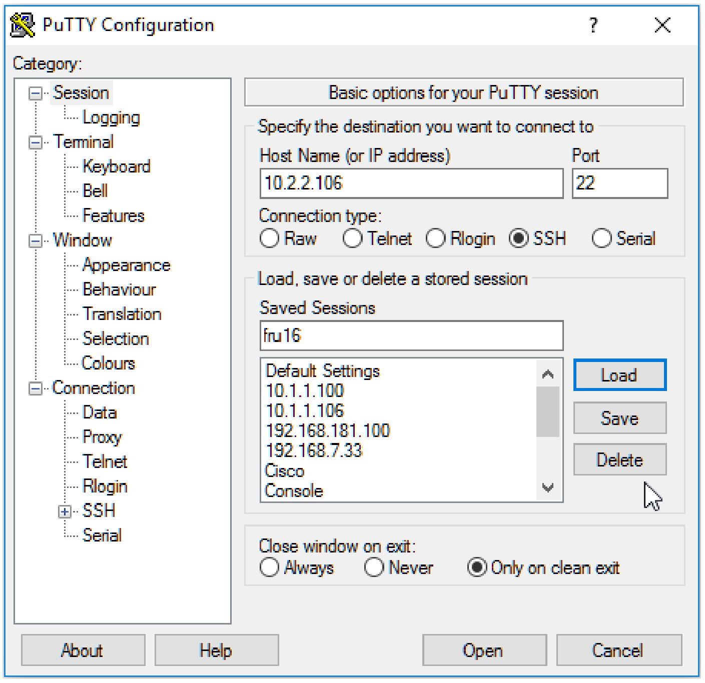
No.	Hits [1 month]	Name	Source	Destination	VPN	Service	Action	Track	Install On	Time
⊟	**Stealth Rule (Rule 1)**									
1	9K (Medium)	Stealth	Local_NET fru16	allinone	Any Traffic	Any	drop	Alert	Policy Targets	Any
⊟	**Suppressed Logging (Rules 2-4)**									
2	25K (Very High)	Drop NBT	Any	Any	Any Traffic	NBT	drop	- None	Policy Targets	Any
3	1K (Low)	No DNS logging	fru16 allinone Local_NET	Any	Any Traffic	dns	accept	- None	Policy Targets	Any
4	2K (Low)	No NTP logging	fru16	Any	Any Traffic	ntp	accept	- None	Policy Targets	Any
⊟	**Inbound Syslog Rules (Rule 5)**									
5	254 (Low)	Permit inbound Syslog	fru16	allinone	Any Traffic	UDP syslog	accept	Log	Policy Targets	Any
⊟	**Enable during RADIUS Server update window (Rules 6-7)**									
6	88 (Low)	Enable for Updates only	fru16	Any	Any Traffic	TCP http	accept	Log	Policy Targets	Updt_Wind_1
7	1 (Low)	Enable for Updates only	fru16	Any	Any Traffic	TCP https	accept	Log	Policy Targets	Updt_Wind_1
⊟	**RADIUS Management Access (Rule 8)**									
8	23 (Low)	RADIUS Management Access	Precision	fru16	Any Traffic	TCP ssh_version_2	accept	Log	Policy Targets	Any
⊟	**RADIUS Authentication (Rule 9)**									
9	18 (Low)	Internal NET RADIUS Auth	Local_NET	fru16	Any Traffic	UDP NEW-RADIUS	accept	Log	Policy Targets	Any
⊟	**General Access Rules (Rule 10)**									
10	398 (Low)	Internal NET general access	Local_NET	fru16	Any Traffic	Any	accept	Log	Policy Targets	Any
⊟	**Cleanup Rule (Rule 11)**									
11	5K (Medium)	Cleanup	Any	Any	Any Traffic	Any	drop	Log	Policy Targets	Any

During the initial configuration, you may enable logging for the “dns” and “ntp” traffic to verify its functionality. Once confirmed, it makes sense to set “Track” values to “- None” to avoid excessive logging.

The purpose of rule # 5 will become clear once we get to the “RADIUS Logging” section of this Guide.

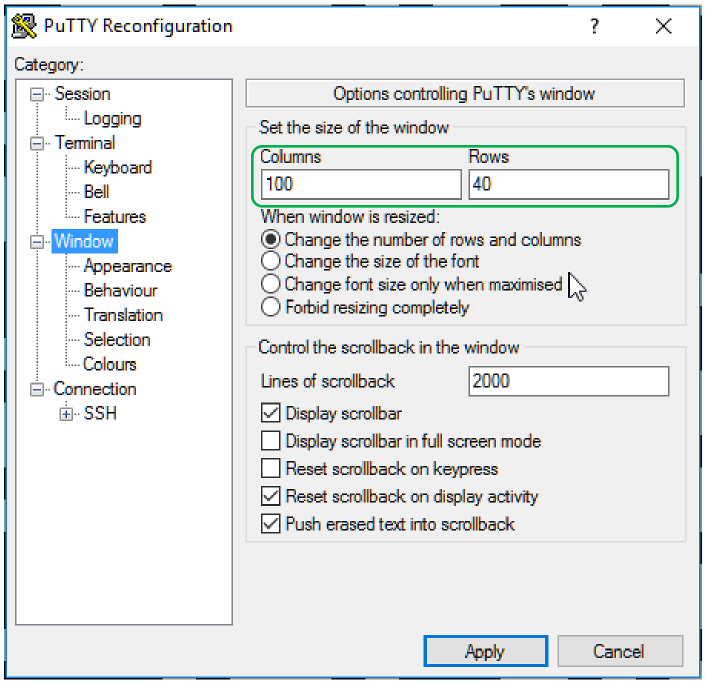
Configuring the Ubuntu Server

Once the server is rebooted, do not logon locally. Instead, use the SSH client residing on your Windows management PC, (typically PuTTY), to establish a connection to your Ubuntu server. Adjust "Window" parameters as shown below if your own are same as defaults. We will need the "Column" count exceeding default value to display uncorrupted QR codes; the Rows count is arbitrary, since we can adjust the height dynamically by resizing terminal window:


PuTTY Configuration

Category: Session (Logging), Terminal (Keyboard, Bell, Features), Window (Appearance, Behaviour, Translation, Selection, Colours), Connection (Data, Proxy, Telnet, Rlogin, SSH, Serial)

Basic options for your PuTTY session
Specify the destination you want to connect to
Host Name (or IP address): 10.2.2.106
Port: 22
Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial
Load, save or delete a stored session
Saved Sessions: fru16
Default Settings
10.1.1.100
10.1.1.106
192.168.181.100
192.168.7.33
Cisco
Console
Close window on exit: ○ Always ○ Never ● Only on clean exit


PuTTY Reconfiguration

Category: Session (Logging), Terminal (Keyboard, Bell, Features), Window (Appearance, Behaviour, Translation, Selection, Colours), Connection (SSH)

Options controlling PuTTY's window
Set the size of the window
Columns: 100
Rows: 40
When window is resized:
● Change the number of rows and columns
○ Change the size of the font
○ Change font size only when maximised
○ Forbid resizing completely
Control the scrollback in the window
Lines of scrollback: 2000
☑ Display scrollbar
☐ Display scrollbar in full screen mode
☐ Reset scrollback on keypress
☑ Reset scrollback on display activity
☑ Push erased text into scrollback

Click "Yes" when prompted with "PuTTY Security Alert" and you should have established an SSH session.

You can now perform remote administration as well as copy/paste operations.

You must elevate your privileges to continue installation:

```
fradmin@fru16:~$ sudo su
[sudo] password for fradmin: "Password assigned to fradmin"
root@fru16:/home/fradmin#
```

to get to the root of your directory (there is a space between **cd** and **~**):

```
root@fru16:/home/fradmin# cd ~
root@fru16:~#
```

Download the package lists from the repositories and update them to get information on the newest versions of packages and their dependencies:

```
sudo apt-get update
```

Install NTP modules, as we will rely on time-based tokens, (you may forego this step if, for example, you are running Ubuntu on a hypervisor and it is syncing time with the host):

```
sudo apt-get install ntp
```

When prompted:

Do you want to continue? [Y/n], press **y** and **Enter**

Check the default time zone and change it to the one you are in, (press **Enter** after each blue line):

```
root@fru16:/home/fradmin# date
Wed Nov  9 15:08:27 PST 2016
root@fru16:/home/fradmin# cd ~
root@fru16:~# timedatectl set-timezone America/New_York
root@fru16:~# timedatectl
      Local time: Wed 2016-11-09 18:11:14 EST
  Universal time: Wed 2016-11-09 23:11:14 UTC
        RTC time: Wed 2016-11-09 23:11:16
       Time zone: America/New_York (EST, -0500)
 Network time on: yes
NTP synchronized: no
 RTC in local TZ: no
root@fru16:~# service ntp restart
root@fru16:~# date
Wed Nov  9 18:11:30 EST 2016
root@fru16:~#
reboot
```

If you do not know the correct definition of your Time Zone, perform search, (I am using Tahiti as example):

```
fradmin@fru16:~$ timedatectl list-timezones | grep Tahiti
Pacific/Tahiti
```

Then use the complete output of the search in your zone definition.

Run the update again, to bring all the information about dependencies in line with freshly installed components.

Execute:

```
sudo apt-get update
```

Installing FreeRADIUS, QR Generator and Supplemental Packages

Now install FreeRADIUS, a widely used open source RADIUS server:

Execute:

```
sudo apt-get install build-essential libpam0g-dev freeradius git libqrencode3
```

When prompted:

```
 Do you want to continue? [Y/n]:
```

press **y** and press **Enter**

After a few seconds, (~ 30, depending on available bandwidth), FreeRADIUS is installed.

Installing Google Authenticator

Paste this line to the terminal and press **Enter** to install the Google Authenticator components:

```
sudo apt-get install libpam-google-authenticator
```

Configuring FreeRADIUS for MFA with Google Authenticator

Create a group for users whose access you may want to temporary disable in the future without deleting accounts:

```
sudo addgroup radius-disabled
```

Edit configuration files:

```
sudo nano /etc/freeradius/radiusd.conf
```

Find lines:

```
user = freerad
group = freerad
```

(Use **Ctrl + w** to search in Nano)

Comment both lines out, copy and paste these lines below:

```
user = root
group = root
```

Press **Ctrl+x**; **y** and **Enter** to save changes and exit Nano.

Execute:

```
sudo nano /etc/freeradius/users
```

Look for this block of text:

```
#   Deny access for a group of users.
#
#   Note that there is NO 'Fall-Through' attribute, so the user will not
#   be given any additional resources.
```

And paste these three lines under it:

```
DEFAULT         Group == "radius-disabled", Auth-Type := Reject
                Reply-Message = "Your account has been disabled."
DEFAULT        Auth-Type := PAM
```

Press **Ctrl+x**; **y** and **Enter** to save changes and exit Nano.

Execute:

```
sudo nano /etc/freeradius/sites-enabled/default
```

Search for this block of text containing commented-out word "pam". Insert uncommented line "pam" under it:

```
#  Pluggable Authentication Modules.
#          pam
           pam
```

Press **Ctrl+x**; **y** and **Enter** to save changes and exit Nano.

Configure PAM to use a combination of the local user's password and the PIN generated by Google Authenticator:

Execute:

```
sudo nano /etc/pam.d/radiusd
```

Find and comment-out these four lines by prepending them with "#":

```
@include common-auth
@include common-account
@include common-password
@include common-session
```

Paste these two lines under the four lines commented out in the step above:

```
auth requisite pam_google_authenticator.so forward_pass
auth required pam_unix.so use_first_pass
```

Your file should now look like this:

```
# /etc/pam.d/radiusd - PAM configuration for FreeRADIUS
#
# We fall back to the system default in /etc/pam.d/common-*
#
#@include common-auth
#@include common-account
#@include common-password
#@include common-session
auth requisite pam_google_authenticator.so forward_pass
auth required pam_unix.so use_first_pass
```

Press **Ctrl+x**; **y** and **Enter** to save changes and exit Nano.

Execute:

```
sudo nano /etc/freeradius/radiusd.conf
```

Ctrl+ w; find “auth”, comment it out and add line auth = yes below:

```
#auth = no
auth = yes
        #  Log passwords with the authentication requests.
        #  auth_badpass  - logs password if it's rejected
        #  auth_goodpass - logs password if it's correct
        #
        #  allowed values: {no, yes}
        #
        auth_badpass = no
        auth_goodpass = no
```

Press **Ctrl +x**; **y** and **Enter** to save changes and exit Nano.

```
sudo reboot
```

You are now ready to set up clients and users, issue google-authenticator tokens and perform RADIUS authentication.

Configuring Clients on FreeRADIUS

Clients are the hosts that authenticate Users against RADIUS.

Example below shows how to configure two clients, one being our Check Point firewall's DMZ IP address facing RADIUS server, the other - our management workstation. Generally, the workstation would not be a client, but having it defined as such will allow us to conduct tests against RADIUS directly and to confirm its functionality in advance.

Execute:

```
sudo nano /etc/freeradius/clients.conf
```

Scroll down with the arrow keys and append the file with section containing your Check Point management server, as well as your management workstation. Replace IPs and "secret" with those pertinent to your environment. "Secret" is an authentication method that the Clients will be using when addressing RADIUS and corresponding entries will be defined later in your firewall and your workstation. "Shortname" is simply a name under which RADIUS will be logging your Client's activity.

```
client 10.2.2.1 {
       secret          = SECRET
       shortname       = allinoneDMZ
}
#
client 192.168.7.147 {
       secret          = SECRET
       shortname       = Precision
}
```

Press **Ctrl+x**; **y** and **Enter** to save changes and exit Nano.

Configuring Users on FreeRADIUS

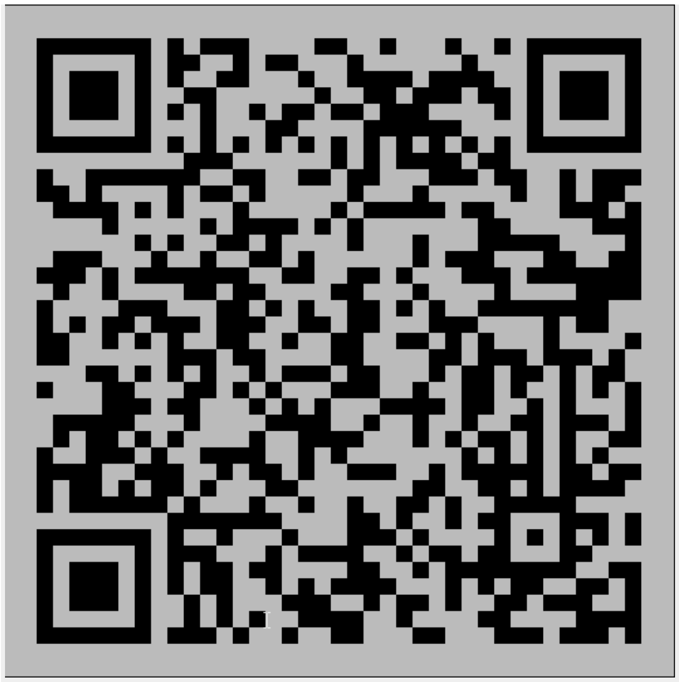
The process of adding a user consists of the following steps:

1. Add user
2. Navigate to user's home directory
3. Change the ownership of a login session to the one you have created
4. Create google-authenticator token
5. Set google-authenticator parameters
6. Register google-authenticator using QR or secret code
7. Exit user's session
8. Restart FreeRADIUS process

Execute commands depicted in blue below:

```
root@fru16:~# adduser cpmonitor
Adding user `cpmonitor' ...
Adding new group `cpmonitor' (1004) ...
Adding new user `cpmonitor' (1003) with group `cpmonitor' ...
Creating home directory `/home/cpmonitor' ...
Copying files from `/etc/skel' ...
Enter new UNIX password: new user's password
Retype new UNIX password: new user's password
passwd: password updated successfully
Changing the user information for cpmonitor
Enter the new value, or press ENTER for the default
        Full Name []:CheckPoint Monitor
        Room Number []: 123
        Work Phone []: +1-212-555-5555
        Home Phone []: +1-973-999-9999
        Other []: CheckPoint trainee account
Is the information correct? [Y/n] Y
root@fru16~# cd /home/cpmonitor
root@fru16/home/cpmonitor# su cpmonitor
cpmonitor@fru16:~$ google-authenticator
Do you want authentication tokens to be time-based (y/n) y

https://www.google.com/chart?chs=200x200&chld=M|0&cht=qr&chl=otpauth://totp/cpmon
itor@fru16%3Fsecret%3DZLVQMR7ZTCRP24LZGRL3WQOWRQ%26issuer%3Dubuntu
```



Note: Actual QR code will take up a square larger than 80 characters wide and long, so you may have to resize the terminal window to accommodate its size.

If you are seeing a corrupt QR code, i.e. one comprised of dashes, it is likely that you have not defined terminal window width "100" on Page 10.

If this is the case, simply press "**n**" to cancel the operation, resize terminal window and rerun "google-authenticator" command.

```
Your new secret key is: ZLVQMR7ZTCRP24LZGRL3WQOWRQ
Your verification code is 443998
Your emergency scratch codes are:
  29294009
  71196238
  77206028
  16714754
  77329657
Do you want me to update your "/home/cpmonitor/.google_authenticator" file (y/n)y

Do you want to disallow multiple uses of the same authentication
token? This restricts you to one login about every 30s, but it increases
your chances to notice or even prevent man-in-the-middle attacks (y/n) Do you
want to disallow multiple uses of the same authentication
token? This restricts you to one login about every 30s, but it increases
your chances to notice or even prevent man-in-the-middle attacks (y/n)  y

By default, tokens are good for 30 seconds. In order to compensate for
possible time-skew between the client and the server, we allow an extra
token before and after the current time. If you experience problems with
poor time synchronization, you can increase the window from its default
size of +-1min (window size of 3) to about +-4min (window size of
17 acceptable tokens).
Do you want to do so? (y/n)  n

If the computer that you are logging into isn't hardened against brute-force
login attempts, you can enable rate-limiting for the authentication module.
```

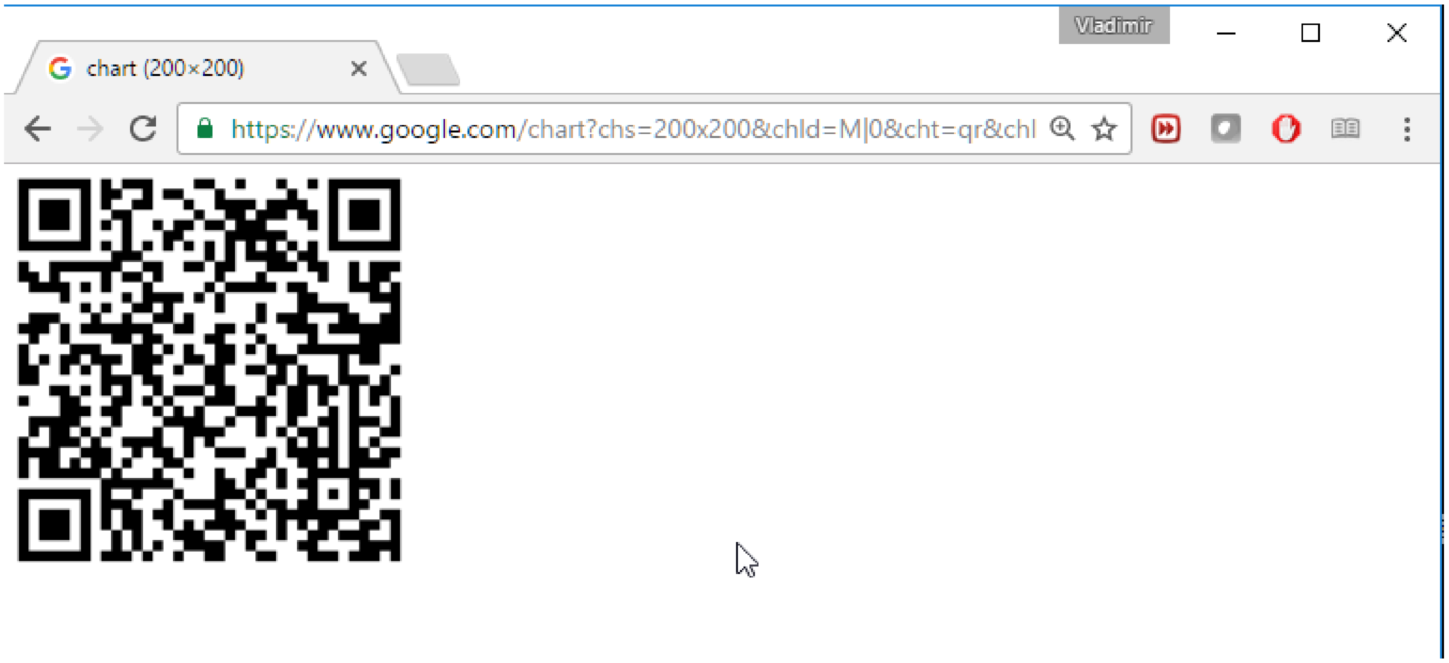
```
By default, this limits attackers to no more than 3 login attempts every 30s.
Do you want to enable rate-limiting (y/n) y
cpmonitor@fru16:~$ exit
exit
root@fru16/home/cpmonitor# cd ~
root@fru16~# service freeradius restart
```

The QR code will remain visible only for the duration of the session. If you would like to provide your users with the remote access to the QR code, email them the URL located immediately above the QR code.

When pasted in the browser,

https://www.google.com/chart?chs=200x200&chld=M|0&cht=qr&chl=otpauth://totp/cpmonitor@fru%3Fsecret%3DZLVQMR7ZTCRP24LZGRL3WQOWRQ%26issuer%3Dubuntu

will generate identical QR code at google.com:


Vladimir
chart (200×200)
https://www.google.com/chart?chs=200x200&chld=M|0&cht=qr&chl

Alternatively, you can email your users their secret key together with the emergency scratch codes located under the QR code during the original session, or located later in the /home/username/.google-authenticator file:

```
root@fru16~# cd /home/cpmonitor
root@fru16/home/cpmonitor# su cpmonitor
cpmonitor@fru16:~$ ls -a
.  ..  .bash_history  .bash_logout  .bashrc  .google_authenticator  .profile
cpmonitor@fru16:~$ cat .google_authenticator
ZLVQMR7ZTCRP24LZGRL3WQOWRQ
" RATE_LIMIT 3 30 1478891731
" DISALLOW_REUSE 49296382
" TOTP_AUTH
29294009
```

```
71196238
77206028
16714754
77329657
cpmonitor@fru16:~$
```

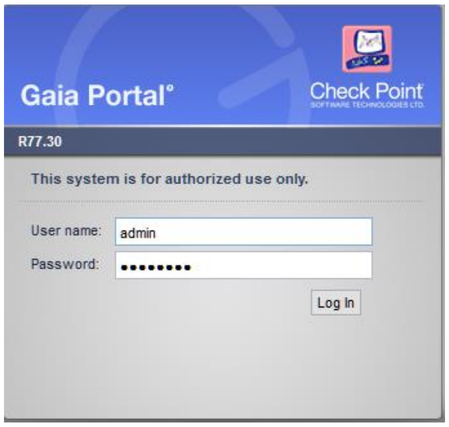
The secret key could be copied and pasted to their Google Authenticator app manually. Note that the account name in the Google Authenticator app could be arbitrary if the secret key is entered manually. It will not affect your ability to authenticate, as long as you are entering a proper user name at the logon prompt. If you are scanning the QR code, user name will be in the format of user@freeradiushostname. Do not enter the “@freeradiushostname” at Check Point Web UI or SSH logon prompts. Check Point does not support special characters in user names.

Test the ability of the user account created to authenticate locally, replacing “<passwd>” with the actual password you have assigned to the user, and “<g-auth-code> with the 6-digit code generated by the Google Authenticator. Port 18120 is not specified in error. This is a pre-defined port for local authentication, as is “testing123” a pre-defined secret for the local host:

```
root@fru16:~# radtest cpmonitor <passwd><g-auth-code> localhost 18120 testing123
Sending Access-Request of id 104 to 127.0.0.1 port 1812
        User-Name = "cpmonitor"
        User-Password = "<unix_password>493895"
        NAS-IP-Address = 10.2.2.106
        NAS-Port = 18120
        Message-Authenticator = 0x00000000000000000000000000000000
rad_recv: Access-Accept packet from host 127.0.0.1 port 1812, id=104, length=20
root@fru16:~#
```

Setting up Local RADIUS Users for Access to Gaia Web UI and Clish

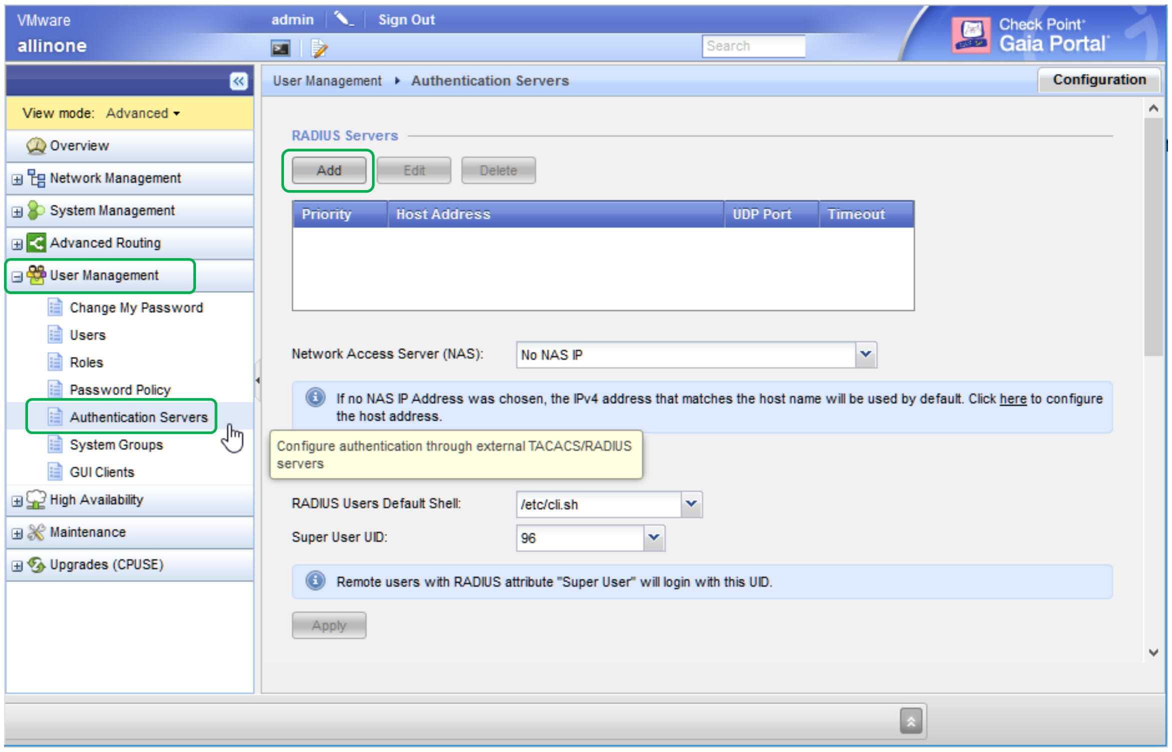
To configure users in Web UI, logon to Gaia Web Portal using administrative credentials defined during installation of the management server or gateway, or those defined via “cpconfig”:



This system is for authorized use only.

User name: admin
Password: ••••••••

Navigate to “User Management” section and click on the “Authentication Servers”. In the right pane on top, click on the “Add” button, below the “RADIUS Servers”:



Priority	Host Address	UDP Port	Timeout

Network Access Server (NAS): No NAS IP

If no NAS IP Address was chosen, the IPv4 address that matches the host name will be used by default. Click here to configure the host address.

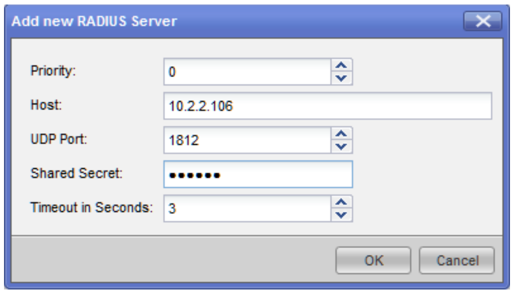
Configure authentication through external TACACS/RADIUS servers

RADIUS Users Default Shell: /etc/cli.sh
Super User UID: 96

Remote users with RADIUS attribute "Super User" will login with this UID.

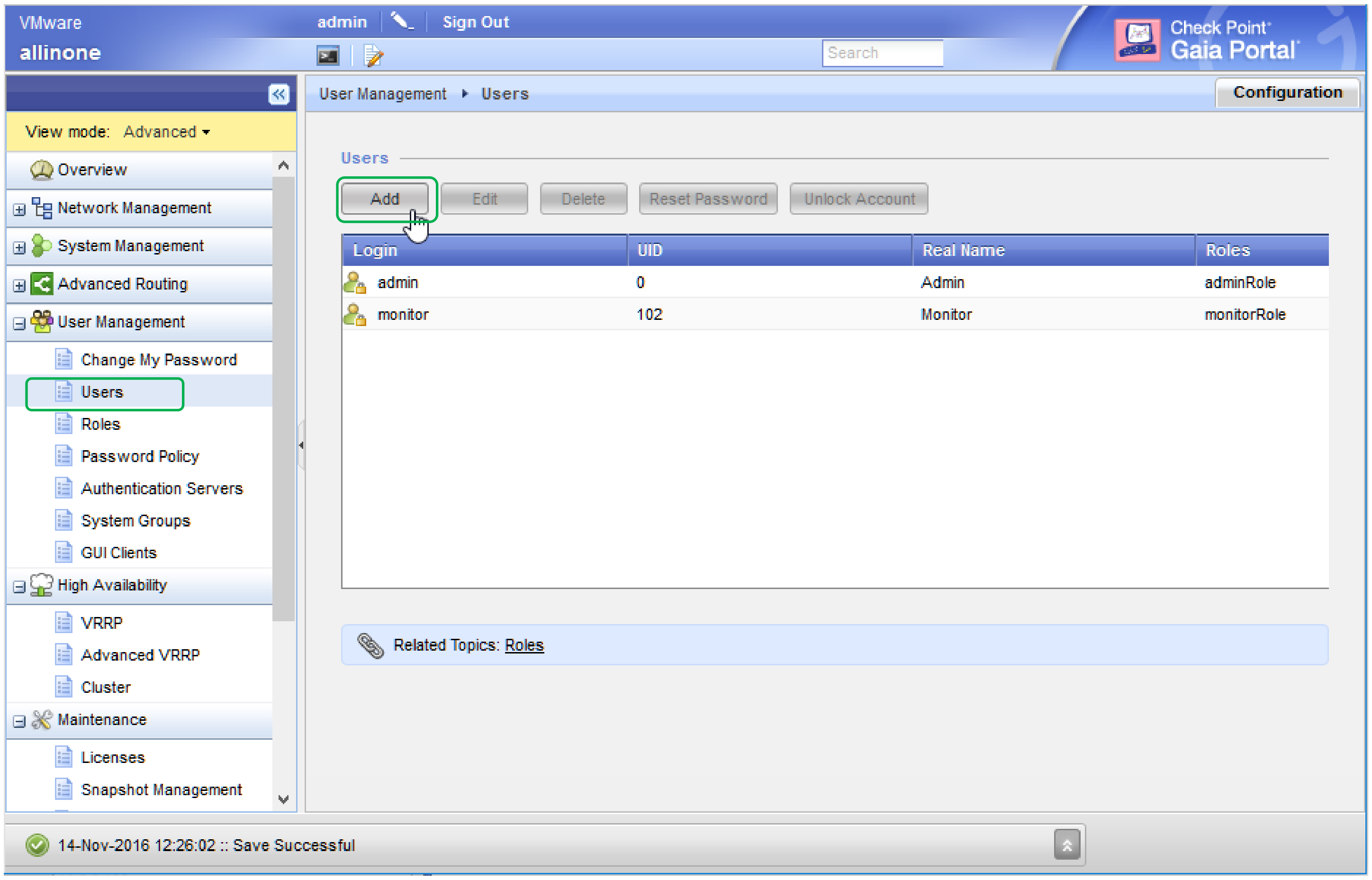
Fill out "Host:" with the IP address of your RADIUS server and the "Secret:" you have defined for it in the "Configuring Clients" section.

Click "Ok":


Add new RADIUS Server

Priority:	0
Host:	10.2.2.106
UDP Port:	1812
Shared Secret:	••••••
Timeout in Seconds:	3

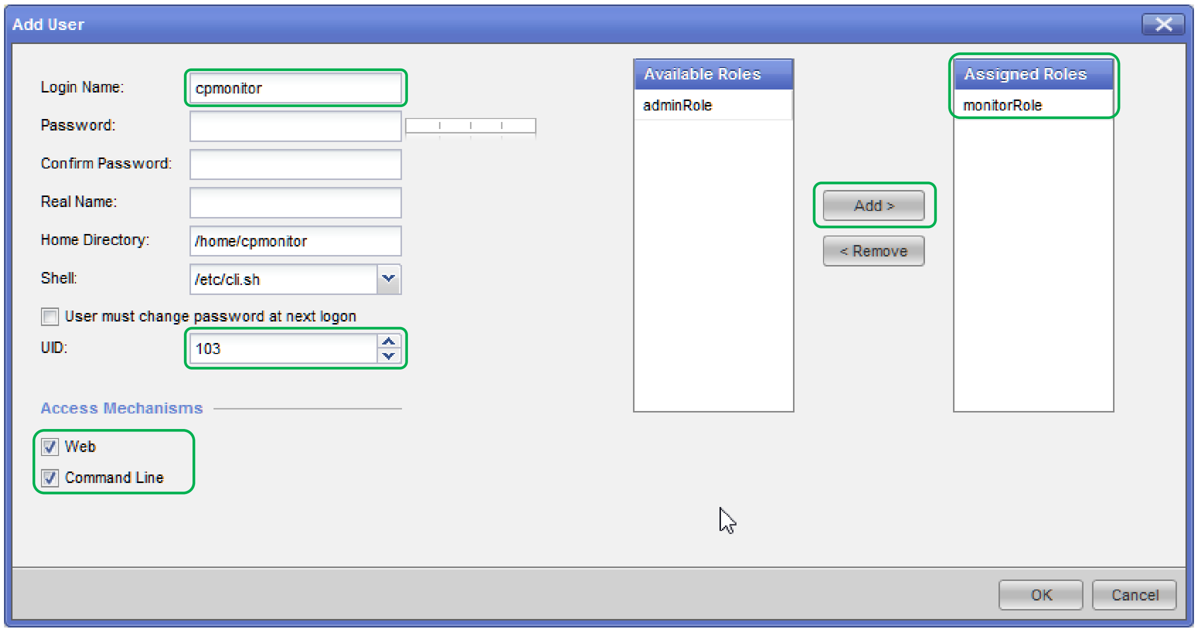
Click on "Users" and, in the right pane on top, click on "Add":



Login	UID	Real Name	Roles
admin	0	Admin	adminRole
monitor	102	Monitor	monitorRole

14-Nov-2016 12:26:02 :: Save Successful

Fill out User's "Login Name", spelling it exactly as you have done in the "Creating Users in RADIUS" section. Specify desired "Access Mechanisms", select preferred role in "Available Roles" and click "Add >" to move the selected role to the "Added Roles" section. Leave the "Password" field blank and click "Ok":



Add User

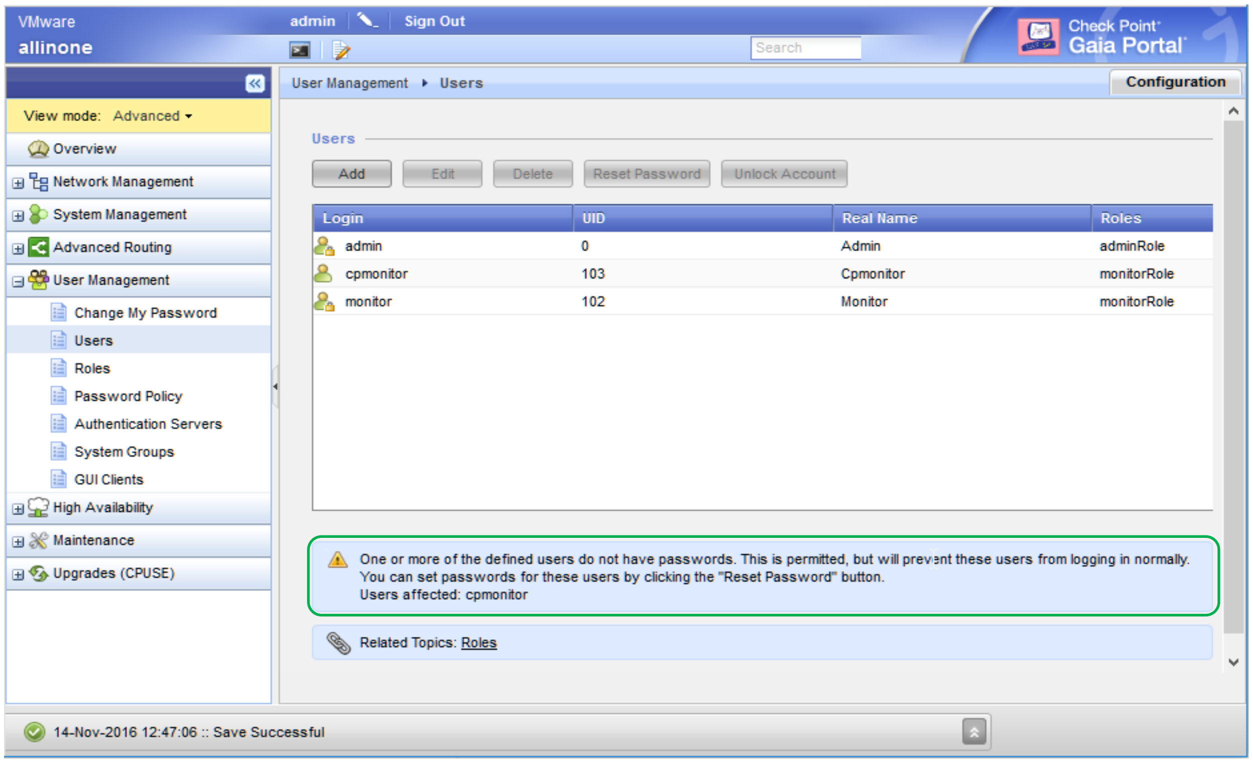
Login Name:	cpmonitor
Password:	
Confirm Password:	
Real Name:	
Home Directory:	/home/cpmonitor
Shell:	/etc/cli.sh
User must change password at next logon	☐
UID:	103

Access Mechanisms

- [✓] Web
- [✓] Command Line

Available Roles: adminRole

Assigned Roles: monitorRole



Users

Login	UID	Real Name	Roles
admin	0	Admin	adminRole
cpmonitor	103	Cpmonitor	monitorRole
monitor	102	Monitor	monitorRole

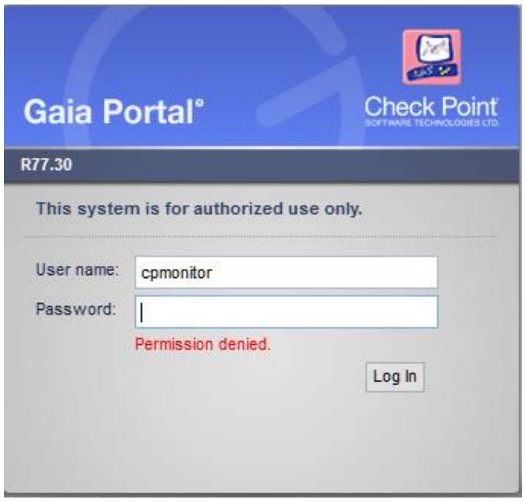
One or more of the defined users do not have passwords. This is permitted, but will prevent these users from logging in normally.
You can set passwords for these users by clicking the "Reset Password" button.
Users affected: cpmonitor

Related Topics: Roles

14-Nov-2016 12:47:06 :: Save Successful

Note the warning about users that do not have passwords assigned.

Users with no passwords assigned are required to supply credentials for their account created in the Authentication Servers defined earlier. Attempts to login without password will fail with “Permission Denied” notification:

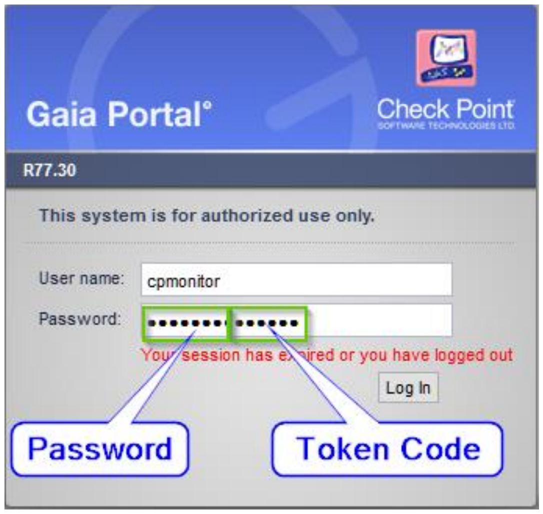


This system is for authorized use only.

User name: cpmonitor
Password:

Permission denied.

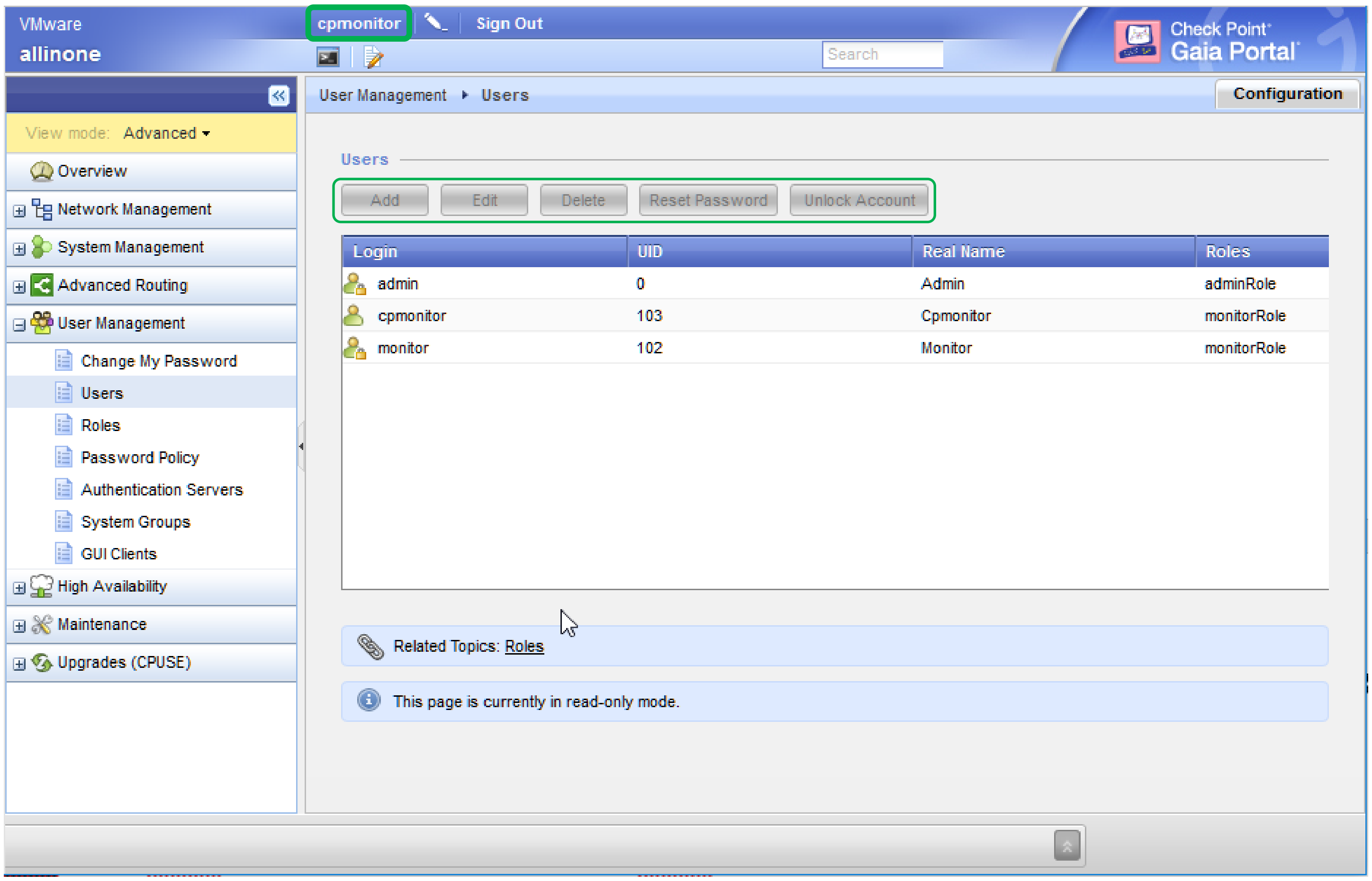
When the correct password defined on our RADIUS server is entered in combination with the code generated by Google Authenticator, logon succeeds and the permissions defined in the Role assigned to the user in Gaia are applied:



This system is for authorized use only.

User name: cpmonitor
Password: •••••••• ••••••

Your session has expired or you have logged out



Login	UID	Real Name	Roles
admin	0	Admin	adminRole
cpmonitor	103	Cpmonitor	monitorRole
monitor	102	Monitor	monitorRole

Related Topics: Roles

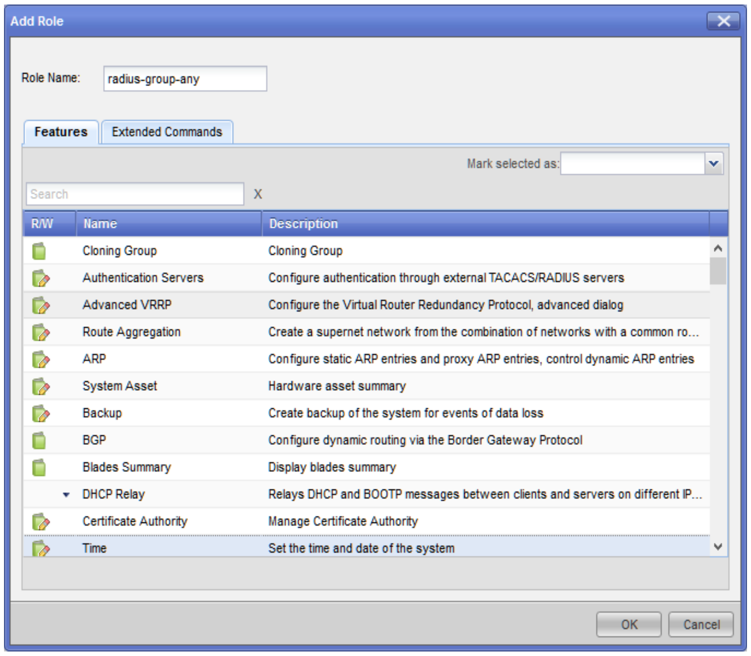
This page is currently in read-only mode.

In the picture above, user “cpmonitor” that was assigned “monitorRole” has all the action buttons grayed-out.

SSH access to Gaia for locally defined RADIUS users* will work in the same exact fashion, permitting only the execution of commands defined in the Role assigned to the user.

*Locally defined RADIUS users, in the context of this document, are the users defined on each Check Point device.

Setting up Non-Local RADIUS Users for Web UI and CLI


Add Role

Role Name: radius-group-any

Features | Extended Commands

Mark selected as:

R/W	Name	Description
	Cloning Group	Cloning Group
	Authentication Servers	Configure authentication through external TACACS/RADIUS servers
	Advanced VRRP	Configure the Virtual Router Redundancy Protocol, advanced dialog
	Route Aggregation	Create a supernet network from the combination of networks with a common ro...
	ARP	Configure static ARP entries and proxy ARP entries, control dynamic ARP entries
	System Asset	Hardware asset summary
	Backup	Create backup of the system for events of data loss
	BGP	Configure dynamic routing via the Border Gateway Protocol
	Blades Summary	Display blades summary
	DHCP Relay	Relays DHCP and BOOTP messages between clients and servers on different IP...
	Certificate Authority	Manage Certificate Authority
	Time	Set the time and date of the system

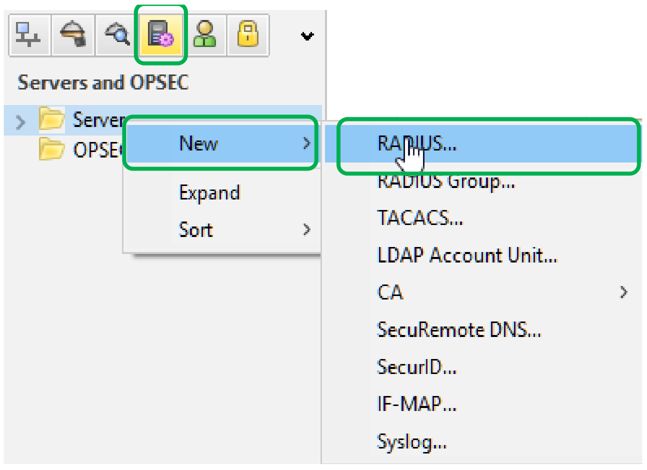
You should be able to logon to CLI and Web UI with users created solely on RADIUS server. This could be achieved by integrating library.checkpoint into your radius server, creating a role “radius-group-any” and assigning it the features and the Extended Commands you would like your users to execute (sk72940; sk93309).

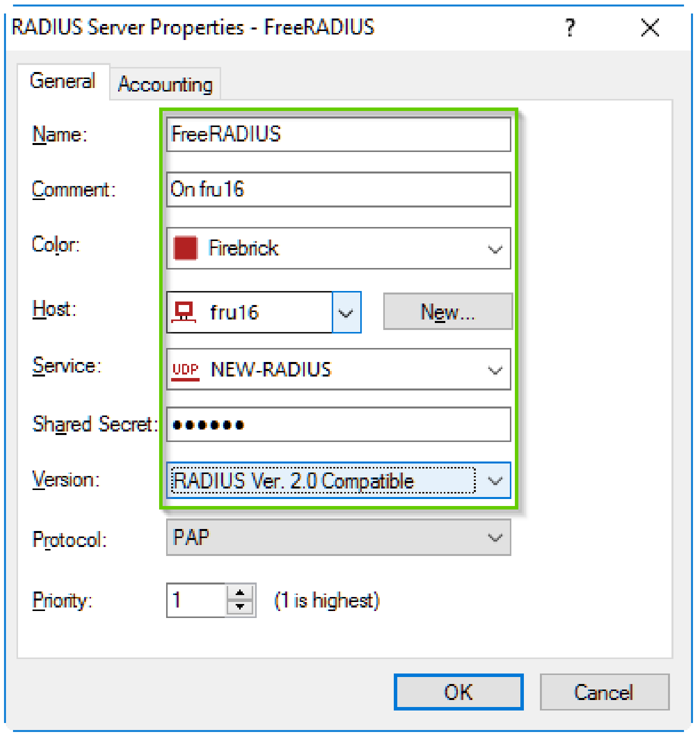
It works well for Web UI, but for CLI there is a bug that Check Point claims was resolved in sk102113, but that in fact persists. You can observe the bug by logging in with RADIUS only account, getting into the “Expert” mode and executing “whoami”. If the bug is present, command returns “_nonlocal”.

Configuration for Access to Smart Console Applications

RADIUS object configuration:

In “Host:” field, specify object defined above; in “Service:” choose “NEW-RADIUS”; specify same secret as we have defined for the firewall object in section “Configuring Clients”; “Version:” should be “RADIUS Ver. 2.0 Compatible” and click “Ok”:

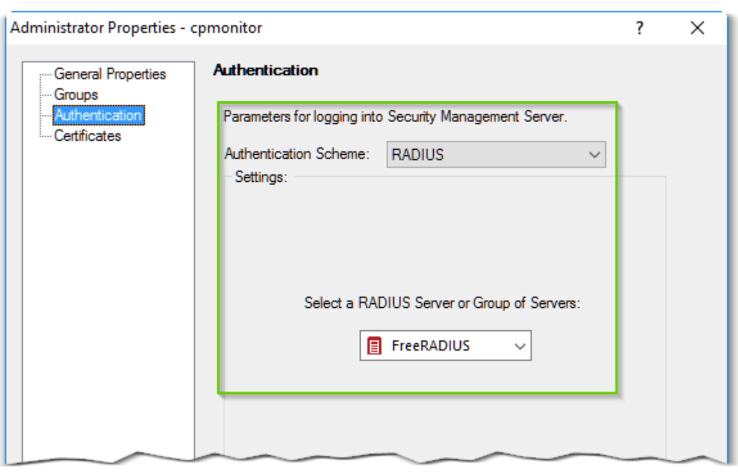



RADIUS Server Properties - FreeRADIUS

General | Accounting

Name: FreeRADIUS
Comment: On fru16
Color: Firebrick
Host: fru16
Service: NEW-RADIUS
Shared Secret: ••••••
Version: RADIUS Ver. 2.0 Compatible
Protocol: PAP
Priority: 1 (1 is highest)

Configuring administrator’s authentication method and selecting RADIUS server or group responsible for it:


Administrator Properties - cpmonitor

Authentication

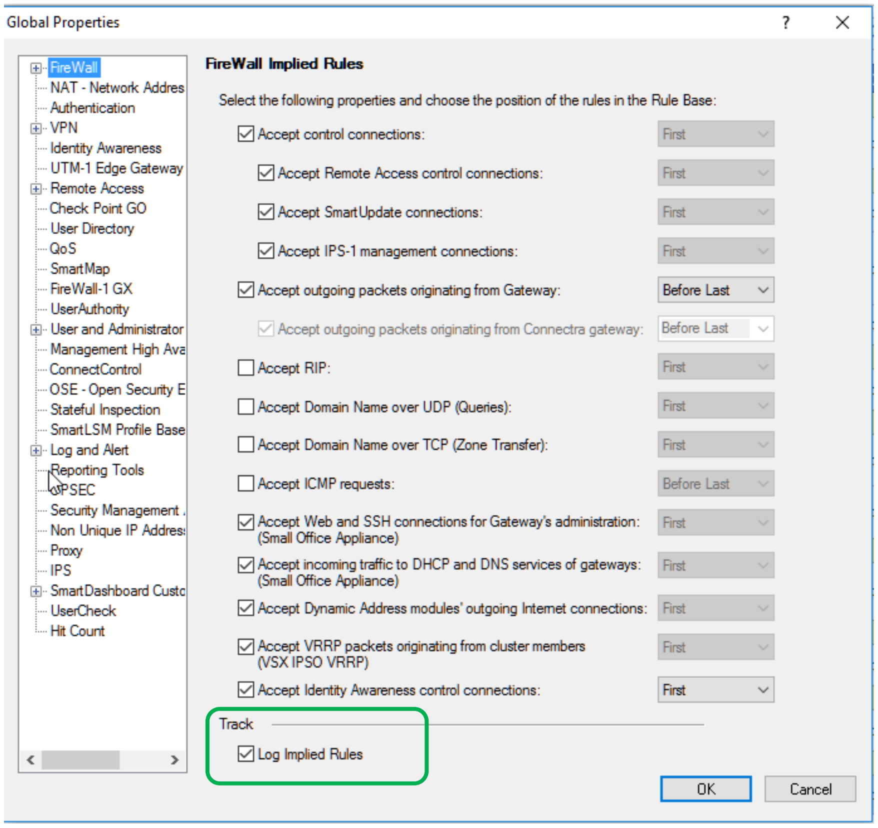
Parameters for logging into Security Management Server.

Authentication Scheme: RADIUS

Settings:

Select a RADIUS Server or Group of Servers: FreeRADIUS

Once the objects are created and the authentication method is defined, load the policy.



Global Properties

FireWall	NAT - Network Addres
Authentication	VPN
Identity Awareness	UTM-1 Edge Gateway
Remote Access	Check Point GO
User Directory	QoS
SmartMap	FireWall-1 GX
UserAuthority	User and Administrator
Management High Ava	ConnectControl
OSE - Open Security E	Stateful Inspection
SmartLSM Profile Base	Log and Alert
Reporting Tools	IPSEC
Security Management	Non Unique IP Addres
Proxy	IPS
SmartDashboard Custo	UserCheck
Hit Count	

FireWall Implied Rules

Select the following properties and choose the position of the rules in the Rule Base:

☑ Accept control connections:	First
☑ Accept Remote Access control connections:	First
☑ Accept SmartUpdate connections:	First
☑ Accept IPS-1 management connections:	First
☑ Accept outgoing packets originating from Gateway:	Before Last
☑ Accept outgoing packets originating from Connectra gateway:	Before Last
☐ Accept RIP:	First
☐ Accept Domain Name over UDP (Queries):	First
☐ Accept Domain Name over TCP (Zone Transfer):	First
☐ Accept ICMP requests:	Before Last
☑ Accept Web and SSH connections for Gateway's administration: (Small Office Appliance)	First
☑ Accept incoming traffic to DHCP and DNS services of gateways: (Small Office Appliance)	First
☑ Accept Dynamic Address modules' outgoing Internet connections:	First
☑ Accept VRRP packets originating from cluster members (VSX IPSO VRRP)	First
☑ Accept Identity Awareness control connections:	First

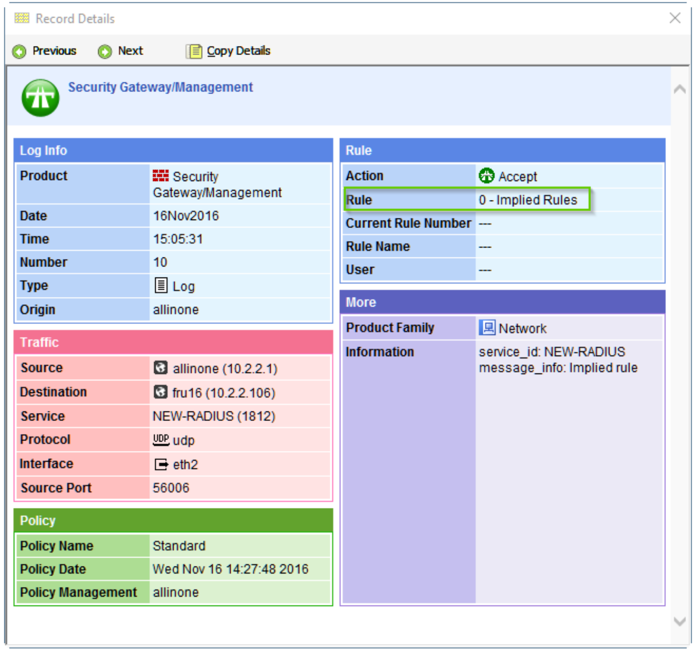
Track

☑ Log Implied Rules

RADIUS authentication is being processed by implied rules.

If "Log Implied Rules" is not enabled in the "Policy\Global Properties", you will not see the authentication attempts in the SmartView Tracker.

If the "Log Implied Rules" settings are enabled, during the RADIUS authentication attempts you will see traces like this:



Security Gateway/Management

Log Info	
Product	Security Gateway/Management
Date	16Nov2016
Time	15:05:31
Number	10
Type	Log
Origin	allinone

Traffic	
Source	allinone (10.2.2.1)
Destination	fru16 (10.2.2.106)
Service	NEW-RADIUS (1812)
Protocol	udp
Interface	eth2
Source Port	56006

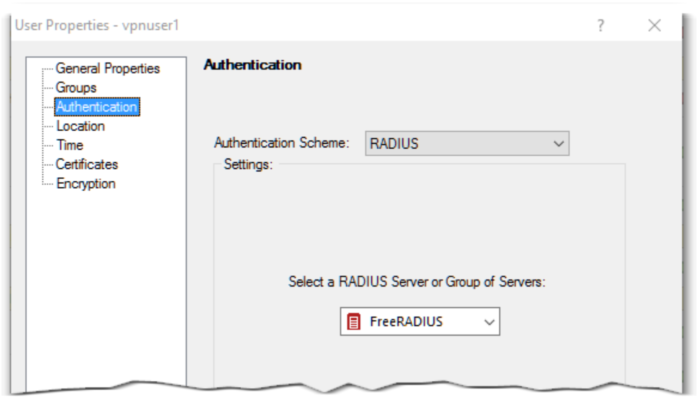
Policy	
Policy Name	Standard
Policy Date	Wed Nov 16 14:27:48 2016
Policy Management	allinone

Rule	
Action	Accept
Rule	0 - Implied Rules
Current Rule Number	---
Rule Name	---
User	---

More	
Product Family	Network
Information	service_id: NEW-RADIUS message_info: Implied rule

Configuring RADIUS Authentication for VPN Users

In addition to individual user's authentication that is similar to that described for Smart Console administrators:



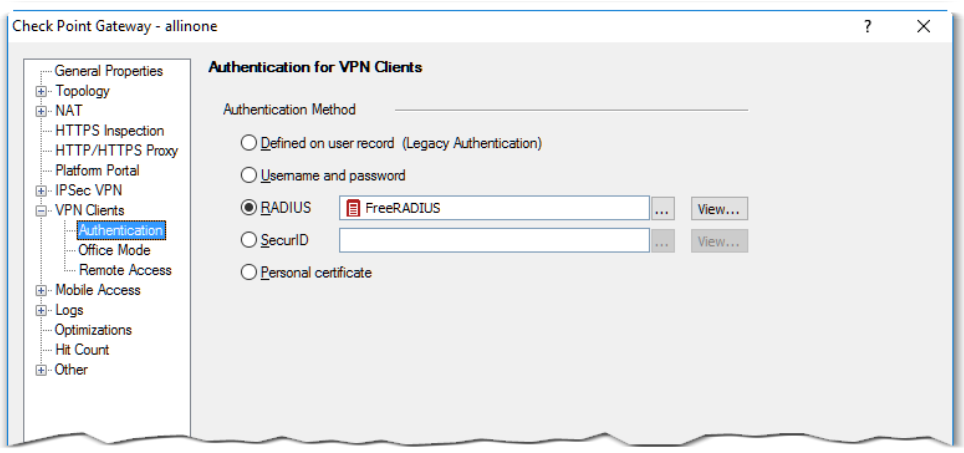
Authentication

Authentication Scheme: RADIUS

Settings:

Select a RADIUS Server or Group of Servers: FreeRADIUS

RADIUS authentication could be enabled by default in the "VPN Clients/Authentication":

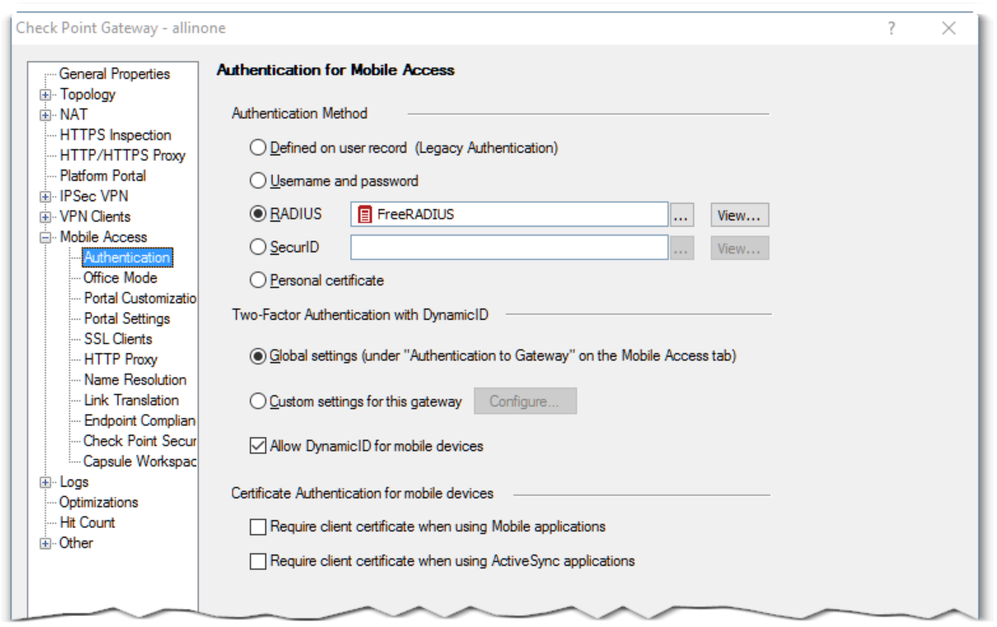


Authentication for VPN Clients

Authentication Method

( ) Defined on user record (Legacy Authentication)
( ) Username and password
[✓ RADIUS] FreeRADIUS
( ) SecurID
( ) Personal certificate

and the “Mobile Access/Authentication” properties of the Gateway:


Check Point Gateway - allinone

Authentication for Mobile Access

Authentication Method

- ○ Defined on user record (Legacy Authentication)
- ○ Username and password
- ● RADIUS: FreeRADIUS
- ○ SecurID
- ○ Personal certificate

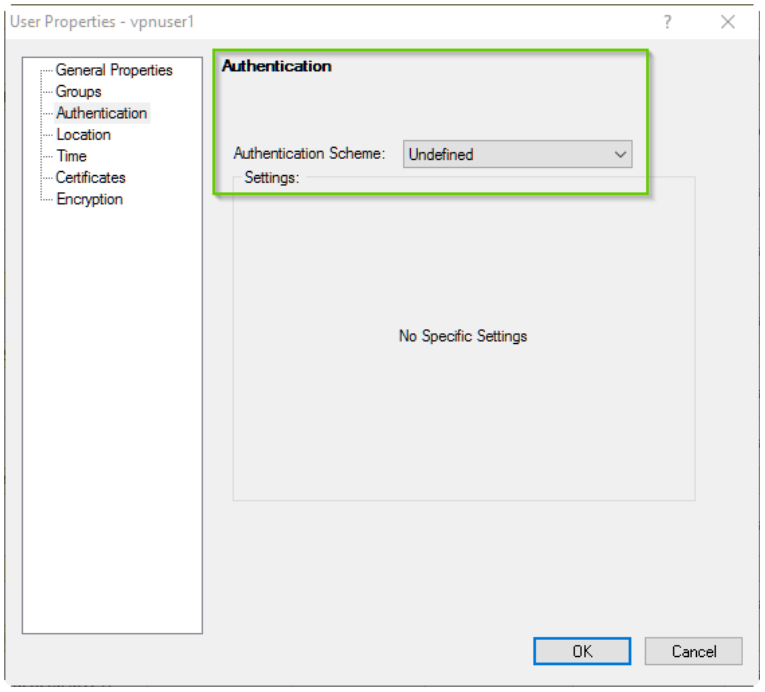
Two-Factor Authentication with DynamicID

- ● Global settings (under "Authentication to Gateway" on the Mobile Access tab)
- ○ Custom settings for this gateway
- ☑ Allow DynamicID for mobile devices

Certificate Authentication for mobile devices

- ☐ Require client certificate when using Mobile applications
- ☐ Require client certificate when using ActiveSync applications

In this case, you can leave the “Authentication” settings for the VPN users, or the template on which you are relying for their creation, as “Undefined”:


User Properties - vpnuser1

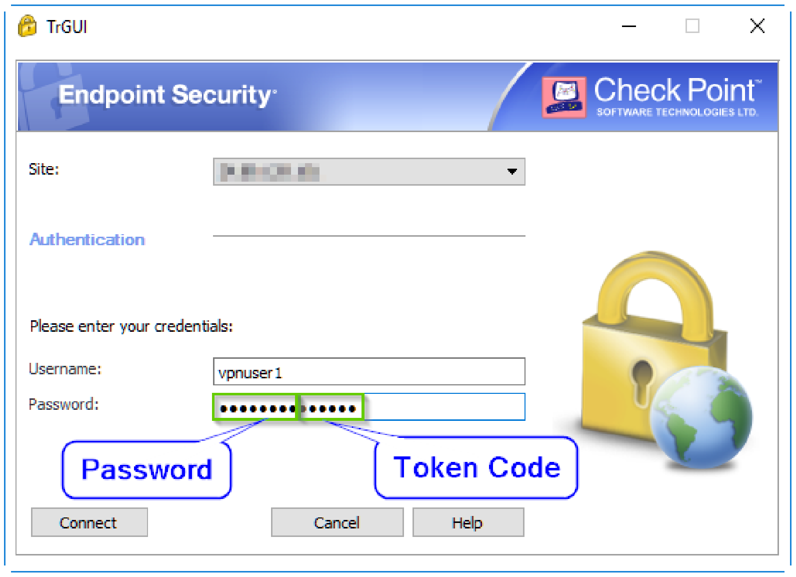
Authentication

Authentication Scheme: Undefined

Settings:

No Specific Settings

Users then will be forced to use the default authentication method assigned to specific blades and will be relying on RADIUS passwords and codes generated by the Google Authenticator.



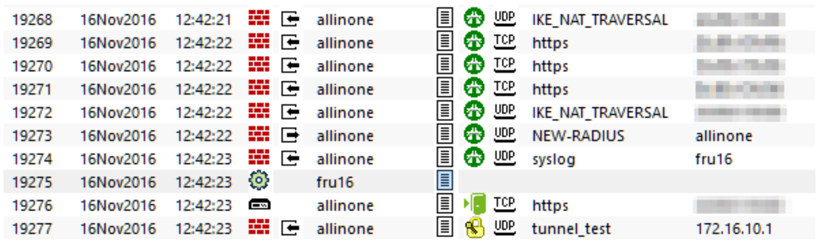
Site: [redacted]

Please enter your credentials:

Username: vpnuser1

Password: ••••••••••••••

A typical VPN establishment and authentication process log will then look like this:



```
19268  16Nov2016  12:42:21  allinone  UDP  IKE_NAT_TRAVERSAL  [redacted]
19269  16Nov2016  12:42:22  allinone  TCP  https              [redacted]
19270  16Nov2016  12:42:22  allinone  TCP  https              [redacted]
19271  16Nov2016  12:42:22  allinone  TCP  https              [redacted]
19272  16Nov2016  12:42:22  allinone  UDP  IKE_NAT_TRAVERSAL  [redacted]
19273  16Nov2016  12:42:22  allinone  UDP  NEW-RADIUS         allinone
19274  16Nov2016  12:42:23  allinone  UDP  syslog             fru16
19275  16Nov2016  12:42:23  fru16
19276  16Nov2016  12:42:23  allinone  TCP  https              [redacted]
19277  16Nov2016  12:42:23  allinone  UDP  tunnel_test        172.16.10.1
```

Logging Authentication Events

By default, FreeRADIUS events are being logged on your Ubuntu server in /var/log/freeradius/radius.log file.

Running the “tail” command will show you the last entries in this log:

```
root@fru16:~# cd /var/log/freeradius/
root@fru16:/var/log/freeradius# ls
radius.log  radwtmp
root@fru16:/var/log/freeradius# tail radius.log
Sat Nov 12 19:59:37 2016 : Auth: Login OK: [cpadmin] (from client allinoneDMZ
port 0)
Sat Nov 12 20:05:52 2016 : Auth: Login OK: [cpadmin] (from client allinoneDMZ
port 0)
Sat Nov 12 20:17:49 2016 : Info: Signalled to terminate
Sat Nov 12 20:17:49 2016 : Info: Exiting normally.
Sat Nov 12 20:20:12 2016 : Info: Loaded virtual server <default>
Sat Nov 12 20:20:12 2016 : Info: Loaded virtual server inner-tunnel
Sat Nov 12 20:20:12 2016 : Info: Ready to process requests.
Sat Nov 12 20:28:21 2016 : Auth: Login OK: [cpadmin] (from client Precision port
0)
```

If you would like to observe the log dynamically, execute:

```
tail -f /var/log/freeradius/radius.log
```

This will result in continuously scrolling log entries in your terminal session.

To filter the content of the log for a client or user, (“cpmonitor” in this example):

```
tail -f /var/log/freeradius/radius.log | grep cpmonitor
```

In this case the output of the log file will be limited to the entries specified in a filter, with filtered parameters highlighted in red:

```
root@fru16:~# tail -f /var/log/freeradius/radius.log | grep cpmonitor
Nov 17 12:54:35 fru16 freeradius[8588]: Login OK: [cpmonitor] (from client
allinoneDMZ port 2083 cli 192.168.7.147)
Nov 18 12:36:36 fru16 freeradius[8588]: Login OK: [cpmonitor] (from client
allinoneDMZ port 28533 cli 192.168.7.147)
Nov 18 12:39:19 fru16 freeradius[8588]: Login OK: [cpmonitor] (from client
allinoneDMZ port 29634)
Nov 18 12:40:53 fru16 freeradius[11960]: Login OK: [cpmonitor] (from client
allinoneDMZ port 30686 cli 192.168.7.147)
root@fru16:~#
```

To exit the scrolling log, press **Ctrl + c**

It is considered good practice to consolidate log files from critical infrastructure components to centralized logging servers. To achieve that, the log output of FreeRADIUS should be redirected to the SIEM facility of your choice.

In the next section of this document I will demonstrate how to do this by forwarding logs to the Check Point management or logging servers.

Redirecting FreeRADIUS Logs to Check Point Management or Log Server

Redirection of the logs requires alteration of default configuration files for FreeRADIUS and rsyslog.

First, let’s make these changes in the radius.conf file:

Execute:

```
sudo nano /etc/freeradius/radius.conf
```

Press **Ctrl + w** and search for “log {“

Once you find this section, comment out the “destination = files” and add the line “destination = syslog” below:

```
log {
        #
        #  Destination for log messages.  This can be one of:
        #
        #       files - log to "file", as defined below.
        #       syslog - to syslog (see also the "syslog_facility", below.
        #       stdout - standard output
        #       stderr - standard error.
        #
        #  The command-line option "-X" over-rides this option, and forces
        #  logging to go to stdout.
        #
        #destination = files
        destination = syslog
```

Continue searching in same file, press **Ctrl + w** and search for “syslog_facility = daemon“.

Once found, comment this line out and add "syslog_facility = local1" below, so that this portion of the file looks like:

```
        #  The exact values permitted here are OS-dependent.  You probably
        #  don't want to change this.
        #
        #syslog_facility = daemon
        syslog_facility = local1
        #  Log the full User-Name attribute, as it was found in the request.
        #
        # allowed values: {no, yes}
        #
        stripped_names = no
```

Press **Ctrl + x** ; **y** and **Enter** to save changes and exit the nano editor.

Execute:

```
sudo nano /etc/rsyslog.conf
```

Press **Ctrl + w** and search for "Set the default permissions for all log files "

Comment out lines "$PrivDropToUser syslog" and "$PrivDropToGroup syslog", and paste these two lines below:

```
$PrivDropToUser adm
$PrivDropToGroup adm
```

Scroll down to the bottom of the file and paste these two lines at the end:

```
local1.*        /var/log/fradius.log
*.*     @ip_address_of_CheckPoint_management_or_log_server
```

Replace "ip_address_of_CheckPoint_management_or_log_server" with the actual IP Address of your log server.

Note that FreeRADIUS log file destination was changed from /var/log/freeradius/radius.log to /var/log/fradius.log.

Press **Ctrl + x** ; **y** and **Enter** to save changes and exit the nano editor.

Note that in our case, since FreeRADIUS is isolated in DMZ and is configured as a narrow purpose server, we can forward all the logs to the Check Point. If you would like to narrow down the scope of forwarded events, you may use the two lines shown below instead of those shown earlier:

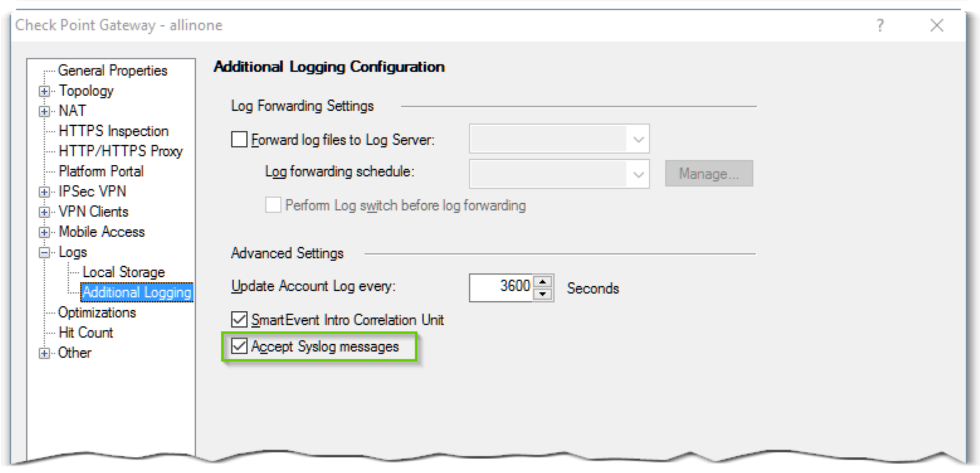
```
local1.*        /var/log/fradius.log
*.=notice     @ip_address_of_CheckPoint_management_or_log_server
```

Once you are out of the nano editor, execute the following line for changes to take effect:

```
sudo reboot now
```

Configure Check Point Management or Log Server to Accept Syslog Messages

In SmartDashboard, open the properties of your management or logging server, navigate to “Logs\Additional Logging” and check the box labeled “Accept Syslog messages”:


Check Point Gateway - allinone

Additional Logging Configuration

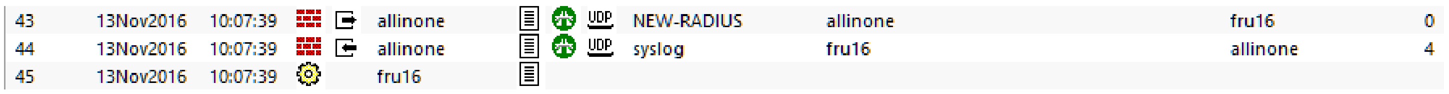
Log Forwarding Settings
- [ ] Forward log files to Log Server:
- Log forwarding schedule:
- [ ] Perform Log switch before log forwarding

Advanced Settings
- Update Account Log every: 3600 Seconds
- [x] SmartEvent Intro Correlation Unit
- [x] Accept Syslog messages

Load the policy.

Now your management or logging server is ready to receive the logon events from FreeRADIUS.

With the “Log Implied Rules” enabled in the “Global Properties”, a typical sequence of authentication looks like this:

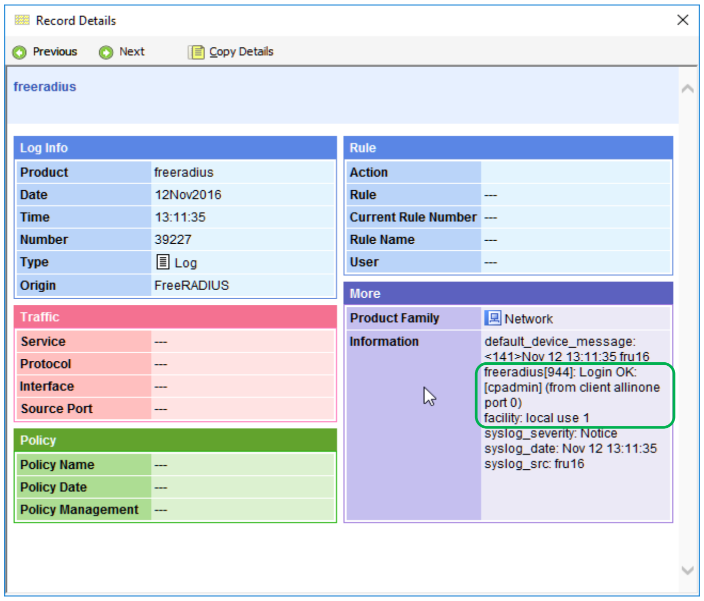

```
43  13Nov2016  10:07:39  allinone  UDP  NEW-RADIUS  allinone  fru16    0
44  13Nov2016  10:07:39  allinone  UDP  syslog      fru16     allinone 4
45  13Nov2016  10:07:39  fru16
```

You can see on line 43, that the NEW-RADIUS connection is treated by Implied Rules, (0 on the right is indicative of the Implied Rules).

Syslog inbound connection on line 44, triggered by the authentication event, is treated by the policy rule number 4.

Line 45 indicates that syslog payload is being processed and contains information about login attempts.

Syslog messages do not require "Log Implied Rules" to be enabled and will be visible in the tracker, where they could be subjected to filtering based on the content of "Information" field.



freeradius

Log Info	
Product	freeradius
Date	12Nov2016
Time	13:11:35
Number	39227
Type	Log
Origin	FreeRADIUS

Rule	
Action	
Rule	---
Current Rule Number	---
Rule Name	---
User	---

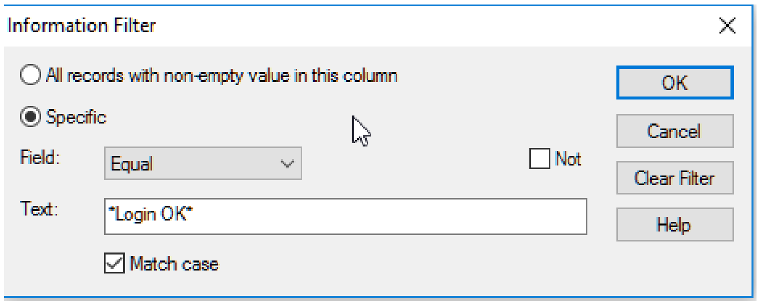
Traffic	
Service	---
Protocol	---
Interface	---
Source Port	---

Policy	
Policy Name	---
Policy Date	---
Policy Management	---

More	
Product Family	Network
Information	default_device_message: <141>Nov 12 13:11:35 fru16 freeradius[944]: Login OK: [cpadmin] (from client allinone port 0) facility: local use 1 syslog_severity: Notice syslog_date: Nov 12 13:11:35 syslog_src: fru16

In the SmartView Tracker, scroll to the right until column "Information" becomes visible.

Right click on the "Information" tab and select "Edit Filter":



All records with non-empty value in this column [✓ Specific]

Field: Equal

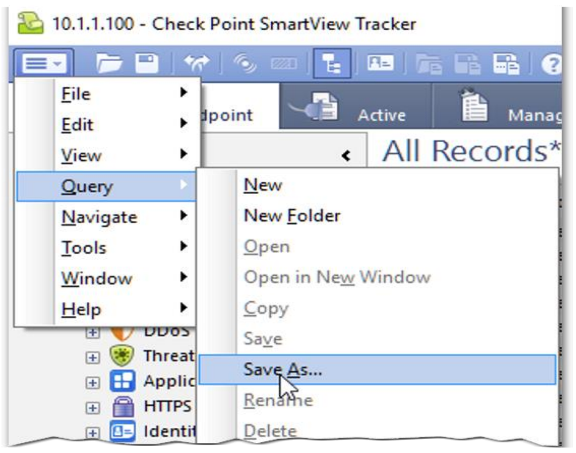
Not

Text: *Login OK*

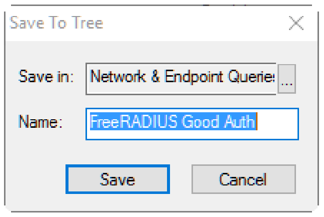
[✓] Match case

Enter “*Login OK*” in the “Text” field, select “Match Case” and click “OK”.

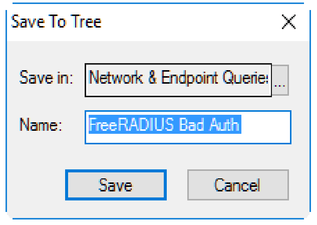
In SmartView Tracker, click on the “Launch Menu”, “Query” and “Save As”:

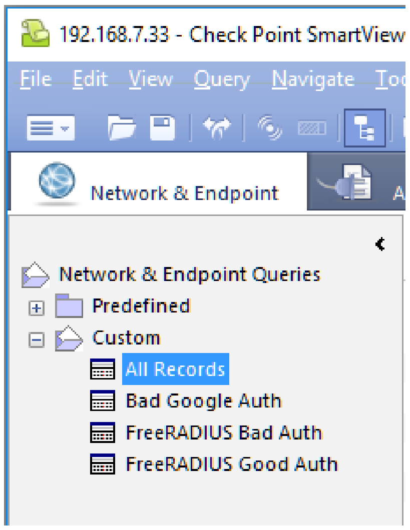


In the “Save to Tree” dialog box, enter the name you would like to assign to successful RADIUS authentication attempts and click “Save”:



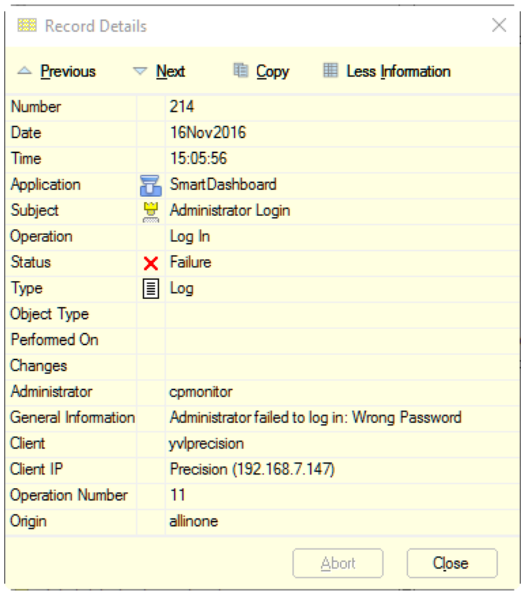
Repeat above steps, changing filter parameters to “*Login incorrect*” and save the failed login query:



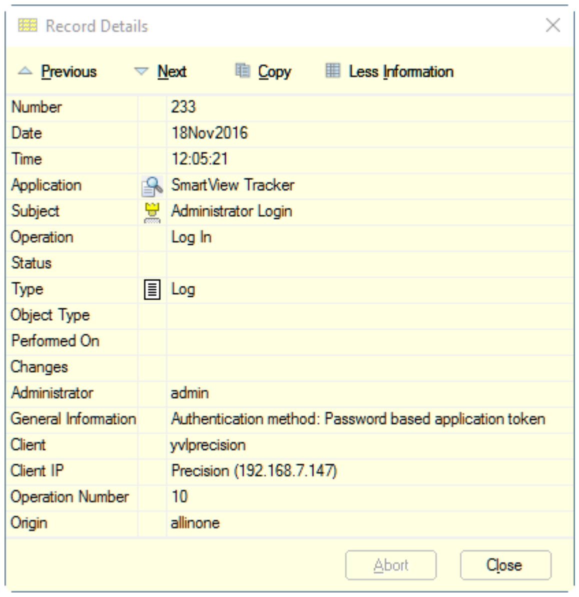


You will now have custom queries at your fingertips under the “Network and Endpoint” log tab, in “Custom” section:

This is in addition to the actual administrative logon attempts registered under the “Management” Tab:



Number	214
Date	16Nov2016
Time	15:05:56
Application	SmartDashboard
Subject	Administrator Login
Operation	Log In
Status	Failure
Type	Log
Object Type	
Performed On	
Changes	
Administrator	cpmonitor
General Information	Administrator failed to log in: Wrong Password
Client	yvlprecision
Client IP	Precision (192.168.7.147)
Operation Number	11
Origin	allinone



Number	233
Date	18Nov2016
Time	12:05:21
Application	SmartView Tracker
Subject	Administrator Login
Operation	Log In
Status	
Type	Log
Object Type	
Performed On	
Changes	
Administrator	admin
General Information	Authentication method: Password based application token
Client	yvlprecision
Client IP	Precision (192.168.7.147)
Operation Number	10
Origin	allinone

Testing and Debugging

To verify FreeRADIUS functionality and to observe the authentication attempts details, execute:

```
sudo service freeradius stop
```

and

```
sudo freeradius -X
```

You will now have a verbose output of FreeRADIUS startup and logon attempt processing, which can be of great help in troubleshooting.

This is a typical output of logon attempt in debugging mode:

```
Ready to process requests.
rad_recv: Access-Request packet from host 10.2.2.1 port 29378, id=62, length=94
        User-Name = "cpmonitor"
        User-Password = "clear_text_password968932"
        NAS-IP-Address = 192.168.7.33
        NAS-Identifier = "sshd"
        NAS-Port = 28352
        NAS-Port-Type = Virtual
        Service-Type = Authenticate-Only
        Calling-Station-Id = "192.168.7.147"
# Executing section authorize from file /etc/freeradius/sites-enabled/default
+group authorize {
++[preprocess] = ok
++[chap] = noop
++[mschap] = noop
++[digest] = noop
[suffix] No '@' in User-Name = "cpmonitor", looking up realm NULL
[suffix] No such realm "NULL"
++[suffix] = noop
[eap] No EAP-Message, not doing EAP
++[eap] = noop
[files] users: Matched entry DEFAULT at line 70
++[files] = ok
++[expiration] = noop
++[logintime] = noop
[pap] WARNING! No "known good" password found for the user.  Authentication may
fail because of this.
++[pap] = noop
+} # group authorize = ok
Found Auth-Type = PAM
# Executing group from file /etc/freeradius/sites-enabled/default
+group authenticate {
pam_pass: using pamauth string <radiusd> for pam.conf lookup
pam_pass: authentication succeeded for <cpmonitor>
++[pam] = ok
+} # group authenticate = ok
Login OK: [cpmonitor] (from client allinoneDMZ port 28352 cli 192.168.7.147)
```

```
# Executing section post-auth from file /etc/freeradius/sites-enabled/default
+group post-auth {
++[exec] = noop
+} # group post-auth = noop
Sending Access-Accept of id 62 to 10.2.2.1 port 29378
Finished request 1.
Going to the next request
Waking up in 4.9 seconds.
Cleaning up request 1 ID 62 with timestamp +59
Ready to process requests.
```

Keep in mind, that when you are running FreeRADIUS in debugging mode, its actions are NOT written to the logs, they are outputted only to your console session.

To return FreeRADIUS to normal operation, execute:

```
sudo service freeradius stop
```

and

```
sudo service freeradius start
```

Possible Future Improvements and Acknowledgements

FreeRADIUS implementation described above is suitable for small to medium environments where administrators would like to take advantage of Google Authenticator for a limited number of accounts.

This exercise was inspired by the blog entry http://www.supertechguy.com/help/security/freeradius-google-auth by Jeremy Cox.

For larger environments, use FreeRADIUS with MySQL as a back end, setup MySQL replication to secondary server and define multiple RADIUS servers in Gaia, as well as RADIUS group in the SmartDashboard.

You may also decide to set up a front-end Web UI for your FreeRADIUS implementation using DaloRADIUS, (see https://thenetworkcable.wordpress.com/2014/10/26/configuring-daloradius-as-a-freeradius-frontend/

for references).

There is an excellent write-up on the subject of MFA at:

https://www.howtoforge.com/two-factor-authentication-with-otp-using-privacyidea-and-freeradius-on-centos

and

https://www.howtoforge.com/manage-two-factor-authentication-in-your-serverfarm-with-privacyidea

that I am planning to investigate further.