

PacketFence 

Installation Guide

PacketFence v11.1.0

Version 11.1.0 - January 2022

Table of Contents

1. About this Guide	2
1.1. Other sources of information	2
2. Introduction	3
3. System Requirements	4
3.1. Assumptions	4
3.2. Minimum Hardware Requirements	4
3.3. Operating System Requirements	4
4. Installation	6
4.1. Installing PacketFence from the ZEN	6
4.2. Installing PacketFence on existing Linux	7
5. Getting Started	10
5.1. Going Through the Configurator	10
5.2. Connecting PacketFence to Microsoft Active Directory	11
5.3. Configuring Cisco Catalyst 2960 Switch	11
5.4. Adding the Switch to PacketFence	13
5.5. Configuring the Connection Profile	13
5.6. Configuring Microsoft Windows Suppllicant	14
5.7. Testing	14
5.8. Alerting	14
6. Enabling the Captive Portal	15
6.1. Creating Authentication Source for Guests	15
6.2. Configure switchport for Web Authentication	15
6.3. Adjust Switch Configuration in PacketFence	16
6.4. Enabling Portal on Management Interface	16
6.5. Configuring the Connection Profile	17
6.6. Testing	17
7. Authentication Sources	18
7.1. Email Authentication for Guests	19
7.2. Adding SMS Authentication for Guests	20
8. Introduction to Role-based Access Control	22
8.1. Adding Roles	22
8.2. Using the Employee Role	23
8.3. Using the Corporate_Machine Role	23
9. Supported Enforcement Modes	25
9.1. Technical Introduction to Inline Enforcement	25
9.2. Technical Introduction to Out-of-band Enforcement	26
9.3. Technical Introduction to Hybrid Enforcement	30
9.4. Technical Introduction to RADIUS Enforcement	31
9.5. Technical Introduction to DNS Enforcement	31
10. Adding Inline Enforcement to Existing Installation	33
10.1. Introduction	33
10.2. Preparing the Operating System	33
10.3. Adding Inline Interface	33
10.4. Network Devices	35
10.5. Adding Connection Profile for Inline	35
10.6. Testing the Inline Configuration	35

10.7. Advanced Inline Topics	36
11. Adding VLAN Enforcement to Existing Installation	37
11.1. Introduction	37
11.2. Adding the Registration, Isolation and Other Interface	38
11.3. Network Devices	39
11.4. Adding Connection Profile for Registration	40
12. Troubleshooting PacketFence	42
12.1. RADIUS Audit Log	42
12.2. Log files	42
12.3. RADIUS Debugging	42
13. Authentication Mechanisms	44
13.1. Microsoft Active Directory (AD)	44
13.2. OAuth2 Authentication	51
13.3. Eduroam	54
13.4. SAML Authentication	57
13.5. Billing Engine	59
13.6. External API Authentication	71
13.7. Azure AD integration	72
13.8. Google Workspace LDAP Integration	74
14. Advanced Portal Configuration	76
14.1. Portal Modules	76
14.2. Portal Surveys	84
14.3. Devices Registration	89
14.4. Status page	89
14.5. Passthroughs	90
14.6. Proxy Interception	91
14.7. Parked Devices	91
15. Advanced Access Configuration	93
15.1. Connection Profiles	93
15.2. Filter Engine Macro	100
15.3. VLAN Filter Definition	103
15.4. RADIUS Filter Definition	104
15.5. Advanced LDAP Authentication	106
15.6. Advanced Realm Configuration	107
16. Advanced RADIUS Configuration	109
16.1. Local Authentication	109
16.2. Authentication against Active Directory (AD)	109
16.3. EAP Authentication against OpenLDAP	109
16.4. EAP Guest Authentication on Email, Sponsor and SMS Registration	110
16.5. EAP Local User Authentication	111
16.6. Limit Brute Force EAP Authentication	111
16.7. Testing	111
16.8. RADIUS Accounting	112
16.9. RADIUS Proxy	113
16.10. RADIUS EAP Profiles	116
17. Fingerbank Integration	117
17.1. Onboarding	117
17.2. Update Fingerbank Database	117
17.3. Submit Unknown Data	117
17.4. Upstream Interrogation	117
17.5. Local Entries	118
17.6. Settings	118
17.7. Device change detection	118
18. Network Devices Anomaly Detection	119

18.1. Creating Network Behavior Policies	119
18.2. Integration with Security Events	119
19. Tenants	120
19.1. General concepts	120
19.2. Getting started	120
20. Intrusion Detection System Integration	122
20.1. Regex Syslog Parser	122
20.2. Suricata IDS	123
20.3. Security Onion	125
20.4. Security Onion 2.3.10	127
20.5. ERSPAN	130
20.6. StreamScan Compromise Detection System (CDS)	131
21. Firewall SSO Integration	135
21.1. Barracuda	135
21.2. Checkpoint	137
21.3. Cisco ISE-PIC	141
21.4. FortiGate	143
21.5. iBoss	146
21.6. JSON-RPC	146
21.7. Juniper SRX	147
21.8. Palo Alto	149
22. Performing Compliance Checks	154
22.1. Installation	154
22.2. Configuration	156
22.3. Rapid7 integration	160
23. Integrating Provisioning Agents	168
23.1. PacketFence Apple, Android and Windows Wireless Provisioning	168
23.2. PacketFence Apple, Android and Windows Wireless Provisioning	168
23.3. MobileIron	172
23.4. OPSWAT	182
23.5. SentinelOne	192
23.6. Symantec SEPM	195
23.7. Microsoft Intune	204
23.8. Google Chromebook Provisioner	209
23.9. Creating a service account JSON config	209
23.10. Configure PacketFence	210
24. PKI Integration	212
24.1. Microsoft PKI	212
24.2. PacketFence PKI	225
24.3. AirWatch	245
25. MFA Integration	255
25.1. Assumptions	255
25.2. Create the MFA Configuration	255
26. Best Practices	261
26.1. IPTables	261
26.2. Log Rotations	261
26.3. Large Registration Network	261
26.4. Active Directory fail-over	261
27. Performance Optimizations	264
27.1. NTLM Authentication Caching	264
27.2. SNMP Traps Limit	265
27.3. MariaDB optimizations	266
27.4. Captive Portal Optimizations	269
27.5. Troubleshooting	269

28. Advanced Network Topics	271
28.1. Floating Network Devices	271
28.2. Production DHCP access	272
28.3. Routed Networks	274
28.4. Network Devices Definition (switches.conf)	277
28.5. More on VoIP Integration	281
28.6. DHCP Option 82	283
29. Additional Integration	284
29.1. DHCP Remote Sensor	284
29.2. Active Directory Integration	285
29.3. Switch Login Access	291
29.4. Syslog forwarding	292
29.5. Monit	292
30. Advanced Topics	295
30.1. Reports	295
30.2. Admin Access	305
30.3. Guest pre-registration	306
30.4. Content-Security-Policy (CSP)	306
30.5. pfacct : track bandwidth usage	307
31. Export/Import mechanism	308
31.1. Assumptions and limitations	308
31.2. Export on current installation	309
31.3. Import on new installation	309
32. Automation of upgrades	312
32.1. Assumptions and limitations	312
32.2. Full upgrade (for PacketFence version 11.0.0 only)	312
32.3. Full upgrade (for PacketFence versions 11.1.0 and later)	312
33. Additional Information	313
34. Commercial Support and Contact Information	314
35. GNU Free Documentation License	315
36. Appendix	316
Appendix A: Administration Tools	316
Appendix B: Restoring a Percona XtraBackup or Mariabackup dump	318
Appendix C: How to restore a standalone PacketFence server ?	319

Copyright © 2022 Inverse inc.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

The fonts used in this guide are licensed under the SIL Open Font License, Version 1.1. This license is available with a FAQ at: <http://scripts.sil.org/OFL>

Copyright © Łukasz Dziejczak, <http://www.latofonts.com/>, with Reserved Font Name: "Lato".

Copyright © Raph Levien, <http://levien.com/>, with Reserved Font Name: "Inconsolata".

᠑ᠡᠯᠢᠨᠢ

1. About this Guide

This guide will walk you through the installation and the day to day administration of the PacketFence solution.

The latest version of this guide is available at <https://packetfence.org/documentation/>

1.1. Other sources of information

Clustering Guide

Covers installation in a clustered environment.

Developer's Guide

Covers API, captive portal customization, application code customizations and instructions for supporting new equipment.

Network Devices Configuration Guide

Covers switches, WiFi controllers and access points configuration.

Upgrade Guide

Covers compatibility related changes, manual instructions and general notes about upgrading.
[PacketFence News](#) Covers noteworthy features, improvements and bug fixes by release.

These files are included in the package and release tarballs.

2. Introduction

PacketFence is a fully supported, trusted, Free and Open Source network access control (NAC) system. Boosting an impressive feature set including a captive portal for registration and remediation, centralized wired and wireless management, 802.1X support, layer-2 isolation of problematic devices, integration with IDS, vulnerability scanners and firewalls; PacketFence can be used to effectively secure networks - from small to very large heterogeneous networks. For a more detailed presentation on PacketFence please visit <https://packetfence.org>.

3. System Requirements

3.1. Assumptions

PacketFence reuses many components in an infrastructure. Nonetheless, it will install the following ones and manage them itself:

- database server (MariaDB)
- web server (Apache)
- DHCP server (PacketFence)
- RADIUS server (FreeRADIUS)
- firewall (iptables)

In this guide, we assume that all those components are running on the same server (i.e., "localhost" or "127.0.0.1") that PacketFence will be installed on.

Good understanding of those underlying components and GNU/Linux is required to install PacketFence. When installing PacketFence, all these components will be properly installed. Moreover, PacketFence will manage the services listed above. Make sure that all the other services are automatically started by your operating system.

3.2. Minimum Hardware Requirements

The following provides a list of the minimum server hardware recommendations:

- Intel or AMD CPU 3 GHz, 2 CPU cores
- 12 GB of RAM (16 GB recommended)
- 100 GB of disk space (RAID-1 recommended)
- 1 network card (2 recommended)

3.2.1. Recommendations

- Use logical volume management (LVM) to allocate space

3.3. Operating System Requirements

PacketFence supports the following operating systems on the x86_64 architecture:

- Red Hat Enterprise Linux 8.x Server
- Debian 11.x (Bullseye)

Make sure that you can install additional packages from your standard distribution. For example, if you are using Red Hat Enterprise Linux, you have to be subscribed to the Red Hat Network before continuing with the PacketFence software installation.

Other distributions such as RHEL derivatives are known to work but this document does not cover them.

4. Installation

This section will guide you through the installation of PacketFence from the Zero Effort NAC (ZEN) appliance and from the standard repository of packages we provide - which can be used to install PacketFence on top of a vanilla GNU/Linux installation.

4.1. Installing PacketFence from the ZEN

The ZEN (Zero Effort NAC) edition of PacketFence allows you to rapidly get PacketFence running in your network environment. It consists of a fully installed and preconfigured version of PacketFence distributed as a virtual appliance. It can be deployed on VMware ESX/ESXi, Microsoft Hyper-V and other products. This section covers the deployment of the virtual appliance on VMware-based products. We are not supporting any Xen-based hypervisors yet.

You can download the ZEN here: <https://www.packetfence.org/download.html#/zen>

4.1.1. Virtual Machine

This setup has been tested using VMware ESXi, Fusion and Workstation products with 12 GB of RAM dedicated to the virtual machine. It might work using other VMware products. To properly run the PacketFence virtual appliance, you need a CPU that supports long mode. In other words, you need to have a 64-bit capable CPU on your host. PacketFence ZEN comes in a pre-built virtual disk (OVF). If you are using an ESX type hypervisor, you need to import the OVF using vSphere Client (or vCenter).

First network card of virtual machine is configured to receive an IP through DHCP.

The virtual appliance passwords are:

Management (Console/SSH) user

- Login: root
- Password: p@ck3tf3nc3

WARNING

Be sure to change default passwords if you plan to use this image in production.

4.1.2. Import to ESX

Make sure that there is only one virtual network card created, and also make sure that your vEthernet is connected to a virtual switch (vSwitch). That virtual network card will be used as the PacketFence management interface.

4.1.3. Import to VMware Player/Workstation for Linux

Newer versions of VMware Player handle VLAN trunking a lot better. With that in mind, we can use a single interface on the VM. So, you need to ensure that your VM host is plugged into a physical trunk port with VLAN 1,2,3,5,10 and 200 as the allowed VLAN. These VLANs will be

used later in configuration examples.

4.2. Installing PacketFence on existing Linux

PacketFence provides packages repository for RHEL 8 as well as packages repository fo Debian.

These repositories contain all required dependencies to install PacketFence. This provides numerous advantages. Among them, there are:

- easy installation
- everything is packaged as RPM and Debian packages
- easy upgrade

First install your supported distribution with minimal installation and no additional packages. Then:

On Red Hat-based systems

- Disable firewall
- Disable SELinux

On Debian

- Disable AppArmor
- Disable resolvconf

NOTE: If running **UEFI mode**, make sure **secureboot** is **disabled**.

Make sure your system is up to date and your yum or apt-get database is updated. On a RHEL-based system, do:

```
yum update
```

On a Debian system, do:

```
apt-get update  
apt-get upgrade
```

Regarding SELinux or AppArmor, even if they may be wanted by some organizations, PacketFence will not work properly if SELinux or AppArmor are enabled. You will need to explicitly disable SELinux from the `/etc/selinux/config` file and reboot the machine. For AppArmor, you need to follow instructions on [Debian wiki](#).

Regarding resolvconf, you can remove the symlink to that file and simply create the `/etc/resolv.conf` file with the content you want.

4.2.1. RHEL-based systems

Install kernel development package:

```
yum install kernel-devel-$(uname -r)
```

NOTE

Make sure you are actually running the latest kernel prior to installing the kernel development package. Reboot prior to installing this package if unsure.

RHEL 8.x

You need to have a valid subscription to be able to install PacketFence dependencies.

4.2.2. Debian-based systems

Install kernel development package:

```
apt install linux-headers-$(uname -r)
```

NOTE

Make sure you are actually running the latest kernel prior to installing the kernel development package. Reboot prior to installing this package if unsure.

4.2.3. Software Installation

RHEL-based systems

In order to use the PacketFence repository:

```
yum localinstall  
http://packetfence.org/downloads/PacketFence/RHEL8/packetfence-release-  
11.1.el8.noarch.rpm
```

Once the repository is defined, you can install PacketFence with all its dependencies, and the required external services (database server, DHCP server, RADIUS server) using:

```
yum install --enablerepo=packetfence packetfence
```

Debian-based systems

In order to use the repository, create a file named `/etc/apt/sources.list.d/packetfence.list`:

```
echo 'deb http://inverse.ca/downloads/PacketFence/debian/11.1 bullseye  
bullseye' > \  
/etc/apt/sources.list.d/packetfence.list
```

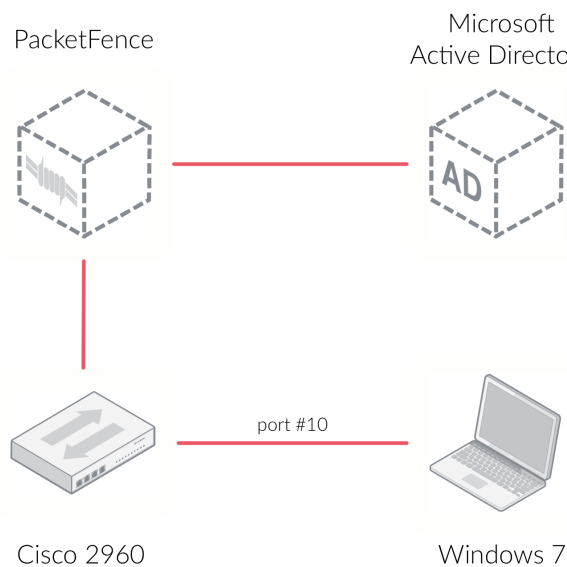
Once the repository is defined, you can install PacketFence with all its dependencies, and the required external services (Database server, DHCP server, RADIUS server) using:

```
apt install gnupg sudo
wget -q -O - https://inverse.ca/downloads/GPG_PUBLIC_KEY | sudo apt-key add -
apt-get update
apt-get install packetfence
```

5. Getting Started

Now that PacketFence is installed, it needs to be configured. The PacketFence web-based configuration interface will automatically be started.

This section will guide you through configuring PacketFence as a simple RADIUS server. PacketFence will provide 802.1X support through Microsoft Active Directory and a Cisco 2960 access switch will be configured to integrate with PacketFence. The 802.1X client will be a Microsoft Windows 7 computer, connected of course on the wired network in the Cisco 2960 access switch. The following architecture diagram shows the interconnection of all components for our example:



NOTE

If you use another access switch, you must refer to PacketFence Network Devices Configuration Guide to adapt your configuration.

5.1. Going Through the Configurator

First open PacketFence's configurator - you can access it from https://@ip_of_packetfence:1443. If you are unsure what IP address you have, run `ip a` in your Linux shell. Perform the following actions:

- Step 1 - **Configure Network** - make sure you define only one interface with the "Management" type. That network interface will be the one to which the Cisco 2960 access switch will talk to. The management interface of PacketFence and the Cisco 2960 should

normally be in the same network. To set the interface to the "Management" type, click on the logical name to edit it

- Step 2 - **Configure PacketFence** - provide the required information to properly create the PacketFence database and also provide your domain name, hostname and other required information. Make you sure to provide the PacketFence's admin username and password to be used
- Step 3 - **Fingerbank** - provide your Fingerbank API key. Fingerbank is used to accurately identify Internet of Things (IoT) devices, medical devices, industrial and robotics equipment and more on your network. It is recommended to have a key for your PacketFence deployment. Without a Fingerbank API key, device profiling will not be available in PacketFence
- Step 4 - **Confirmation** - save the passwords in a secure location and start PacketFence!

Once all services are started, you will automatically be redirected to the PacketFence's web admin interface. It is located at https://@ip_of_packetfence:1443/. Open that link and log in using the username/password specified in Step 2.

5.2. Connecting PacketFence to Microsoft Active Directory

Next, we join the PacketFence server to your existing Microsoft Active Directory domain controller. From PacketFence's web admin interface, go in *Configuration* → *Policies and Access Control* → *Domains* → *Active Directory Domain* and click on the **New domain** button. Provide the required fields. You will need an Active Directory administrative username and password (member of the domain admins) to join the PacketFence server to your domain. Once all the information has been provided, click on the **Create & Join** button.

Once the domain join succeeds, click on the **REALMS** tab. Click on the **Default** realm and set the domain to the Active Directory domain you have just created. That will instruct PacketFence to use that newly created Active Directory for the default authentication realm. Next, do the same thing for the 'NULL' realm.

Next, we add the Microsoft Active Directory domain controller as an authentication source in PacketFence. To do so, from *Configuration* → *Policies and Access Control* → *Authentication Sources*, click on **New internal source AD**. Specify all the required fields. If you need help identifying fields relevant to your Active Directory environment, please use the Active Directory Explorer (AD Explorer) or AdsiEdit.mmc tools from your Active Directory server.

In this new 'Authentication Source', add an 'Authentication Rules' with name 'catchall' with no condition and with the following actions:

- Role - default
- Access duration - 5 days

Make sure the information you provided are valid. Click on the **Test** button to validate the provided information. If you see the message 'Success! LDAP connect, bind and search successful' - you have properly configured your Microsoft Active Directory authentication source. Save your new authentication source by clicking on the **Save** button.

5.3. Configuring Cisco Catalyst 2960 Switch

Next, we configure a switch so that it integrates with PacketFence using 802.1X. In our example, we will use a Cisco Catalyst 2960 access switch and its IP address will be 172.21.2.3. Our

PacketFence's server IP address will be 172.20.100.2 - you will need to adjust this according to your environment.

Connect to that switch over SSH as an admin.

5.3.1. Enable 802.1X

As a first configuration step, you need to enable 802.1X globally on the switch. To do so, use the following:

```
dot1x system-auth-control
```

5.3.2. Configure AAA

The next step is to configure AAA so it will use your newly created PacketFence server. Make sure you replace the PF_MANAGEMENT_IP variable with your actual PacketFence management IP (172.20.100.2 in our example) in the following commands:

```
aaa new-model
aaa group server radius packetfence
  server PF_MANAGEMENT_IP auth-port 1812 acct-port 1813
aaa authentication login default local
aaa authentication dot1x default group packetfence
aaa authorization network default group packetfence
radius-server host PF_MANAGEMENT_IP auth-port 1812 acct-port 1813 timeout 2 key
useStrongerSecret
radius-server vsa send authentication
snmp-server community public RO
snmp-server community private RW
```

5.3.3. Configure Switchport for 802.1X

Once AAA is ready, we can configure some or all switchports to perform 802.1X. In our example, we will only configure port no. 10 to use 802.1X:

```
interface fastEthernet 0/10
switchport mode access
authentication host-mode single-host
authentication order dot1x mab
authentication priority dot1x mab
authentication port-control auto
authentication periodic
authentication timer restart 10800
authentication timer reauthenticate 10800
mab
no snmp trap link-status
dot1x pae authenticator
dot1x timeout quiet-period 2
dot1x timeout tx-period 3
```

Write the switch configuration to memory.

5.4. Adding the Switch to PacketFence

PacketFence must be aware of the equipment it manages. From *Configuration* → *Policies and Access Control* → *Network Devices* → *Switches*, click on **New Switch** **default**. Enter your switch IP address (172.21.2.3 in our example). As a switch type, select **Cisco Catalyst 2960** and select **Production** as the Mode. From the 'Roles' tab, make sure 'Role by VLAN ID' is checked and that the VLAN ID associated to the default role is set to your normal VLAN currently in use on your network. In our example, it will be VLAN 20. That means that once a 802.1X authentication is allowed by PacketFence, access will be properly granted in the default role in VLAN 20.

From the 'RADIUS' tab, specify the 'Secret Passphrase' to use - in our example, it is 'useStrongerSecret'. It is very important to correctly set the RADIUS secret passphrase otherwise PacketFence will prevent the switch from communicating to itself.

Finally, from the 'SNMP' tab, provide the correct 'Community Read' and 'Community Write' values.

5.5. Configuring the Connection Profile

Next, we need to configure the connection profile in PacketFence. That is required so that PacketFence knows how to handle a connection coming from the wired network or WiFi network. In our case, we will create a new connection profile to use our Microsoft Active Directory authentication source and also to let PacketFence know to automatically register any devices that successfully authenticate using 802.1X on the default connection profile.

From *Configuration* → *Policies and Access Control* → *Connection Profiles*, click on **New Connection Profile**. Specify the following information:

- Profile Name: 8021x
- Profile Description: 802.1X wired connections
- Enable profile: checked
- Automatically register devices: checked

- Filters: If any of the following conditions are met:
 - Connection Type: Ethernet-EAP
- Sources: your newly created Active Directory authentication source

Click on **Create** to save all configuration changes.

5.6. Configuring Microsoft Windows Supplicant

To enable 802.1X on the wired adapter of the Microsoft Windows 7 endpoint, you first need to enable the 'Wired AutoConfig' service. To do so, from the Microsoft Windows Services control panel, double-click on **Wired AutoConfig**. Make sure 'Startup type:' is set to 'Automatic' and click on **Start** to enable the service.

Then, from Windows' Network Connection panel, open the Properties window of the LAN interface you will use for testing. From the authentication tab, make sure 'Enable IEEE 802.1X authentication' is checked. As the authentication method, make sure 'Microsoft: Protected EAP (PEAP)' is selected. Then, click on **Settings** and make sure 'Validate server certificate' is unchecked. As authentication method, make sure 'Secured password (EAP-MSCHAPv2)' is selected. Then, click on **Configure ...** and make sure 'Automatically use my Windows logon name and password (and domain if any)' is unchecked.

Save all changes.

5.7. Testing

Now, we are ready to do some testing. First make sure you restart the 'radiusd' service. That is required since we added a new Active Directory domain controller. From *Status* → *Services*, click on the **Restart** button for the 'radiusd' service. PacketFence will take care of restarting that service and the 'radiusd-acct' and 'radiusd-auth' sub-services.

Connect the Microsoft Windows 7 endpoint on port no. 10 from the Cisco Catalyst 2960 switch. From Microsoft Windows, a popup should appear prompting you for a username and password. Enter a valid username and password from your Microsoft Active Directory domain - this should trigger 802.1X (EAP-PEAP) authentication.

To see what's going on from PacketFence, click on the *Auditing* tab from PacketFence's admin interface. You should see an entry for the MAC address of your Microsoft Windows 7 endpoint. Click on the line with the right MAC address to see the RADIUS exchanges. If the 802.1X authentication is successful, you should have 'Accept' as an 'Auth Status'.

5.8. Alerting

PacketFence can send emails to administrators, users and guests. So, it is important to properly configure the mail sending functionality of PacketFence. From *Configuration* → *System Configuration* → *Alerting*, set at least the following fields:

- Sender - the "From" address of emails being sent by PacketFence
- SMTP server - IP or DNS name of the SMTP server used by PacketFence to send all emails

If your SMTP server requires authentication or encryption to relay emails, you will have to properly configure the SMTP encryption, username and password parameters.

6. Enabling the Captive Portal

In the previous section, we have successfully configured 802.1X using PacketFence, Microsoft Active Directory and a Cisco Catalyst 2960 switch. While this demonstrates the fundamental role and capabilities of a NAC solution, most organizations are also looking at providing access to guests for example. One way of handling guests on a network is showing them a captive portal and let them register their own devices. This section will guide you in achieving this with PacketFence.

There are two ways PacketFence can show its captive portal for unknown (or unregistered) devices:

- it can use Web Authentication (or also known as hotspot-style authentication) - this works with numerous equipment vendors
- it can use a registration VLAN, where PacketFence provides DHCP services and DNS black-holing services - this works with any equipment vendors that support RADIUS dynamic VLAN assignment

For our example, we will use Web Authentication, as it is supported by the Cisco Catalyst 2960. For more information on various enforcement modes, please refer to the 'Supported Enforcement Modes' sections of this document.

6.1. Creating Authentication Source for Guests

To keep our example simple, we will simply create a captive portal for guests where they will only have to accept the terms and conditions prior to gaining network access. To do so, we must first create a 'Null' authentication source. From *Configuration* → *Policies and Access Control* → *Authentication Sources*, click on **New external source** **Null**. As 'Name' and 'Description', specify 'null-source'. Then add an 'Authentication Rules' with name 'catchall' with no condition and with the following tow 'Actions':

- Role - guest
- Access duration - 12 hours

Click on **Save** to save the new authentication source.

6.2. Configure switchport for Web Authentication

Connect to that switch over SSH as an admin.

First, we need to enable Change-of-Authorization (CoA) in our Cisco Catalyst 2960 switch configuration. We essentially need to allow our PacketFence server (172.20.100.2) to send CoA requests to the switch:

```
aaa server radius dynamic-author
  client 172.20.100.2 server-key useStrongerSecret
  port 3799
```

Then, we must enable Web Authentication on switch port no. 10. Add the following configuration to the global section:

```
ip device tracking
ip http server
ip http secure-server
```

Then add the required access list:

```
ip access-list extended registration
deny ip any host 172.20.100.2
permit tcp any any eq www
permit tcp any any eq 443
```

6.3. Adjust Switch Configuration in PacketFence

Next we have to let PacketFence know that Web Auth is to be used on the Cisco Catalyst 2960 switch. From *Configuration* → *Policies and Access Control* → *Switches* and click on your switch's IP to open its configuration options. From the 'Definition' tab, make sure 'Use CoA' and 'External Portal Enforcement' are checked and set the 'CoA Port' to 3799. From the 'Roles' tab, make the following changes:

- in Role by VLAN ID, set the registration and guest VLAN ID to 20 - this will ensure unregistered clients are initially put in VLAN 20 and avoid a VLAN change once they properly authenticate from the captive portal
- make sure 'Role by Switch Role' is checked and set the registration role to 'registration' - this will ensure the registration access list created in the previous section is returned for unregistered users. This will limit their access to the PacketFence captive portal
- make sure 'Role by Web Auth URL' is checked and set the 'registration' URL to 'http://172.20.100.2/Cisco::Catalyst_2960'

Click on **Save** to save all configuration changes.

6.4. Enabling Portal on Management Interface

By default the PacketFence's captive portal does not listen on the management interface. To change this, go in *Configuration* → *Network Configuration* → *Interfaces* and click on the logical name of your management interface to bring the configuration panel. In 'Additional listening daemon(s)' - make sure you add 'portal'.

You must then restart the following services from *Status* → *Services*:

- haproxy-portal

- httpd.portal
- iptables

6.5. Configuring the Connection Profile

For Web Authentication, we will create a new connection profile in PacketFence. That means the default connection profile will be used for 802.1X while the new connection profile will be used for Web Authentication and will be used to display a captive portal with our 'Null' authentication source. From *Configuration* → *Policies and Access Control* → *Connection Profiles* click on **New Profile**. Specify the following information:

- Profile Name: guest
- Filters: If any of the following conditions are met:
 - Connection Type: Ethernet-NoEAP
- Sources: null-source

Click on **Save** to save all configuration changes.

6.6. Testing

First make sure that the Microsoft Windows 7 endpoint is unplugged from the Cisco Catalyst 2960 switch. Then, make sure the endpoint is unregistered from PacketFence. To do this, from the *Nodes* configuration module, locate its MAC address and click on it. From the node property window, change the 'Status' to 'unregistered'.

Next, we need to disable 802.1X from the network configuration card from the Microsoft Windows 7 endpoint. We want to simulate here an authentication by MAC address, so we have to disable 802.1X to do this. From Windows' Network Connection connection panel, ask for the properties of the LAN interface you will use for testing. From the authentication tab, make sure 'Enable IEEE 802.1X authentication' is unchecked. Save all changes.

Next, connect the endpoint in the Cisco Catalyst 2960 switch. After a few second, open a web browser and try to open any website - say <http://packetfence.org>. You should now see the captive portal. You should only need to accept the terms and conditions for gaining network access.

7. Authentication Sources

PacketFence can authenticate users that register devices via the captive portal using various methods. Among the supported methods, there are:

- Active Directory
- Apache htpasswd file
- BlackHole
- Email
- External HTTP API
- Clickatell
- Facebook (OAuth 2)
- Github (OAuth 2)
- Google (OAuth 2)
- Kerberos
- Kickbox
- LDAP
- LinkedIn (OAuth 2)
- Null
- OpenID Connect (OAuth 2)
- RADIUS
- SMS
- Sponsored Email
- Twilio
- Windows Live (OAuth 2)
- Password of the day

and many others. Moreover, PacketFence can also authenticate users defined in its own internal SQL database. Authentication sources can be created from PacketFence administrative GUI - from the *Configuration* → *Policies and Access Control* → *Authentication Sources* section. Authentication sources, rules, conditions and actions are stored in the [/usr/local/pf/conf/authentication.conf](#) configuration file.

Each authentication sources you define will have a set of rules, conditions and actions.

Multiple authentication sources can be defined, and will be tested in the order specified (note that they can be reordered from the GUI by dragging them around). Each source can have multiple rules, which will also be tested in the order specified. Rules can also be reordered, just like sources. Finally, conditions can be defined for a rule to match certain criteria. If the criteria match (one or more), actions are then applied and rules testing stop, across all sources as this is a "first match wins" operation.

When no condition is defined, the rule will be considered as a catch-all. When a catch-all is defined, all actions will be applied for any users that match in the authentication source. Once a source is defined, it can be used from *Configuration → Policies and Access Control → Connection Profiles*. Each connection profile has a list of authentication sources to use.

In the previous section, you configured two authentication sources: Microsoft Active Directory and the Null sources. They were both catch-all sources.

7.1. Email Authentication for Guests

This section will show you how to allow guests to register endpoints using their email address. PacketFence sends a PIN code to the guest's email address. That code will then be required to complete the registration process.

7.1.1. Adding Email Authentication Source

From *Configuration → Policies and Access Control → Authentication Sources*, click **New external source Email**. As 'Name' and 'Description', specify 'email-source'.

Additional options available

- **email_activation_timeout** - This is the delay given to a guest who registered by email confirmation to log into his email and click the activation link.
- **allow_localdomain** - Accept self-registration from email address within the local domain
- **activation_domain** - Set this value if you want to change the hostname in the validation link. Changing this requires to restart haproxy to be fully effective.
- **allowed_domains** - A comma-separated list of domains that are allowed for email registration. Allowed domains are checked after banned domains.
- **banned_domains** - A comma-separated list of domains that are banned for email registration. Banned domains are checked before allowed domains.

Then add an 'Authentication Rules' with name 'catchall' with no condition and with the following two 'Actions':

- Role - guest
- Access duration - 12 hours

Click on **Create** to save the new authentication source.

7.1.2. Configuring the Connection Profile

Now let's add our new Email-based authentication source to our guests captive portal. From *Configuration → Policies and Access Control → Connection Profiles*, click on the **guest** profile that we previously created. In the 'Sources', click on the **(+)** button and add the newly created Email source, 'email-source'. Save the changes by clicking on the **Save** button.

NOTE

You can preview at any time the portal associated with connection profile by clicking on the **Preview** button near the Connexion's title.

7.1.3. Testing

Unplug and unregister your endpoint. Reconnect the endpoint - you should see the captive portal with the new Email-based registration option.

7.2. Adding SMS Authentication for Guests

This section will show you how to enable SMS authentication on the captive portal so that guests use their cellular phone number to register their endpoints. PacketFence will send an SMS PIN code to the guest phone number. That code will be required to complete the registration process. The SMS code will be sent by PacketFence over email - using popular SMTP-to-SMS gateways.

Some of the key concepts presented in this section are:

- Authentication sources

7.2.1. Adding SMS Authentication Source

Now that you understand what authentication sources and alerting are, we will add an SMS authentication source on our guest portal. We previously used the 'Null' source but we will add an other source. Portal profiles can provide multiple authentication sources.

From *Configuration* → *Policies and Access Control* → *Authentication Sources*, click **New external source** **SMS**. As 'Name' and 'Description', specify 'sms-source'. Then add an 'Authentication Rules' with name 'catchall' with no condition and with the following two 'Actions':

- Role - guest
- Access duration - 12 hours

You will also need to select the proper carriers to do your test. Make sure you include the one your are using for your cellular phone.

Click on **Create** to save the new authentication source.

Clickatell Source

To use Clickatell as an SMS source, first register at <https://www.clickatell.com> to get an API Key for the SMS integration. Then add it as an authentication source the same way as above, except choosing 'Clickatell' instead of 'SMS' in 'Add source → External'. Enter a name, description and your Clickatell API key in the source configuration, then add the authentication rule.

7.2.2. Configuring the Connection Profile

Now let's add our new SMS-based authentication source to our guests captive portal. From *Configuration* → *Policies and Access Control* → *Connection Profiles*, click on the 'guest' profile that we previously created. In the **Sources**, click on the **(+)** button and add the newly created SMS source, 'sms-source'. Save the changes by clicking on the **Save** button.

NOTE

You can preview at any time the portal associated with connection profile by clicking on the **Preview** button near the Connexion's title.

7.2.3. Testing

First unplug and unregister again the Microsoft Windows 7 endpoint. Then, connect the endpoint in switch port no. 10 - you should see the captive portal with the new SMS-based registration option. Note that the Null option will also be offered.

8. Introduction to Role-based Access Control

One important key concept from NAC solutions is for segregating network accesses. For example, an employee from the finance department might not have the same network access level as an other employee from the marketing department. Guests should also not have the same access level as normal employees within an organization. PacketFence uses roles internally to identify and differentiate users. For segregating network access, PacketFence can use one or all of the following techniques:

- ACL
- VLAN or VLAN pool
- equipment role

The techniques to use depends on the wired/WiFi equipment itself. A role in PacketFence will be eventually mapped to a VLAN, an ACL or an external role. You must define the roles to use in your organization for network access.

In our previous configuration examples, we made use of two roles that come by default in PacketFence: default and guest. We will now add two new roles - one for consultants and one used to authenticate machines on the network.

8.1. Adding Roles

Roles in PacketFence can be created from *Configuration* → *Policies and Access Control* → *Roles*. From this interface, you can also limit the number of devices users belonging to certain roles can register.

Roles are dynamically computed by PacketFence, based on the rules (ie., a set of conditions and actions) from authentication sources, using a first-match wins algorithm. Roles are then matched to VLAN or VLAN pool or internal roles or ACL on equipment from the *Configuration* → *Policies and Access Control* → *Switches* module. For a VLAN pool instead of defining a VLAN identifier, you can set a value like that: 20..23,27..30 - which means that the VLAN returned by PacketFence can be 20 to 23 and 27 to 30 (inclusively). There are three algorithms: one based on a hash of the username (default one), another one based on a round-robin (last registered device +1) and one that selects a VLAN randomly in the pool.

Configuration → *Policies and Access Control* → *Roles*, click on **New Role**. Provide the following information:

- Name: employee
- Description: Role used for employees
- Max nodes per user: 2

Redo the operation of the other role:

- Name: corporate_machine
- Description: Corporate owned machines

- Max nodes per user: 1

Let's say we have two roles: employee and corporate_machine (defined above).

Now, we want to assign roles to employees and their corporate machines using Active Directory (over LDAP), both using PacketFence's captive portal.

8.2. Using the Employee Role

From the *Configuration* → *Policies and Access Control* → *Authentication Sources*, we select **New internal source AD**. We provide the following information:

- **Name:** ad1
- **Description:** Active Directory for Employees
- **Host:** 192.168.1.2:389 without SSL/TLS
- **Base DN:** CN=Users,DC=acme,DC=local
- **Scope:** subtree
- **Username Attribute:** sAMAccountName
- **Bind DN:** CN=Administrator,CN=Users,DC=acme,DC=local
- **Password:** acme123

Then, we add an **Authentication rules** by clicking on the **Add rule** button and provide the following information:

- **Name:** employees
- **Description:** Rule for all employees
- Don't set any condition (as it's a catch-all rule)
- Set the following **actions:**
 - Role - employee
 - Access duration - 7 days

Test the connection and save everything. Using the newly defined source, any username that actually matches in the source (using the **sAMAccountName**) will have the employee role and a 7 days Access Duration.

8.3. Using the Corporate_Machine Role

If you would like to differentiate user authentication and machine authentication using Active Directory, one way to do it is by creating a second authentication sources, for machines:

- **Name:** ad2
- **Description:** Active Directory for Corporate Machines
- **Host:** 192.168.1.2:389 without SSL/TLS
- **Base DN:** CN=Computers,DC=acme,DC=local
- **Scope:** One-level
- **Username Attribute:** servicePrincipalName

- **Bind DN:** CN=Administrator,CN=Users,DC=acme,DC=local
- **Password:** acme123

Then, we add an 'Authentication rules':

- **Name:** machines
- **Description:** Rule for corporate machines
- Don't set any condition (as it's a catch-all rule)
- Set the following **actions:**
- Role - corporate_machine
- Access duration - 7 days

Using this configuration, employees can only connect corporate machines, not personal devices.

NOTE

When a rule is defined as a catch-all, it will always match if the username attribute matches the queried one. This applies for Active Directory, LDAP and Apache htpasswd file sources. Kerberos and RADIUS will act as true catch-all, and accept everything.

NOTE

If you want to use other LDAP attributes in your authentication source, add them in *Configuration* → *System Configuration* → *Main Configuration* → *Advanced* → *Custom LDAP attributes*. They will then be available in the rules you define.

9. Supported Enforcement Modes

Prior configuring PacketFence, you must chose an appropriate enforcement mode to be used by PacketFence with your networking equipment. The enforcement mode is the technique used to enforce registration and any subsequent access of devices on your network. PacketFence supports the following enforcement modes:

- Inline
- Out-of-band using SNMP or RADIUS
- Hostpot-style (or Web Auth)
- RADIUS only
- DNS

It is also possible to combine enforcement modes. For example, you could use the out-of-band mode on your wired switches, while using the inline mode on your old WiFi access points.

The following sections will explain these enforcement modes. It will also explain how to properly configure PacketFence to use each enforcement mode.

9.1. Technical Introduction to Inline Enforcement

9.1.1. Introduction

In many other NAC solutions, it is not possible to support unmanageable devices such as entry-level consumer switches or access-points. Using PacketFence, with the new inline mode, it can be use in-band for those devices. So in other words, PacketFence would become the gateway of that inline network, and NAT or route the traffic using IPTables/IPSet to the Internet (or to another section of the network). Let see how it works.

9.1.2. Device Configuration

No special configuration is needed on the unmanageable device. That's the beauty. You only need to ensure that the device is "talking" on the inline VLAN. At this point, all the traffic will be passing through PacketFence since it is the gateway for this VLAN.

9.1.3. Access Control

The access control relies entirely on IPTables/IPSet. When a user is not registered, and connects in the inline VLAN, PacketFence will give him an IP address. At this point, the user will be marked as unregistered in the ipset session, and all the Web traffic will be redirected to the captive portal and other traffic blocked. The user will have to register through the captive portal as in VLAN enforcement. When he registers, PacketFence changes the device's ipset session to allow the user's mac address to go through it.

9.1.4. Limitations

Inline enforcement, because of its nature, has several limitations that you must be aware of.

- Everyone behind an inline interface is on the same Layer 2 LAN
- Every packet of authorized users goes through the PacketFence server increasing the server's load considerably: Plan ahead for capacity
- Every packet of authorized users goes through the PacketFence server: it is a single point of failure for Internet access
- Ipset can store up to 65536 entries, so it is not possible to have an inline network class greater than a class B

This is why it is considered a poor man's way of doing access control. We have avoided it for a long time because of the above mentioned limitations. That said, being able to perform both inline and VLAN enforcement on the same server at the same time is a real advantage: it allows admins to maintain maximum security while they deploy new and more capable network hardware providing a clean migration path to VLAN enforcement.

9.2. Technical Introduction to Out-of-band Enforcement

9.2.1. Introduction

VLAN assignment is currently performed using several different techniques. These techniques are compatible one to another, but not on the same switch port. This means that you can use the more secure and modern techniques for your latest switches and another technique on the old switches that doesn't support latest techniques. As its name implies, VLAN assignment means that PacketFence is the server that assigns the VLAN to a device. This VLAN can be one of your VLANs or it can be a special VLAN where PacketFence presents the captive portal for authentication or remediation.

VLAN assignment effectively isolate your hosts at the OSI Layer2 meaning that it is the trickiest method to bypass and is the one which adapts best to your environment since it glues into your current VLAN assignment methodology.

9.2.2. VLAN assignment techniques

Wired: 802.1X + MAC Authentication

802.1X provides port-based authentication, which involves communications between a supplicant, authenticator (known as NAS), and authentication server (known as AAA). The supplicant is often software on a client device, such as a laptop, the authenticator is a wired Ethernet switch or wireless access point, and the authentication server is generally a RADIUS server.

The supplicant (i.e., client device) is not allowed access through the authenticator to the network until the supplicant's identity is authorized. With 802.1X port-based authentication, the supplicant provides credentials, such as user name / password or digital certificate, to the authenticator, and the authenticator forwards the credentials to the authentication server for verification. If the credentials are valid (in the authentication server database), the supplicant (client device) is allowed to access the network. The protocol for authentication is called Extensible Authentication Protocol (EAP) which have many variants. Both supplicant and authentication servers need to speak the same EAP protocol. Most popular EAP variant is PEAP-

MsCHAPv2 (supported by Windows / Mac OSX / Linux for authentication against AD).

In this context, PacketFence runs the authentication server (a FreeRADIUS instance) and will return the appropriate VLAN to the switch. A module that integrates in FreeRADIUS does a remote call to the PacketFence server to obtain that information. More and more devices have 802.1X supplicant which makes this approach more and more popular.

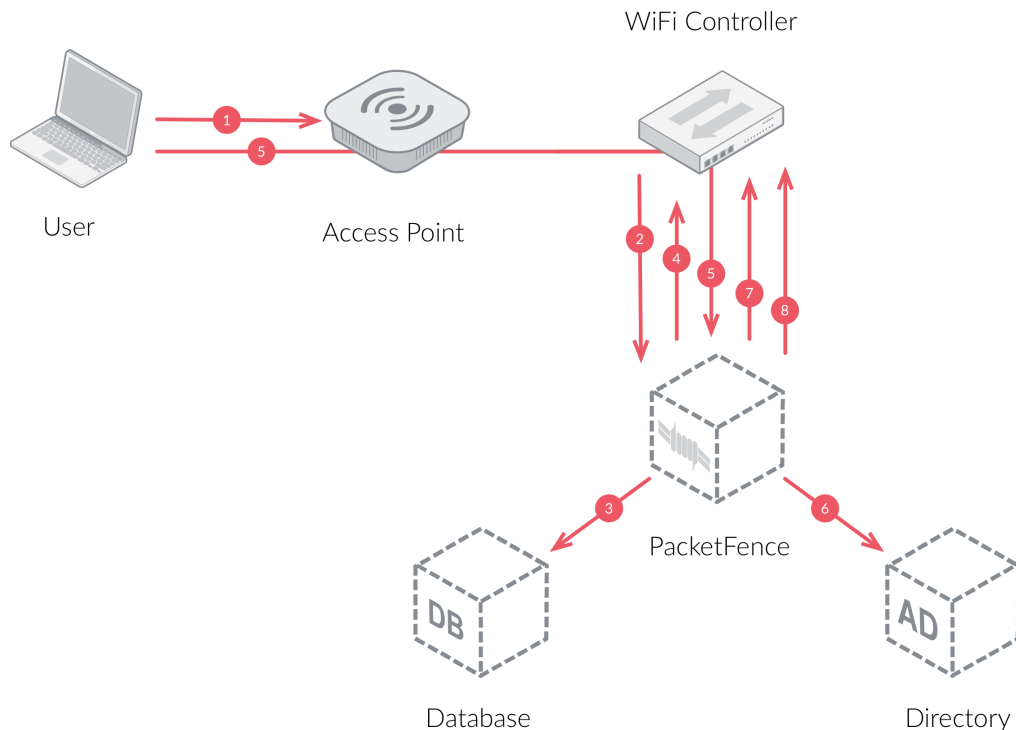
MAC Authentication is a new mechanism introduced by some switch vendor to handle the cases where a 802.1X supplicant does not exist. Different vendors have different names for it. Cisco calls it MAC Authentication Bypass (MAB), Juniper calls it MAC RADIUS, Extreme Networks calls it Netlogin, etc. After a timeout period, the switch will stop trying to perform 802.1X and will fallback to MAC Authentication. It has the advantage of using the same approach as 802.1X except that the MAC address is sent instead of the user name and there is no end-to-end EAP conversation (no strong authentication). Using MAC Authentication, devices like network printer or non-802.1X capable IP Phones can still gain access to the network and the right VLAN.

Wireless: 802.1X + MAC authentication

Wireless 802.1X works like wired 802.1X and MAC authentication is the same as wired MAC Authentication. Where things change is that the 802.1X is used to setup the security keys for encrypted communication (WPA2-Enterprise) while MAC authentication is only used to authorize (allow or disallow) a MAC on the wireless network.

On wireless networks, the usual PacketFence setup dictate that you configure two SSIDs: an open one and a secure one. The open one is used to help users configure the secure one properly and requires authentication over the captive portal (which runs in HTTPS).

The following diagram demonstrates the flow between a mobile endpoint, a WiFi access point, a WiFi controller and PacketFence:



1. User initiates association to WLAN AP and transmits MAC address. If user accesses network via a registered device in PacketFence, go to step 8.
2. The WLAN controller transmits MAC address via RADIUS to the PacketFence server to authenticate/authorize that MAC address on the AP.
3. PacketFence server conducts address audit in its database. If it does not recognize the MAC address, go to step 4. If it does, go to step 8.
4. PacketFence server directs WLAN controller via RADIUS (RFC2868 attributes) to put the device in an "unauthenticated role" (set of ACLs that would limit/redirect the user to the PacketFence captive portal for registration, or we can also use a registration VLAN in which PacketFence does DNS blackholing and is the DHCP server).
5. The user's device issues a DHCP/DNS request to PacketFence (which is a DHCP/DNS server on this VLAN or for this role) which sends the IP and DNS information. At this point, ACLs are limiting/redirecting the user to the PacketFence's captive portal for authentication. PacketFence fingerprints the device (user-agent attributes, DHCP information & MAC address patterns) to which it can take various actions including: keep device on registration portal, direct to alternate captive portal, auto-register the device, auto-block the device, etc. If the device remains on the registration portal the user registers by providing the information (username/password, cell phone number, etc.). At this time PacketFence could also require the device to go through a posture assessment (using Nessus, OpenVAS, etc.).
6. If authentication is required (username/password) through a login form, those credentials are validated via the Directory server (or any other authentication sources - like LDAP, SQL, RADIUS, SMS, Facebook, Google+, etc.) which provides user attributes to PacketFence which creates user+device policy profile in its database.
7. PacketFence performs a Change of Authorization (RFC3576) on the controller and the user must be re-authenticated/reauthorized, so we go back to step 1.
8. PacketFence server directs WLAN controller via RADIUS to put the device in an "authenticated role", or in the "normal" VLAN.

Web Authentication Mode

Web authentication is a method on the switch that forwards HTTP traffic of the device to the captive portal. With this mode, your device will never change of VLAN ID but only the ACL associated to your device will change. Refer to the Network Devices Configuration Guide to see a sample web auth configuration on a Cisco WLC.

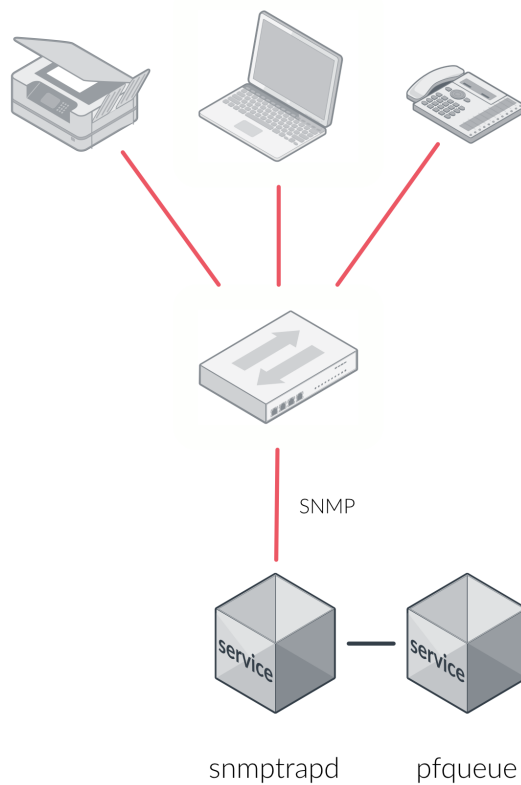
Port-security and SNMP

Relies on the port-security SNMP Traps. A fake static MAC address is assigned to all the ports this way any MAC address will generate a security violation and a trap will be sent to PacketFence. The system will authorize the MAC and set the port in the right VLAN. VoIP support is possible but tricky. It varies a lot depending on the switch vendor. Cisco is well supported but isolation of a PC behind an IP Phone leads to an interesting dilemma: either you shut the port (and the phone at the same time) or you change the data VLAN but the PC doesn't do DHCP (didn't detect link was down) so it cannot reach the captive portal.

Aside from the VoIP isolation dilemma, it is the technique that has proven to be reliable and that has the most switch vendor support.

9.2.3. More on SNMP traps VLAN isolation

When the VLAN isolation is working through SNMP traps all switch ports (on which VLAN isolation should be done) must be configured to send SNMP traps to the PacketFence host. On PacketFence, we use `snmptrapd` as the SNMP trap receiver. As it receives traps, it reformats and sends them into a redis queue, managed by `pfqueue` service. The multiprocessed `pfqueue` service reads these traps from the redis queue and takes a decision based on type of traps. For example, it can respond to them by setting the switch port to the correct VLAN. Currently, we support switches from Cisco, Edge-Core, HP, Intel, Linksys and Nortel (adding support for switches from another vendor implies extending the `pf::Switch` class). Depending on your switches capabilities, `pfqueue` will act on different types of SNMP traps.



You need to create a registration VLAN (with a DHCP server, but no routing to other VLANs) in which PacketFence will put unregistered devices. If you want to isolate computers which have open security event in a separate VLAN, an isolation VLAN needs also to be created.

Link Changes (deprecated)

When a host connects to a switch port, the switch sends a linkUp trap to PacketFence. Since it takes some time before the switch learns the MAC address of the newly connected device, PacketFence immediately puts the port in the Registration VLAN in which the device will send DHCP requests in order for the switch to learn its MAC address. Then `pfqueue` will send

periodical SNMP queries to the switch until the switch learns the MAC of the device. When the MAC address is known, `pfqueue` checks its status (existing ? registered ? any security event?) in the database and puts the port in the appropriate VLAN. When a device is unplugged, the switch sends a 'linkDown' trap to PacketFence which puts the port into the Registration VLAN.

When a computer boots, the initialization of the NIC generates several link status changes. And every time the switch sends a linkUp and a linkDown trap to PacketFence. Since PacketFence has to act on each of these traps, this generates unfortunately some unnecessary load on `pfqueue`. In order to optimize the trap treatment, PacketFence stops every thread for a 'linkUp trap' when it receives a 'linkDown' trap on the same port. But using only linkUp/linkDown traps is not the most scalable option. For example in case of power failure, if hundreds of computers boot at the same time, PacketFence would receive a lot of traps almost instantly and this could result in network connection latency.

MAC Notification Traps (deprecated)

If your switches support MAC notification traps (MAC learned, MAC removed), we suggest that you activate them in addition to the linkUp/linkDown traps. This way, `pfqueue` does not need, after a linkUp trap, to query the switch continuously until the MAC has finally been learned. When it receives a linkUp trap for a port on which MAC notification traps are also enabled, it only needs to put the port in the Registration VLAN and can then free the process. When the switch learns the MAC address of the device it sends a MAC learned trap (containing the MAC address) to PacketFence.

Port Security Traps

In its most basic form, the Port Security feature remembers the MAC address connected to the switch port and allows only that MAC address to communicate on that port. If any other MAC address tries to communicate through the port, port security will not allow it and send a port-security trap.

If your switches support this feature, **we strongly recommend to use it rather than linkUp/linkDown and/or MAC notifications**. Why? Because as long as a MAC address is authorized on a port and is the only one connected, the switch will send no trap whether the device reboots, plugs in or unplugs. This drastically reduces the SNMP interactions between the switches and PacketFence.

When you enable port security traps you should not enable linkUp/linkDown nor MAC notification traps.

9.3. Technical Introduction to Hybrid Enforcement

9.3.1. Introduction

In previous versions of PacketFence, it was not possible to have RADIUS enabled for inline enforcement mode. Now with the new hybrid mode, all the devices that supports 802.1X or MAC-authentication can work with this mode. Let's see how it works.

9.3.2. Device Configuration

You need to configure inline enforcement mode in PacketFence and configure your switch(es) / access point(s) to use the VLAN assignment techniques (802.1X or MAC-authentication). You also need to take care of a specific parameter in the switch configuration window, "Trigger to enable

inline mode". This parameter is working like a trigger and you have the possibility to define different sort of triggers:

ALWAYS

PORT

MAC

SSID

where ALWAYS means that the device is always in inline mode, PORT specify the ifIndex of the port which will use inline enforcement, MAC a mac address that will be put in inline enforcement technique rather than VLAN enforcement and SSID an ssid name. An example:

```
SSID::GuestAccess,MAC::00:11:22:33:44:55
```

This will trigger all the nodes that connects to the *GuestAccess* SSID to use inline enforcement mode (PacketFence will return a void VLAN or the **inlineVlan** if defined in switch configuration) and the MAC address **00:11:22:33:44:55** client if it connects on another SSID.

9.4. Technical Introduction to RADIUS Enforcement

9.4.1. Introduction

The concept of having a RADIUS enforcement is to not use registration, isolation, nor the portal capabilities of PacketFence. Everything here is for RADIUS integration only. By default the management interface will be the RADIUS interface. If needed, it is possible to add an other interface from *Configuration* → *Network Configuration* → *Networks* → *Interface*. When doing so, you must select **Other** as the type of interface. Moreover, you must select **radius** as an additionnal listening daemon.

Using RADIUS enforcement, everytime a device connects to the network, a matching production VLAN will be assigned, depending on the rules in *Configuration* → *Policies and Access Control* → *Authentication Sources*.

9.5. Technical Introduction to DNS Enforcement

9.5.1. Introduction

DNS enforcement allows you to control the network access of the device by using the **pfdns** service on PacketFence.

The architecture of DNS enforcement is as following :

- DHCP and DNS are provided by the PacketFence server
 - The PacketFence DHCP server will provide the IP of your network equipment as the gateway and the IP address of the PacketFence DNS server to resolve names.
- Routing is provided by another equipment on your network (core switch, firewall, router,...)
- **pfdns** will respond to DNS requests depending on your configuration :
 - user registration on portal : it will return IP address of the captive portal
 - access to another site : it will resolve name externally and use it in reply

This enforcement mode used by itself can be bypassed by the device by using a different DNS server or by using its own DNS cache.

The first can be prevented using an ACL on your routing equipment, the second can be prevented by combining DNS enforcement with Single-Sign-On on your network equipment. Please see the Firewall Single-Sign-On documentation for details on how to accomplish this.

In order to configure DNS enforcement, you first need to go in *Configuration* → *Network Configuration* → *Networks* → *Interface* then select one of your interfaces and set it in DNS enforcement mode.

After, you need to configure a routed network for this interface by clicking **New routed network**. See the 'Routed Networks' section of this document for details on how to configure it.

NOTE

If you are not using a routed network, you need to use Inline enforcement as DNS enforcement can only be used for routed networks.

Once this is done, you need to restart the **pfdhcp** and **pfdns** services.

10. Adding Inline Enforcement to Existing Installation

10.1. Introduction

The inline enforcement is a very convenient method for performing access control on older network equipment that is not capable of doing VLAN enforcement or that is not supported by PacketFence.

An important configuration parameter to have in mind when configuring inline enforcement is that the DNS reached by these users should be your actual production DNS server - which shouldn't be in the same broadcast domain as your inline users. The next section shows you how to configure the proper inline interface and it is in this section that you should refer to the proper production DNS.

Inline enforcement uses `ipset` to mark nodes as registered, unregistered and isolated. It is also now possible to use multiple inline interfaces. A node registered on the first inline interface is marked with an IP:MAC tuple (for L2, only ip for L3), so when the node tries to register on an other inline interface, PacketFence detects that the node is already registered on the first inline network. It is also possible to enable `inline.should_reauth_on_vlan_change` to force users to reauthenticate when they change inline network - you can change this from 'Configuration→Network Configuration→Inline' - by checking or not the 'Reauthenticate node' checkbox.

By default the inline traffic is forwarded through the management network interface but it is possible to specify another one by adding in `pf.conf` the option `interfaceSNAT` in inline section of the `pf.conf` configuration file. Alternatively, you can change this from 'Configuration→Network Configuration→Inline' in the 'SNAT Interface' section. It is a comma delimited list of network interfaces like `eth0,eth1.2`. It's also possible to specify a network that will be routed instead of using NAT by adding in `conf/networks.conf` an option `nat=no` under one or more network sections (take care of the routing table of the PacketFence server).

10.2. Preparing the Operating System

In order to build an inline deployment of PacketFence setup you need :

- 2 network interfaces for the VM (1 for the Inline and another one to go out)
- a switch port in the management network for the PacketFence server
- a switch port in the inline network for the PacketFence server which needs to be configured in access mode and in the same access VLAN as every switchport on which devices will be connected

10.3. Adding Inline Interface

PacketFence can be configured right from the start using the PacketFence configurator for inline

enforcement. In this example, we will continue building on top of our initial deployment by adding a new inline interface to our PacketFence installation.

The first step is to add a dedicated Network Interface Card (NIC) to your current PacketFence installation. In our example, our new NIC will be named `ens192`. The PacketFence web interface will list all currently installed network interfaces on the system. An IP and a netmask will be visible if the network interface is configured (either by DHCP or already manually configured). You can edit those ones, create/delete VLANs on physical interfaces and enable/disable interfaces. Note that these changes are effective immediately. Persistence will be written only for **enabled** interfaces. Which means that if you change your management IP address, to pursue the configurator, you will need to go on this new IP address you just set. At all time, you will need to set a Management interface. That means that the required interface types for inline enforcement are:

```
Management
Inline layer 2
```

Note that PacketFence will provide these services on its inline interface:

- PacketFence provides its own DHCP service. It will take care of IP address distribution in our Inline network. PacketFence will not provide DHCP services on the management network - this is the responsibility of your own infrastructure.
- PacketFence provides its own DNS service. However, for the inline mode, you will also need to provide access to the DNS server of your infrastructure.

From 'Configuration→Network Configuration→Interfaces', click on the `ens192` logical name. Provide the following information:

```
IP Address: 192.168.2.1
Netmask: 255.255.255.0
Type: Inline Layer 2
Additional listening daemon(s): portal
DNS Servers: 10.0.0.10
```

Click on 'Save' and toggle the new interface to 'On'.

Once done, your PacketFence server should have the following network layout:

Please refer to the following table for IP and subnet information :

Networ k Card	Name	Subnet	Gateway	PacketFence Address
ens160	Management	172.20.100.0/16	172.20.0.1	172.20.100.2
ens192	Inline	192.168.2.0/24	192.168.2.1	192.168.2.1

Finally, from *Status→Services*, restart the `haproxy-portal`, `pfdhcp`, `iptables`, `pfdhcplistener`, `pfdns` services.

10.4. Network Devices

In an inline configuration, the required configurations for network devices (desktops, tablets, printers, etc.) will be to make sure they can all communicate with PacketFence. In other words for a switch you will need to configure every ports on which devices will be connected using the access mode with all of them in the same inline network. Access point will be connected as device to be in the inline subnetwork.

Example with a Cisco switch:

You should be in mode '#conf-t' if not execute 'configuration terminal' in your CLI.

```
interface range [port-range]
switchport mode access vlan 1
no shutdown
interface [packetfence_ens192]
switchport mode access vlan 1
no shutdown
end
copy running-configuration startup-configuration
```

Now you can connect any devices that you want to be in the inline network in any of the port you have just configured.

10.5. Adding Connection Profile for Inline

Next thing we do is to add a new connection profile - for devices coming from the inline network. We want to show users the captive portal with our Null authentication sources.

From 'Configuration→Policies and Access Control→Connection Profiles', click on 'Add Profile'. Provide the following information:

- Profile Name: inline
- Filters: If **any** Network 192.168.2.0/24
- Sources: null-source

Then click on 'Save'.

10.6. Testing the Inline Configuration

You can now test the registration process. In order to do so:

- connect an unregistered device into the switch
- make sure PacketFence provides an IP address to the device. Look into the following log file: [/usr/local/pf/logs/packetfence.log](#) or verify on the computer you obtain an IP in the right subnet range

From the computer:

- open a web browser

- try to connect to a HTTP site (Not HTTPS, eg. <http://www.packetfence.org>)
- make sure that whatever site you want to connect to, you have only access to the registration page.

Register the computer using using the Null authentication source.

Once a computer has been registered:

- make sure PacketFence changes the firewall (`ipset -L`) rules so that the user is authorized through. Look into PacketFence log file: `/usr/local/pf/logs/packetfence.log`
- from the web administrative interface, go under Nodes and make sure you see the computer as 'Registered'.
- the computer has access to the network and the Internet.

10.7. Advanced Inline Topics

10.7.1. Traffic Shaping

It's possible to enable traffic shaping based on the role of the device. In order to enable it you need to go in 'Configuration → Network Configuration → Inline Traffic Shaping' and select the role you want to define a limit. Set a upload and download speed limit and save.

Next restart the `tc` service to apply the new rules.

11. Adding VLAN Enforcement to Existing Installation

11.1. Introduction

In order to build a VLAN isolation setup you need :

- a supported switch (please consult the list of supported switch vendors and types in the *Network Devices Configuration Guide* including information on uplinks)
- a normal, registration and isolation VLAN (VLAN numbers and subnets)
- a switch port for the PacketFence (PacketFence) server which needs to be configured as a dot1q trunk (several VLANs on the port)

Throughout this configuration example we use the following assumptions for our network infrastructure:

- VLAN 20 is the management VLAN
- VLAN 102 is the registration VLAN (unregistered devices will be put in this VLAN)
- VLAN 103 is the isolation VLAN (isolated devices will be put in this VLAN)
- VLAN 104 is the normal VLAN (registered devices will be put in this VLAN)

Please refer to the following table for IP and Subnet information :

VLAN ID	VLAN Name	Subnet	Gateway
20	Management	172.20.100.0/16	172.20.0.1
102	Registration	192.168.102.0/24	192.168.102.1
103	Isolation	192.168.103.0/24	192.168.103.1
104	Normal	10.0.104.0/24	10.0.104.1

VLAN ID	VLAN Name	PacketFence Address	DHCP	DNS
20	Management	172.20.100.2	infrastructure DHCP server	infrastructure DNS server
102	Registration	192.168.102.1	PF	PF
103	Isolation	192.168.103.1	PF	PF
104	Normal		infrastructure DHCP server	infrastructure DNS server

Note that PacketFence will provide these services on its registration and isolation VLANs:

- PacketFence provides its own DHCP services. It will take care of IP address distribution in VLANs 102 and 103. PacketFence will not provide DHCP services on VLAN 104 - this is the responsibility of your own infrastructure
- PacketFence provides its own DNS service. It will take care of naming resolution in VLANs 102 and 103. PacketFence will not provide DNS services on VLAN 104 - this is the responsibility of your own infrastructure

11.2. Adding the Registration, Isolation and Other Interface

First of all, make sure you add a new NIC to your PacketFence server and you set the switch port where that NIC is connected in **trunk**. If you prefer, you can also set your management interface as trunk and set the PVID to your management VLAN on the switch port where that management is connected.

We will create three interfaces VLAN for registration, isolation and normal using the management interface.

The required interface types for VLAN enforcement are:

- Management
- Registration
- Isolation
- Other

Note that you can only set **one** (1) management interface.

In our example, we will create three new VLANs on the wired interface on our new trunk interface (**ens224**) To do so, click the 'Add VLAN' button besides the wired interface for each of the needed VLAN:

Here's a sample configuration for both of them:

Registration

```
Virtual LAN ID: 102
IP Address: 192.168.102.1
Netmask: 255.255.255.0
```

Isolation

```
Virtual LAN ID: 103
IP Address: 192.168.103.1
Netmask: 255.255.255.0
```

Normal

```
Virtual LAN ID: 104
```

NOTE

Ignore the High-Availability options for now. If you are interested in a PacketFence cluster, please refer to the PacketFence Clustering Guide.

According to our example, we'll associate the correct type the each interfaces.

```
ens160: Management
ens224 VLAN 102: Registration
ens224 VLAN 103: Isolation
ens224 VLAN 104: Other
```

Make sure that those three interfaces are in an **enabled** state for the persistence to occur. We also need to set the Default Gateway which will generally be the gateway of the management network.

Finally, from *Status*→*Services*, restart the `haproxy-portal`, `pfdhcp`, `iptables`, `pfdhcplistener`, `pfdns` services.

11.3. Network Devices

Now let's modify our switch configuration to enable our new registration and isolation VLANs. From 'Configuration→Policies and Access Control→Switches', click on our Cisco 2960 switch we added earlier (172.21.2.3).

From the Roles tab, make sure you specify the following information:

```
Role by VLAN ID: checked
registration VLAN: 102
isolation VLAN: 103
default: 104
guest: 104
```

Disable 'Role by Switch Role' and 'Role by Web Auth URL'.

Click on the 'Save' button once completed.

11.3.1. Configure the Cisco Catalyst 2960

In previous sections, we correctly configured our switch to do 802.1X. Now let's slightly modify that configuration so that we enable MAC authentication and 802.1X on a new switch port. This will demonstrate the configuration differences.

11.3.2. Configure Switchport for MAB

Once AAA is ready, we can configure some or all switchports to perform MAB (MAC Authentication Bypass) and 802.1X. In our example, we will only configure port no. 11 without VoIP support:

```
switchport mode access
authentication host-mode single-host
authentication order mab dot1x
authentication priority mab dot1x
authentication port-control auto
authentication periodic
authentication timer restart 10800
authentication timer reauthenticate 10800
mab
no snmp trap link-status
dot1x pae authenticator
dot1x timeout quiet-period 2
dot1x timeout tx-period 3
```

If you want to test some ports with a VoIP phone (ex: Voice VLAN 200), add the following lines to your interface configuration:

```
switchport voice vlan 200
authentication host-mode multi-domain
```

11.3.3. Configure SNMP

Finally, for some operations (like VoIP), PacketFence still need to have SNMP access to the switch. Make sure you configure the two SNMP communities like:

```
snmp-server community ciscoRead ro
snmp-server community ciscoWrite rw
```

NOTE | You can refer to the [Cisco Catalyst documentation](#) for more options.

11.3.4. Save the Configuration

When done, don't forget to save your configuration changes using the `write mem` command.

11.4. Adding Connection Profile for Registration

Next thing we do is to add a new connection profile - for devices coming from the registration network. We want to show users the captive portal with our Null authentication sources.

From 'Configuration→Policies and Access Control→Connection Profiles', click on 'Add Profile'. Provide the following information:

- Profile Name: registration
- Filters: If **any** VLAN 102
- Sources: null-source

Then click on 'Save'.

11.4.1. Testing VLAN Based Enforcement

You can now test the registration process. In order to do so:

- connect an unregistered device into the switch
- make sure PacketFence receives the radius authentication request from the switch. Look into the PacketFence log file: `/usr/local/pf/logs/packetfence.log`
- make sure PacketFence handles RADIUS requests and sets the switch port to the registration VLAN (VLAN 102). Look again into PacketFence log file: `/usr/local/pf/logs/packetfence.log`

On the computer:

- open a web browser
- try to connect to a HTTP site (Not HTTPS, eg. <http://www.packetfence.org>)
- make sure that whatever site you want to connect to, you have only access to the registration page.

Register the computer using the Null authentication source.

Once a computer has been registered, make sure:

- PacketFence puts the switch port into the normal VLAN (VLAN 104)
- The computer has access to the network and the Internet.

12. Troubleshooting PacketFence

12.1. RADIUS Audit Log

PacketFence provides a RADIUS auditing module which allows you to be aware of all the incoming RADIUS requests/responses handled by PacketFence. The RADIUS auditing module is available from *Auditing* → *RADIUS Audit Log*. Advanced search criterias can be specified to create complex search expressions - which can be saved for later use. Clicking on a RADIUS log entry will display the endpoint information, where the RADIUS request originated from and the RADIUS payload exchanged between the NAS and PacketFence.

12.2. Log files

Here are the most important PacketFence log files:

- `/usr/local/pf/logs/packetfence.log` – PacketFence Core Log
- `/usr/local/pf/logs/httpd.portal.access` – Apache – Captive Portal Access Log
- `/usr/local/pf/logs/httpd.portal.error` – Apache – Captive Portal Error Log
- `/usr/local/pf/logs/httpd.admin.access` – Apache – Web Admin/Services Access Log
- `/usr/local/pf/logs/httpd.admin.error` – Apache – Web Admin/Services Error Log
- `/usr/local/pf/logs/httpd.webservices.access` – Apache – Webservices Access Log
- `/usr/local/pf/logs/httpd.webservices.error` – Apache – Webservices Error Log
- `/usr/local/pf/logs/httpd.aaa.access` – Apache – AAA Access Log
- `/usr/local/pf/logs/httpd.aaa.error` – Apache – AAA Error Log

There are other log files in `/usr/local/pf/logs/` that could be relevant depending on what issue you are experiencing. Make sure you take a look at them.

The main logging configuration file is `/usr/local/pf/conf/log.conf`. It contains the configuration for the `packetfence.log` file (`Log::Log4Perl`) and you normally don't need to modify it. The logging configuration files for every service are located under `/usr/local/pf/conf/log.conf.d/`.

12.3. RADIUS Debugging

First, check the FreeRADIUS logs. The file is located at `/usr/local/pf/logs/radius.log`.

If this didn't help, run FreeRADIUS in debug mode. To do so, start it using the following commands.

For the authentication radius process:


```
radiusd -X -d /usr/local/pf/radddb -n auth
```

For the accounting radius process:

```
radiusd -X -d /usr/local/pf/radddb -n acct
```

Additionally there is a `raddebug` tool that can extract debug logs from a running FreeRADIUS daemon. PacketFence's FreeRADIUS is pre-configured with such support.

In order to have an output from `raddebug`, you need to either:

1. Make sure user `pf` has a shell in `/etc/passwd`, add `/usr/sbin` to `PATH` (`export PATH=/usr/sbin:$PATH`) and execute `raddebug` as `pf`
2. Run `raddebug` as root (less secure!)

Now you can run `raddebug` easily:

```
raddebug -t 300 -f /usr/local/pf/var/run/radiusd.sock
```

The above will output FreeRADIUS' authentication debug logs for 5 minutes.

Use the following to debug radius accounting:

```
raddebug -t 300 -f /usr/local/pf/var/run/radiusd-acct.sock
```

See `man raddebug` for all the options.

13. Authentication Mechanisms

This section details most of the authentication mechanisms supported by PacketFence. It walks you through the required steps to properly use an authentication mechanism on your captive portal, for example. For Public Key Infrastructure (PKI) integration, please refer to the PKI Integration section from this document.

13.1. Microsoft Active Directory (AD)

Go in the Administration interface under *Configuration* → *Policies and Access Control* → *Domains* → *Active Directory Domains*.

NOTE

If you can't access this section and you have previously configured your server to bind to a domain externally to PacketFence, make sure you run `/usr/local/pf/addons/AD/migrate.pl`

Click **New Domain** and fill in the information about your domain.

The screenshot shows the 'New Domain' configuration interface. The left sidebar contains navigation options like 'Policies and Access Control', 'Roles', 'Domains', 'Realms', 'Authentication Sources', 'Network Devices', 'Switches', 'Switch Groups', 'Connection Profiles', 'Compliance', 'Integration', 'Advanced Access Configuration', 'Network Configuration', and 'System Configuration'. The main content area is titled 'New Domain' and has tabs for 'Settings' and 'NTLM cache'. The 'Settings' tab is active, showing the following fields:

- Identifier:** mydomain. Description: Specify a unique identifier for your configuration. This doesn't have to be related to your domain.
- Workgroup:** DOMAIN
- DNS name of the domain:** DOMAIN.NET. Description: The DNS name (FQDN) of the domain.
- This server's name:** packetfence. Description: This server's name (account name) in your Active Directory. Use '%h' to automatically use this server hostname.
- Sticky DC:** adserver.domain.net. Description: This is used to specify a sticky domain controller to connect to. If not specified, default '*' will be used to connect to any available domain controller.
- Active Directory server:** 192.168.1.20. Description: The IP address or DNS name of your Active Directory server.
- DNS server(s):** 192.168.1.20. Description: The IP address(es) of the DNS server(s) for this domain. Comma delimited if multiple.
- OU:** Computers. Description: Use a specific OU for the PacketFence account. The OU string read from top to bottom without RDNs and delimited by a '?'. E.g. "Computers/Servers/Unix". IMPORTANT NOTE: Due to a bug in the current version of samba, you will need to precreate a computer object in the OU you specify above when you're not using the default value ("Computers"). Otherwise you will get the following error: "Failed to join domain: failed to precreate account in ou ou=XYZ,dc=ACME,dc=CORP: No such object"
- ntlmv2 only:** Disabled (toggle).
- Allow on registration:** Disabled (toggle).

A yellow note box at the bottom states: "Note 'Allow on registration' option requires passthroughs to be enabled as well as configured to allow both the domain DNS name and each domain controllers DNS name (or *.dns name) Example: inverse.local, *.inverse.local". At the bottom of the form are 'Create & Join' and 'Reset' buttons.

Where :

- **Identifier** is a unique identifier for your domain. It's purpose is only visual.
- **Workgroup** is the workgroup of your domain in the old syntax (like NT4).
- **DNS name of the domain** is the FQDN of your domain. The one that suffixes your account

names.

- **This server's name** is the name that the server's account will have in your Active Directory.
- **Sticky DC** is the preferred domain controller to connect to.
- **Active Directory server** is the IP or DNS name of one of the DC of the domain.
- **DNS server** is the IP address of the DNS server of this domain. Make sure that the server you put there has the proper DNS entries for this domain.
- **OU** is the OU in the Active Directory where you want to create your computer account.
- **ntlmv2 only** forces the NTLNM authentication (802.1X on AD) to use the NTLM version 2.
- **Allow on registration** would allow devices in the registration network to communicate with the DC.

NOTE

If you are using an Active/Active cluster, each member of the cluster must be joined separately. Please follow the instructions in the PacketFence Clustering Guide.

13.1.1. Troubleshooting

- In order to troubleshoot unsuccessful binds, please refer to the following file : `/chroots/<mydomain>/var/log/samba<mydomain>/log.winbindd`. Replace `<mydomain>` with the identifier you set in the domain configuration.
- You can validate the domain bind using the following command : `chroot /chroots/<mydomain> wbinfo -u`
- You can test the authentication process using the following command `chroot /chroots/<mydomain> ntlm_auth --username=administrator`

NOTE

Under certain conditions, the test join may show as unsuccessful in the Administration interface but the authentication process will still work properly. Try the test above before doing any additional troubleshooting. Also try reloading the page in the GUI since in some case the browser side of the ajax call may time out while the join actually succeeds.

13.1.2. Default Domain Configuration

You should now define the domain you want to use as the default one by creating the following realm in *Configuration* → *Policies and Access Control* → *Domains* → *REALMS*.

Status Reports Auditing Nodes Users Configuration API dashboard

Filter

- Policies and Access Control
 - Roles
 - Domains
 - Active Directory Domains
 - Realms
 - Authentication Sources
 - Network Devices
 - Switches
 - Switch Groups
 - Connection Profiles
- Compliance
- Integration
 - Advanced Access Configuration
- Network Configuration
- System Configuration

Realm DEFAULT

Realm: DEFAULT

NTLM Auth Configuration

Domain: mydomain
The domain to use for the authentication in that realm.

Freeradius Proxy Configuration

Realm Options: strip
You can add FreeRADIUS options in the realm definition.

RADIUS AUTH:
The RADIUS Server(s) to proxy authentication.

Type: Keyed Balance
Home server pool type.

Authorize from PacketFence
Should we forward the request to PacketFence to have a dynamic answer or do we use the remote proxy server answered attributes?

RADIUS ACCT:
The RADIUS Server(s) to proxy accounting.

Type: Load Balance
Home server pool type.

Freeradius Eduroam Proxy Configuration

Eduroam Realm Options:
You can add Eduroam FreeRADIUS options in the realm definition.

Eduroam RADIUS AUTH:
The RADIUS Server(s) to proxy authentication.

Type: Keyed Balance
Home server pool type.

Authorize from PacketFence
Should we forward the request to PacketFence to have a dynamic answer or do we use the remote proxy server answered attributes?

Eduroam RADIUS ACCT:
The RADIUS Server(s) to proxy accounting.

Type: Load Balance
Home server pool type.

Stripping Configuration

Strip on the portal
Should the usernames matching this realm be stripped when used on the captive portal.

Strip on the admin
Should the usernames matching this realm be stripped when used on the administration interface.

Strip in RADIUS authorization
Should the usernames matching this realm be stripped when used in the authorization phase of 802.1x.
Note that this doesn't control the stripping in FreeRADIUS, use the options above for that.

Custom attributes
Allow to use custom attributes to authenticate 802.1x users (attributes are defined in the source).

LDAP source:
The LDAP Server to query the custom attributes.

Save Reset Clone Delete

Next, restart PacketFence in *Status → Services*

13.1.3. Multiple Domains Authentication

First configure your domains in *Configuration → Policies and Access Control → Domains → Active Directory Domains*.

Once they are configured, go in *Configuration → Policies and Access Control → Domains → REALMS*.

Create a new realm that matches the DNS name of your domain **AND** one that matches your workgroup. In the case of this example, it will be DOMAIN.NET tied to mydomain.

[Status](#)
[Reports](#)
[Auditing](#)
[Nodes](#)
[Users](#)
[Configuration](#)

[API](#)
[dashboard](#)

Filter

- Policies and Access Control**
 - Roles
 - Domains
 - Active Directory Domains
 - Realms
 - Authentication Sources
 - Network Devices
 - Switches
 - Switch Groups
 - Connection Profiles
- Compliance**
- Integration**
 - Advanced Access Configuration
- Network Configuration**
- System Configuration**

New Realm

Realm: DOMAIN.NET

NTLM Auth Configuration

Domain: mydomain
The domain to use for the authentication in that realm.

Freeradius Proxy Configuration

Realm Options: strip
You can add FreeRADIUS options in the realm definition.

RADIUS AUTH:
The RADIUS Server(s) to proxy authentication.

Type: Keyed Balance
Home server pool type.

Authorize from PacketFence
Should we forward the request to PacketFence to have a dynamic answer or do we use the remote proxy server answered attributes?

RADIUS ACCT:
The RADIUS Server(s) to proxy accounting.

Type: Load Balance
Home server pool type.

Freeradius Eduroam Proxy Configuration

Eduroam Realm Options:
You can add Eduroam FreeRADIUS options in the realm definition.

Eduroam RADIUS AUTH:
The RADIUS Server(s) to proxy authentication.

Type: Keyed Balance
Home server pool type.

Authorize from PacketFence
Should we forward the request to PacketFence to have a dynamic answer or do we use the remote proxy server answered attributes?

Eduroam RADIUS ACCT:
The RADIUS Server(s) to proxy accounting.

Type: Load Balance
Home server pool type.

Stripping Configuration

Strip on the portal
Should the usernames matching this realm be stripped when used on the captive portal.

Strip on the admin
Should the usernames matching this realm be stripped when used on the administration interface.

Strip in RADIUS authorization
Should the usernames matching this realm be stripped when used in the authorization phase of 802.1x.
Note that this doesn't control the stripping in FreeRADIUS, use the options above for that.

Custom attributes
Allow to use custom attributes to authenticate 802.1x users (attributes are defined in the source).

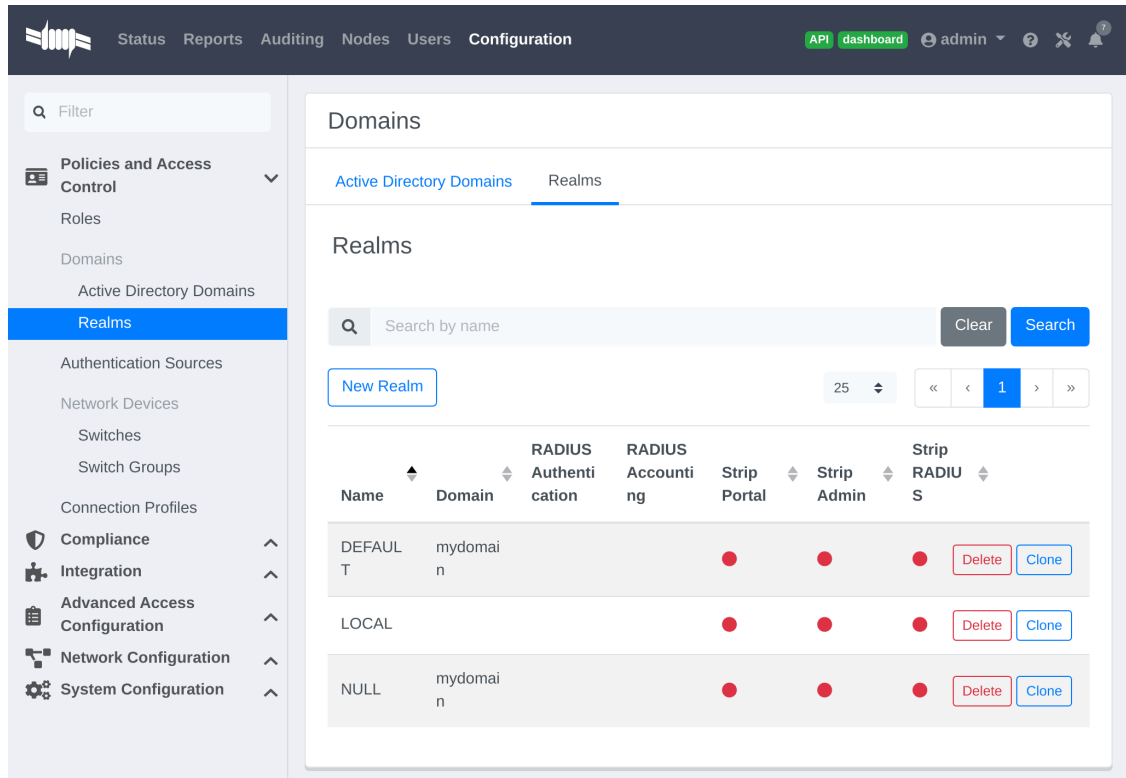
LDAP source:
The LDAP Server to query the custom attributes.

Where :

- **Realm** is either the DNS name (FQDN) of your domain or the workgroup
- **Domain** is the Active Directory domain where PacketFence sends the NTLM request
- **Realm options** are any realm options that you want to add to the FreeRADIUS configuration
- **Domain** is the domain which is associated to this realm
- **RADIUS Auth** is the RADIUS authentication server to proxy the request to
- **Type** is the home server pool type
- **Authorize from PacketFence** specifies if we forward the request to PacketFence to have a dynamic answer or do we use the remote proxy server answered attributes
- **RADIUS Acct** is the RADIUS accounting server to proxy the request to
- **Type** is the home server pool type
- **Eduroam Realm Options** You can add Eduroam FreeRADIUS options in the realm definition
- **Eduroam RADIUS Auth** is the RADIUS Eduroam authentication server to proxy the request to
- **Type** is the home server pool type
- **Authorize from PacketFence** specifies if we forward the request to PacketFence to have a dynamic answer or do we use the remote proxy server answered attributes
- **Eduroam RADIUS Acct** is the RADIUS Eduroam accounting server to proxy the request to
- **Type** is the home server pool type
- **Strip on the portal** Should the usernames matching this realm be stripped when used on the captive portal
- **Strip on the admin** Should the usernames matching this realm be stripped when used on the administration interface
- **Strip in RADIUS authorization** Should the usernames matching this realm be stripped when used in the authorization phase of 802.1X
- **Custom attributes** Allow to use custom attributes to authenticate 802.1X users (attributes are defined in the source)
- **LDAP source** The LDAP Server to query the custom attributes

Now associate **DEFAULT** and **NULL** realms to your domain.

You should now have the following realm configuration



13.2. OAuth2 Authentication

NOTE | OAuth2 authentication does not work with Webauth enforcement

NOTE | OAuth2 authentication will fail by design when previewed through "Connection Profiles"

The captive portal of PacketFence allows a guest/user to register using his Google, Facebook, LinkedIn, Windows Live, OpenID Connect or Github account.

For each providers, we maintain an allowed domain list to punch holes into the firewall so the user can hit the provider login page. This list is available in each OAuth2 authentication source.

You must enable the passthrough option in your PacketFence configuration (fencing.passthrough in pf.conf).

13.2.1. Google

In order to use Google as a OAuth2 provider, you need to get an API key to access their services. Sign up here : <http://code.google.com/apis/console>. In the Google APIs Console, go into 'Credentials → Create Credentials → OAuth client ID → Web Application', then enter a name and make sure you use this URI for the "Authorized redirect URIs" field : https://YOUR_PORTAL_HOSTNAME/oauth2/callback. Of course, replace the hostname with the values from `general.hostname` and `general.domain`. Save to get the Client ID and Client secret.

You can keep the default configuration, modify the App ID & App Secret (Given by Google on the developer platform) and Portal URL (https://YOUR_PORTAL_HOSTNAME/oauth2/callback).

Also, add the following Authorized domains : *.google.com, *.google.ca, *.google.fr, *.gstatic.com,googleapis.com,accounts.youtube.com (Make sure that you have the google domain from your country like Canada ☞ *.google.ca, France ☞ *.google.fr, etc...)

Once you have your client id, and API key, you need to configure the OAuth2 provider. This can be done by adding a Google OAuth2 authentication source from *Configuration* → *Policies and Access Control* → *Authentication Sources*. Remember to add the Authentication Rules with at least two Actions (example: Role and Access duration).

Moreover, don't forget to add Google as a **Source** from your connection profile definition, available from *Configuration* → *Policies and Access Control* → *Connection Profiles*.

13.2.2. Facebook

To use Facebook as an authentication source, you also need an API code and a secret key. To get one, go here: <https://developers.facebook.com/apps>. When you create your App, make sure you specify the following as the Website URL: https://YOUR_PORTAL_HOSTNAME/oauth2/callback Of course, replace the hostname with the values from `general.hostname` and `general.domain`.

To find the secret, go in your newly created app, and click on 'Settings → Basic'.

While in 'Settings → Basic', add YOUR_PORTAL_HOSTNAME in the **App Domains** field. Next, you will need to add the product **Facebook Login**. Click on **Set up**, and choose **Web** platform. Go through the 5 steps, then on the left side of the screen, go in *Settings* under Facebook Login. For **Valid OAuth Redirect URIs**, enter https://YOUR_PORTAL_HOSTNAME/oauth2/callback and then save changes.

Also, add the following Authorized domains : *.facebook.com, *.fbcdn.net, *.akamaihd.net, *.akamaiedge.net, *.edgekey.net, *.akamai.net (May change)

Once you have your information, you need to configure the OAuth2 provider. This can be done by adding a Facebook OAuth2 authentication source from *Configuration* → *Policies and Access Control* → *Authentication Sources*. Remember to add the Authentication Rules with at least two Actions (example: Role and Access duration).

You can keep the default configuration, modify the App ID & App Secret (Given by Facebook on the developer platform) and Portal URL (https://YOUR_PORTAL_HOSTNAME/oauth2/callback).

Moreover, don't forget to add Facebook as a **Source** from your connection profile definition, available from *Configuration* → *Policies and Access Control* → *Connection Profiles*.

CAUTION

By allowing OAuth through Facebook, you will give Facebook access to the users while they are sitting in the registration VLAN.

13.2.3. Github

To use Github, you also need an API code and a secret key. To get one, you need to create an App here: <https://github.com/settings/applications/new>. When you create your App, make sure you specify the following as the Callback URL https://YOUR_PORTAL_HOSTNAME/oauth2/callback

Of course, replace the hostname with the values from `general.hostname` and `general.domain`.

Once you have your information, you need to configure the OAuth2 provider. This can be done

by adding a GitHub OAuth2 authentication source from *Configuration* → *Policies and Access Control* → *Authentication Sources*. Remember to add the Authentication Rules with at least two Actions (example: Role and Access duration).

Moreover, don't forget to add GitHub as a **Source** from your connection profile definition, available from *Configuration* → *Policies and Access Control* → *Connection Profiles*.

13.2.4. Kickbox

To use Kickbox, you need a API key. To get one, first create an account on <https://kickbox.io>, then navigate to <https://app.kickbox.com/settings/keys>. Click on 'API Keys → Create Key'. Pick a name and choose 'Production' mode and 'Single' verification.

Once you have your API key, you need to configure the OAuth2 provider. This can be done by adding a Kickbox authentication source from *Configuration* → *Policies and Access Control* → *Authentication Sources*. Remember to add the Authentication Rules with at least two Actions (example: Role and Access duration).

Moreover, don't forget to add Kickbox as a **Source** from your connection profile definition, available from *Configuration* → *Policies and Access Control* → *Connection Profiles*.

13.2.5. LinkedIn

To use LinkedIn, you also need an API code and a secret key. To get one, you need to create an App here: <https://developer.linkedin.com/>. When you create your App, make sure you specify the following as the Callback URL https://YOUR_PORTAL_HOSTNAME/oauth2/callback

You can get more details about how to configure your LinkedIn application inside [Microsoft documentation](#).

Of course, replace the hostname with the values from `general.hostname` and `general.domain`.

Once you have your information, you need to configure the OAuth2 provider. This can be done by adding a LinkedIn OAuth2 authentication source from *Configuration* → *Policies and Access Control* → *Authentication Sources*. Remember to add the Authentication Rules with at least two Actions (example: Role and Access duration).

Moreover, don't forget to add LinkedIn as a **Source** from your connection profile definition, available from *Configuration* → *Policies and Access Control* → *Connection Profiles*.

NOTE | When testing LinkedIn OAuth2, use a different LinkedIn account to setup the application and to test the Source in the captive portal.

13.2.6. OpenID Connect

Using OpenID Connect is a bit different than other OAuth2 sources. The reason behind that is because you will setup your own OpenID Connect source or depend on a provider for it. Configuration like token path, authorize path or API URL are specific to your setup. For more information on how to create your own or get a host please visit: <http://openid.net/connect/>.

When you create your App, make sure you specify the following as the Callback URL, https://YOUR_PORTAL_HOSTNAME/oauth2/callback.

Of course, replace the hostname with the values from `general.hostname` and `general.domain`.

OpenID connect have different ways to be configured, make sure to create a client ID and a client secret to work with PacketFence.

Once you have your information, you need to configure the OAuth2 provider. This can be done by adding an OpenID OAuth2 authentication source from *Configuration → Policies and Access Control → Authentication Sources*. Remember to add the Authentication Rules with at least two Actions (example: Role and Access duration).

Moreover, don't forget to add OpenID as a **Source** from your connection profile definition, available from *Configuration → Policies and Access Control → Connection Profiles*.

13.2.7. Twilio

To use Twilio, first create an account on <https://www.twilio.com>. From the console (dashboard) <https://www.twilio.com/console> create a **3rd Party Integration**. Note the **Account SID** and **Auth Token** for later use. From the Phone Manager <https://www.twilio.com/console/phone-numbers/incoming> click the "+" button to **Buy a number** with SMS capability - no payment is needed to start using this phone number right away.

Once you have your information, you need to configure the OAuth2 provider. This can be done by adding a Twilio OAuth2 authentication source from *Configuration → Policies and Access Control → Authentication Sources*. Enter your 'Account SID', 'Auth Token' and 'Phone Number (From)' from above. Remember to add the Authentication Rules with at least two Actions (example: Role and Access duration).

Moreover, don't forget to add Twilio as a **Source** from your connection profile definition, available from *Configuration → Policies and Access Control → Connection Profiles*.

13.2.8. Windows Live

To use Windows live, you also need an API code and a secret key. To get one, you need to create an App here: https://portal.azure.com/#blade/Microsoft_AAD_RegisteredApps/ApplicationsListBlade When you create your App, make sure you specify the following as the Callback URL https://YOUR_PORTAL_HOSTNAME/oauth2/callback replacing the hostname with the values from **general.hostname** and **general.domain**.

Once you have your information, you need to configure the OAuth2 provider. This can be done by adding a WindowsLive OAuth2 authentication source from *Configuration → Policies and Access Control → Authentication Sources*. Remember to add the Authentication Rules with at least two Actions (example: Role and Access duration).

The **App ID** in PacketFence will be **Application (client) ID** in the Azure portal.

The **App secret** must be a client secret created in the **Certificates & secrets** section of your app on Azure AD. Note that Azure AD secrets do expire so make sure you set a reminder to update your secret before it expires.

Moreover, don't forget to add WindowsLive as a **Source** from your connection profile definition, available from *Configuration → Policies and Access Control → Connection Profiles*.

13.3. Eduroam

Eduroam (education roaming) is the secure, world-wide roaming access service developed for the international research and education community.

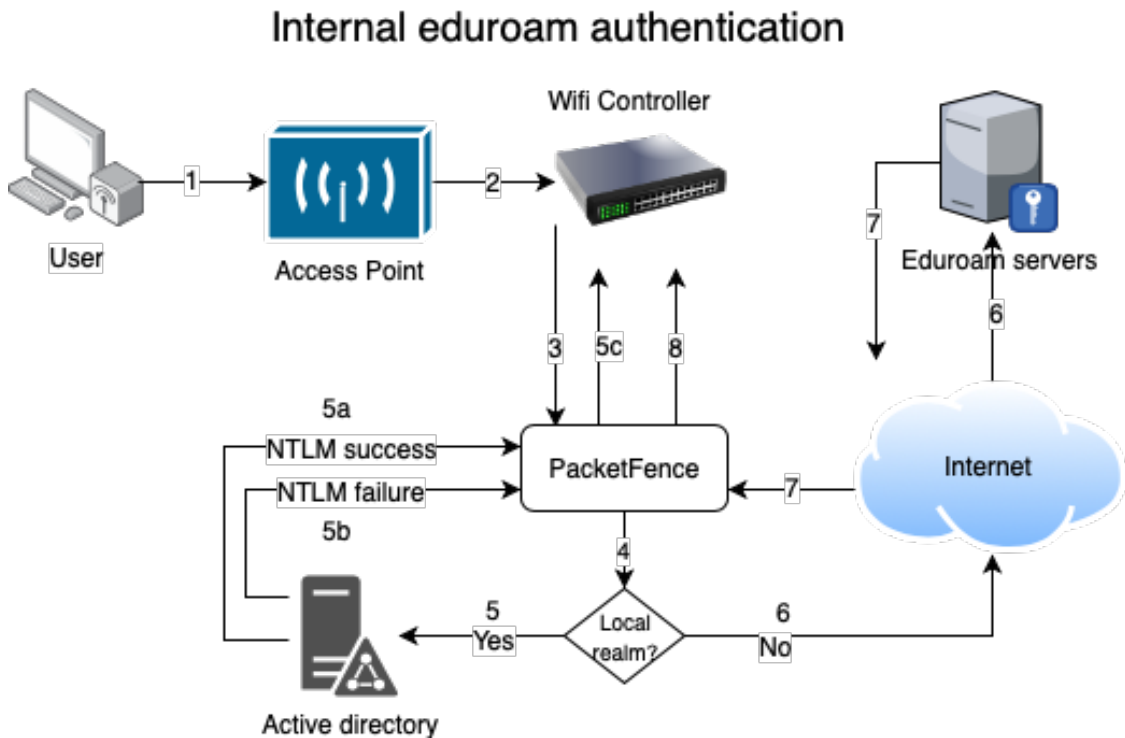
Eduroam allows students, researchers and staff from participating institutions to obtain Internet connectivity across campus and when visiting other participating institutions by simply opening their laptop.

– Eduroam, <https://www.eduroam.org/>

PacketFence supports Eduroam and allows participating institutions to authenticate both locally visiting users from other institutions as well as allowing other institutions to authenticate local users.

Understanding of the Eduroam authentication workflow.

13.3.1. Local authentication



1. The device connects on the Eduroam SSID.
2. The access point forwards the authentication request to the wireless controller.
3. The controller sends the RADIUS authentication to PacketFence on port 11812.
4. PacketFence checks if it's a local REALM.
5. If it's local REALM, PacketFence does a NTLM request to the Active Directory (AD) domain controller to verify the identity.
 - a. The AD validated the credentials.
 - b. The AD did not validate the credentials. PacketFence sends a RADIUS Reject.

- c. After a successful NTLM authentication, PacketFence returns a Radius Access Accept to the wireless controller to apply the production VLAN for that MAC address.
- 6. If it's a not local REALM, PacketFence proxies the radius request to the Eduroam servers.
- 7. The Eduroam servers validate the identity.
- 8. PacketFence returns a Radius Access Accept to the wireless controller to apply the production VLAN for that MAC address.

13.3.2. Configure the Eduroam source

Open the PacketFence administration web interface and go to *Configuration → Policies and Access Control → Authentication Sources*.

Local **Exclusive Sources** and click on **New exclusive source** then **Eduroam**.

The information to configure that source could be found on the Eduroam platform.

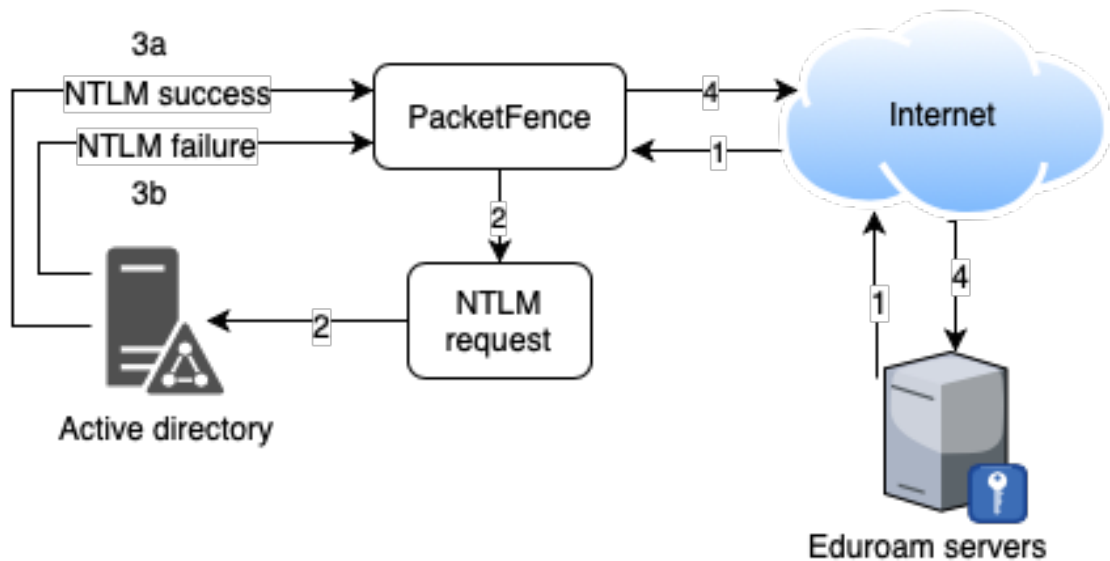
13.3.3. Create the connection profile for local authentication

Go to *Configuration → Policies and Access Control → Connection Profiles → New Connection Profile*.

Create a connection profile named **Local and external Eduroam authentication** Check **Automatically register devices** then create a SSID filter **Eduroam**. Make sure to add the Active Directory source to match on the local users.

13.3.4. Inbound authentication (TLRS to PF)

Inbound eduroam authentication



- 1. Eduroam sends the RADIUS authentication to a public IP address (NAT/PAT) bound to PacketFence on the management IP address (Management VIP for a cluster) on port 1812.

2. PacketFence forwards the NTLM request to the Active Directory.
3. NTLM response
 - a. Successful user identify authentication on the AD
 - b. NTLM request fails because of a bad identity
4. PacketFence replies to the Eduroam servers either a RADIUS Access Accept for a successful authentication or a RADIUS access reject for an unsuccessful authentication. PacketFence sets the REALM to Eduroam for all successful authentications.

First, you need to refer to the previous step [Configure the Eduroam source](#).

Once the source is configured, you will need to create a new connection profile.

13.3.5. Create the connection profile for outbound authentication

Go to *Configuration* → *Policies and Access Control* → *Connection Profiles* → *New Connection Profile*.

Create the Connection Profile named [External Eduroam authentication](#) Check [Automatically register devices](#) then create a REALM filter [Eduroam](#). Next, make sure to add the Eduroam source previously created.

13.4. SAML Authentication

PacketFence supports SAML authentication in the captive portal in combination with another internal source to define the level of authorization of the user.

First, transfer the Identity Provider metadata on the PacketFence server. In this example, it will be under the path [/usr/local/pf/conf/idp-metadata.xml](#).

Then, transfer the certificate and CA certificate of the Identity provider on the server. In this example, they will be under the paths [/usr/local/pf/conf/ssl/idp.crt](#) and [/usr/local/pf/conf/ssl/idp-ca.crt](#). If it is a self-signed certificate, then you will be able to use it as the CA in the PacketFence configuration. Make sure [-----BEGIN CERTIFICATE-----](#) and [-----END CERTIFICATE-----](#) headers are present in these certificate files.

Then, to configure SAML in PacketFence, go in *Configuration* → *Policies and Access Control* → *Sources* and then create a new Internal source of the type SAML and configure it.

The screenshot shows the 'New Authentication Source' configuration window in PacketFence. The form is titled 'New Authentication Source' with a 'SAML' tag. It contains the following fields:

- Name: mysaml
- Description: Acme Inc.
- Service Provider entity ID: PF_ENTITY_ID
- Path to Service Provider key (x509): /usr/local/pf/conf/ssl/server.key
- Path to Service Provider cert (x509): /usr/local/pf/conf/ssl/server.crt
- Identity Provider entity ID: IDP_ENTITY_ID
- Path to Identity Provider metadata: /usr/local/pf/conf/idp-metadata.xml
- Path to Identity Provider cert (x509): /usr/local/pf/conf/ssl/idp.crt
- Path to Identity Provider CA cert (x509): /usr/local/pf/conf/ssl/idp-ca.crt. A note below this field states: 'If your Identity Provider uses a self-signed certificate, put the path to its certificate here instead.'
- Attribute of the username in the SAML response: urn:oid:0.9.2342.19200300.100.1.1
- Authorization source: inverse. A note below this field states: 'The source to use for authorization (rule matching).'

At the bottom of the form, there are two buttons: 'Create' (highlighted in blue) and 'Reset'.

Where :

- **Service Provider entity ID** is the identifier of the Service Provider (PacketFence). Make sure this matches your Identity Provider configuration.
- **Path to Service Provider key** is the path to the key that will be used by PacketFence to sign its messages to the Identity Provider. A default one is provided under the path : `/usr/local/pf/conf/ssl/server.key`
- **Path to Service Provider cert** is the path to the certificate associated to the key above. A self-signed one is provided under the path : `/usr/local/pf/conf/ssl/server.key`
- **Path to Identity Provider metadata** is the path to the metadata file you transferred above (should be in `/usr/local/pf/conf/idp-metadata.xml`)
- **Path to Identity Provider cert** is the path to the certificate of the identity provider you transferred on the server above (should be in `/usr/local/pf/conf/ssl/idp.crt`).
- **Path to Identity Provider CA cert** is the path to the CA certificate of the identity provider you transferred on the server above (should be in `/usr/local/pf/conf/ssl/ca-idp.crt`). If the certificate above is self-signed, put the same path as above in this field.
- **Attribute of the username in the SAML response** is the attribute that contains the username in the SAML assertion returned by your Identity Provider. The default should fit at least

SimpleSAMLphp.

- **Authorization source** is the source that will be used to match the username against the rules defined in it. This allows to set the role and access duration of the user. The 'Authentication' section of this document contains explanations on how to configure an LDAP source which can then be used here.

Once this is done, save the source and you will be able to download the Service Provider metadata for PacketFence using the link 'Download Service Provider metadata' on the page.

Configure your identity provider according to the generated metadata to complete the Trust between PacketFence and your Identity Provider.

In the case of SimpleSAMLPHP, the following configuration was used in `metadata/saml20-sp-remote.php` :

```
$metadata['PF_ENTITY_ID'] = array(
    'AssertionConsumerService' => 'http://PORTAL_HOSTNAME/saml/assertion',
    'SingleLogoutService' => 'http://PORTAL_HOSTNAME/saml/logoff',
);
```

NOTE

PacketFence does not support logoff on the SAML Identity Provider. You can still define the URL in the metadata but it will not be used.

13.4.1. Passthroughs

In order for your users to be able to access the Identity Provider login page, you will need to activate passthroughs and add the Identity Provider domain to the allowed passthroughs.

To do so, go in *Configuration* → *Network Configuration* → *Networks* → *Fencing*, then check **Passthroughs** and add the Identity Provider domain name to the **Passthroughs** list.

Next, restart `iptables` and `pfdns` services to apply your new passthroughs.

13.5. Billing Engine

PacketFence integrates the ability to use a payment gateway to bill users to gain access to the network. When configured, the user who wants to access the network / Internet is prompted by a page asking for it's personal information as well as it's credit card information.

PacketFence currently supports two payment gateways: Authorize.net, Paypal and Stripe.

In order to activate the billing, you will need to configure the following components :

- Billing source(s)
- Billing tier(s)

13.5.1. Configuring a billing source

First select a billing provider and follow the instructions below.

Paypal

NOTE

This provider requires that your PacketFence server is accessible on the public domain. For this your PacketFence portal should be available on a public IP using the DNS server name configured in PacketFence.

If you have a business account and do not want to configure a test environment, you can skip the next section.

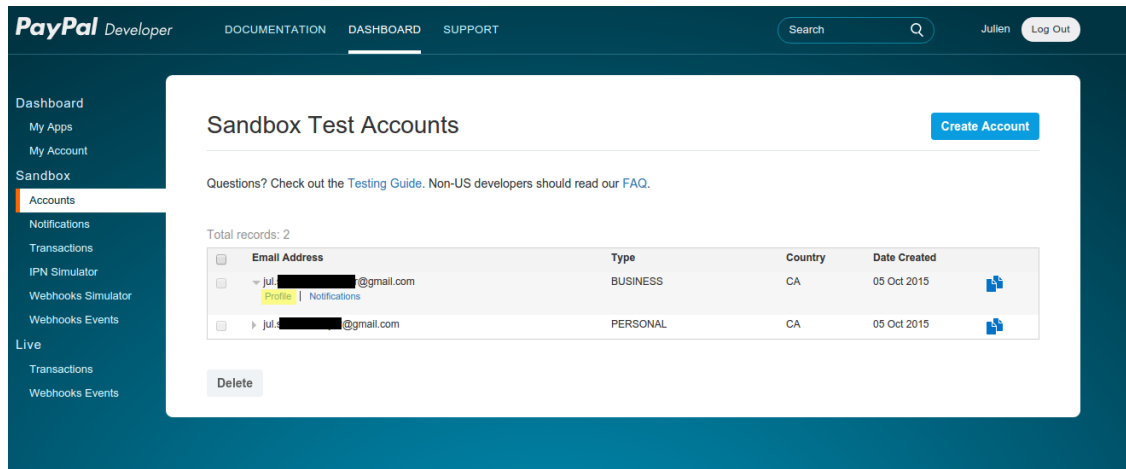
Sandbox account

To configure a sandbox paypal account for use in PacketFence, head to <https://developer.paypal.com/> and either sign up or login into your existing account.

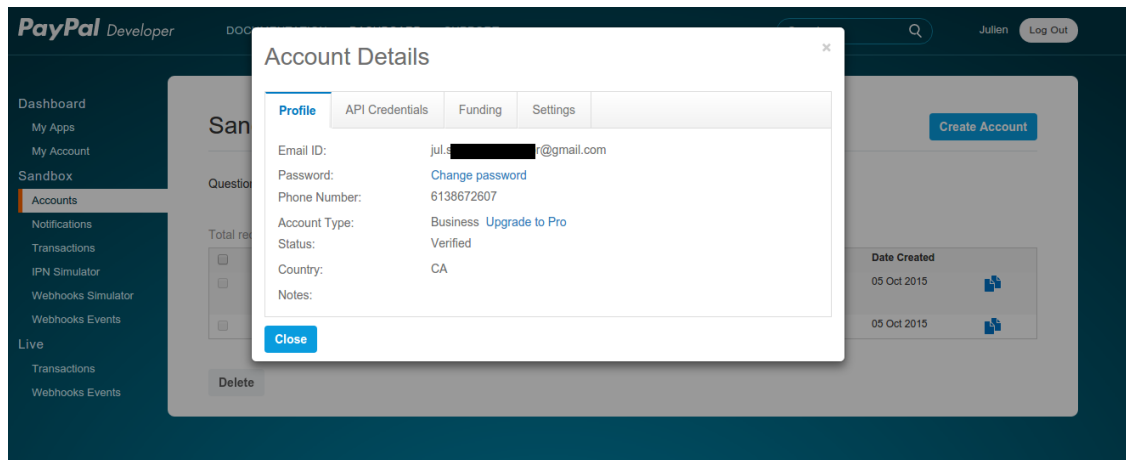
Then in the Sandbox menu, click **Accounts**

Create an account that has the type **Personal** and one that has the type **Business**.

Afterwards, go back into accounts, and expand the business account, then click **Profile**



Now click the 'Change password' link and change the password and note it.



Do the same thing with the personal account you created

Configuring the merchant account

Login into the Paypal business account that you created at <https://www.sandbox.paypal.com/> if you are using a sandbox account or on <https://www.paypal.com/> if you are using a real account.

Next go in *My Account* → *Profile* in order to go into your profile configuration.

Next in the **Selling Preferences** you will need to select **Website Payment Preferences**

Configure the settings so they match the screenshot below.

You should turn on **Auto Return**, set the return URL to https://YOUR_PORTAL_HOSTNAME/billing/paypal/verify.

You should also take note of the **Identity Token** as it will be required in the PacketFence configuration.

The screenshot shows the PayPal 'Website Payment Preferences' configuration page. At the top, there is a navigation bar with tabs for 'My Account', 'Send Money', 'Create an Invoice', 'Merchant Services', 'Products & Services', and 'Community'. Below this is a sub-navigation bar with 'Overview', 'Add Funds', 'Withdraw', 'History', 'Resolution Centre', and 'Profile'. The main heading is 'Website Payment Preferences' with a 'Back to My Profile' link. The section is titled 'Auto Return for Website Payments'. A description states: 'Auto Return for Website Payments brings your buyers back to your website immediately after payment completion. Auto Return applies to PayPal Website Payments, including Buy Now, Donations, Subscriptions, and Shopping Cart. [Learn More](#)'. There are two radio buttons for 'Auto Return': 'On' (selected) and 'Off'. Below this is a 'Return URL' field with the text 'Return URL: '. A 'Return URL Requirements' section lists three bullet points: 1. Per the user agreement, you must provide verbiage on the page displayed by the Return URL that will help the buyer understand that the payment has been made and that the transaction has been completed. 2. You must provide verbiage on the page displayed by the Return URL that explains that payment transaction details will be emailed to the buyer. 3. Example: Thank you for your payment. Your transaction has been completed, and a receipt for your purchase has been emailed to you. You may log into your account at www.sandbox.paypal.com/ca to view details of this transaction. Below this is a section for 'Payment Data Transfer (optional)'. A description states: 'Payment Data Transfer allows you to receive notification of successful payments as they are made. The use of Payment Data Transfer depends on your [system configuration](#) and your Return URL. Please note that in order to use Payment Data Transfer, you **must** turn on Auto Return.' There are two radio buttons for 'Payment Data Transfer': 'On' (selected) and 'Off'. At the bottom, the 'Identity Token' is displayed as 'A-WI2 [REDACTED] TAAD9He'.

Next go back in your profile configuration *My account* → *Profile* and select **Encrypted Payment Settings**

Now on this page you will need to submit the certificate used by PacketFence to Paypal

(`/usr/local/pf/conf/ssl/server.crt` by default).

Once you have submitted it, note its associated **Cert ID** as you will need to configure it in PacketFence.

Still on that page, click the **Download** link to download the Paypal public certificate and put it on the PacketFence server under path : `/usr/local/pf/conf/ssl/paypal.pem`

PayPal

My Account | Send Money | Create an Invoice | Merchant Services | Products & Services | Community

Overview | Add Funds | Withdraw | History | Resolution Centre | **Profile**

Website Payment Certificates [Back to My Profile](#)

Dynamically encrypt your Website Payments by downloading PayPal's public certificate and provide PayPal your public certificate. You will need to dynamically encrypt Website Payments with your own code to use this feature. [Learn more](#)

For added protection, you may also block payments that are made using non-encrypted buttons by setting this option on the [Website Payment Preferences](#) page.

You can create simple encrypted Website Payments without downloading keys by using the PayPal [Button Factory](#)

PayPal Public Certificate

PayPal requires that you use the PayPal Public Certificate with your code to encrypt buttons so that only PayPal can decipher the encrypted contents. Click the **Download** button below to download the PayPal Public Certificate.

Download

Your Public Certificates

PayPal will use your public certificate to decipher the encrypted content of your website buttons. You may add up to 6 different certificates.

	Cert ID	Certifying Authority	Expiration Date
<input checked="" type="radio"/>	R[REDACTED]VNG	/C=CA/ST=QC/L=Montreal/O=Inverse/CN=127.0.0.1/emailAddress=support@inverse.ca	May 14, 2016 21:59:11 GMT-04:00

Download | **Remove** | **Add**

[About Us](#) | [Contact Us](#) | [Legal Agreements](#) | [Privacy](#) | [Fees](#) | [Site Feedback](#) [-]

Copyright © 1999-2015 PayPal. All rights reserved.

CAUTION | The certificate will **NOT** be the same if you use a sandbox account or a real account.

Configuring PacketFence

Now, in the PacketFence administration interface, go in *Configuration* → *Policies and Access Control* → *Sources* and create a new source of type 'Billing → Paypal'.

The screenshot shows the 'New Authentication Source' configuration window for Paypal. The left sidebar contains navigation options like Policies and Access Control, Roles, Domains, Authentication Sources, Network Devices, Connection Profiles, Compliance, Integration, Advanced Access Configuration, Network Configuration, and System Configuration. The main form fields are as follows:

- Name: Paypal-test
- Description: Paypal
- Currency: CAD
- Send billing confirmation:
- Test mode:
- Identity token: A-BCDE-FG-HIJKLMN
- Cert ID: RE2IIIIIIING
- Cert file: /usr/local/pf/conf/ssl/server.crt
- Key file: /usr/local/pf/conf/ssl/server.key
- Paypal cert file: /usr/local/pf/conf/ssl/paypal.pem
- Email address: Christopher.test@test.com
- Payment type: Buy Now
- Authorized domains: *.paypal.com,*.paypalobjects.com
- Create Local Account:
- Database passwords hashing method: NTLM
- Password length: 8
- Amount of logins for the local account: 0

Buttons: Create, Reset

Where :

- **Identity token** is the one you noted when on the 'Website Payment Preferences' page.
- **Cert ID** is the one you noted when on the 'Encrypted Payment Settings'.
- **Payment type** is whether the access is donation based (not mandatory to pay for it).

- **Email address** is the email address of the merchant paypal account.
- **Cert file** is the path to the PacketFence certificate (`/usr/local/pf/conf/ssl/server.crt` by default).
- **Key file** is the path to the PacketFence certificate (`/usr/local/pf/conf/ssl/server.key` by default).
- **Paypal cert file** is the path to the Paypal certificate (`/usr/local/pf/conf/ssl/paypal.pem` in this example).
- **Currency** is the currency that will be used in the transactions.
- **Test mode** should be activated if you are using a sandbox account.

NOTE If they aren't already enabled, you will need to enable passthroughs so that users can reach the domains of this provider. Refer to the **Passthroughs** section of this document for details

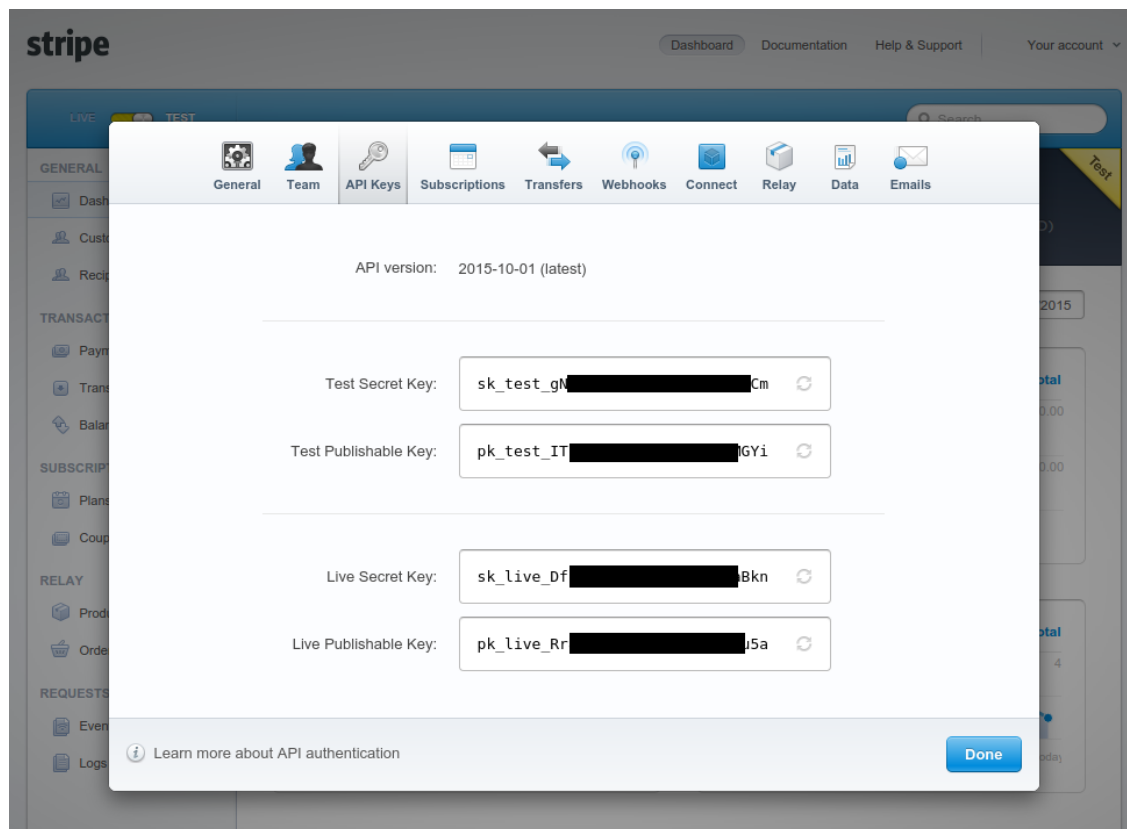
Stripe

Stripe account

First go on <https://dashboard.stripe.com>, create an account and login.

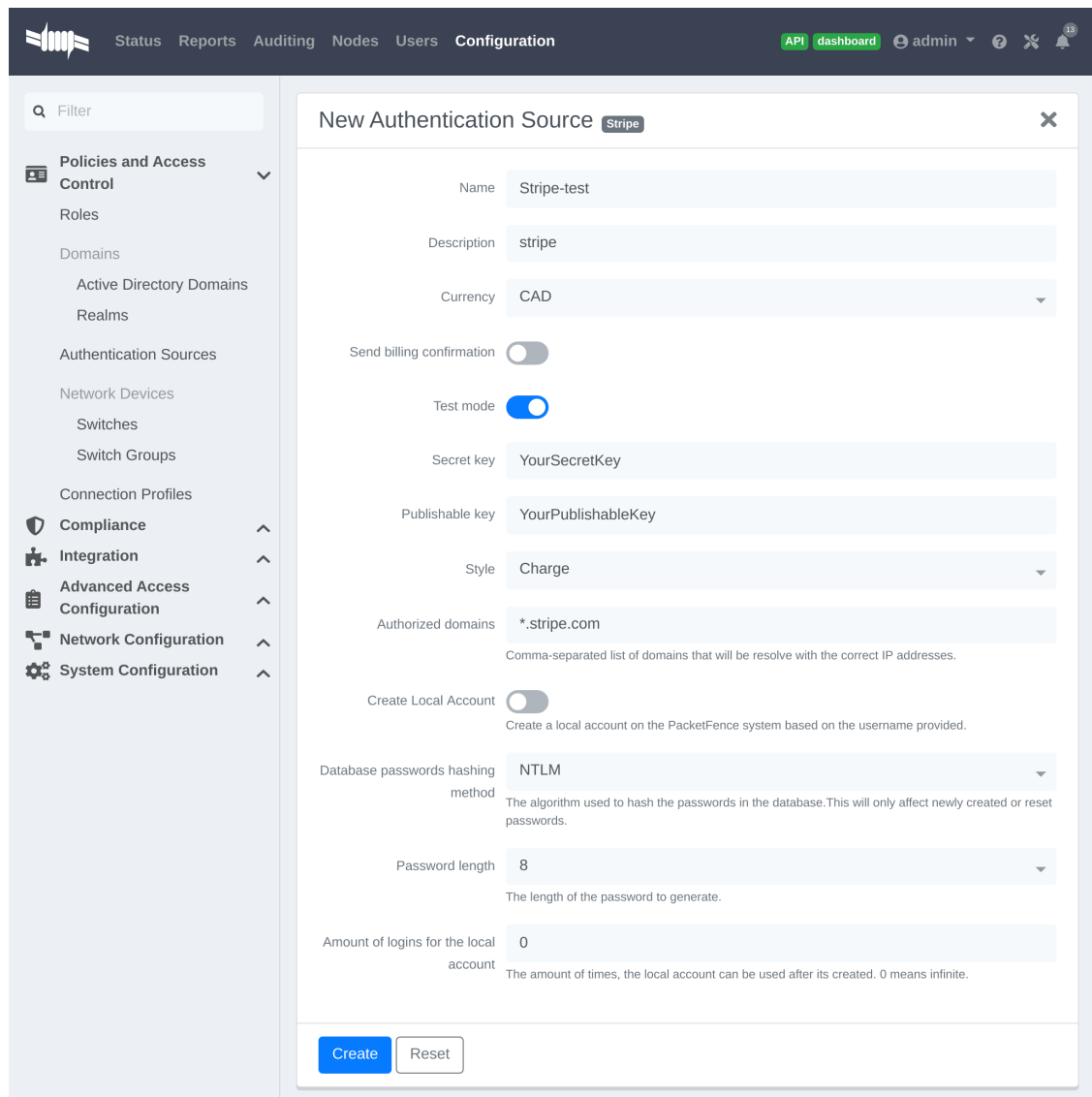
Next on the top right click **Your account** then **Account settings**.

Navigate to the **API keys** tab and note your key and secret. The test key should be used when testing the configuration and the live key when putting the source in production.



Configuring PacketFence

Now, in the PacketFence administration interface, go in *Configuration* → *Policies and Access Control* → *Sources* and create a new source of type *Billing* → *Stripe*



The screenshot shows the PacketFence administration interface. The top navigation bar includes 'Status', 'Reports', 'Auditing', 'Nodes', 'Users', and 'Configuration'. The 'Configuration' section is active, and the 'Policies and Access Control' menu is expanded. The 'New Authentication Source' form is displayed, titled 'New Authentication Source Stripe'. The form fields are as follows:

- Name: Stripe-test
- Description: stripe
- Currency: CAD
- Send billing confirmation:
- Test mode:
- Secret key: YourSecretKey
- Publishable key: YourPublishableKey
- Style: Charge
- Authorized domains: *.stripe.com
- Create Local Account:
- Database passwords hashing method: NTLM
- Password length: 8
- Amount of logins for the local account: 0

Buttons for 'Create' and 'Reset' are located at the bottom of the form.

Where :

- **Secret key** is the secret key you got from your Stripe account.
- **Publishable key** is the publishable key you got from your Stripe account.
- **Style** is whether you are doing a one-time charge or subscription based billing (recurring). See section [Subscription based registration](#) below for details on how to configure it.
- **Currency** is the currency that will be used in the transactions.
- **Test mode** should be activated if you are using the test key and secret account.

NOTE

If they aren't already enabled, you will need to enable passthroughs so that users can reach the domains of this provider. Refer to the [Passthroughs](#) section of this document for details.

Stripe customer portal

PacketFence supports integrating with the Stripe customer portal and will handle subscription cancellations by default using webhooks. Additional hooks can be supported by extending [lib/pf/billing/custom_hook.pm](#).

In order to enable the customer portal in Stripe, go in *Settings* → *Product settings* → *Billing* → *Customer portal*. Next, enable the options you want for the customer portal.

PacketFence supports the following options:

- Allow customers to view their invoice history
- Allow customers to update their billing information
- Allow customers to update their payment method
- Allow customers to cancel subscriptions

Optionally, once this is configured, you need to make sure your captive portal is accessible publicly for Stripe to send it webhooks if you want to support subscription cancellations. Once its accessible publicly, configure a webhook to receive the event `customer.subscription.deleted` on https://PF_DOMAIN_NAME/hook/billing/STRIPE_SOURCE_ID. Replace `STRIPE_SOURCE_ID` by the identifier (name) of your Stripe source in your PacketFence configuration.

Next, in PacketFence, go in your Stripe source (*Configuration* → *Policies and Access Control* → *Authentication Sources*) and enable the option `Customer portal` in your Stripe source.

Now when your users will visit the status page (https://PF_DOMAIN_NAME/status), they will have the option to manage their subscriptions and visit the Stripe customer portal.

13.5.2. Adding billing tiers

Once you have configured one or more billing source, you need to define billing tiers which will define the price and target authentication rules for the user.

In the PacketFence administration interface, go in *Configuration* → *Advanced Access Configuration* → *Billing tiers*

Then click `Add billing tier` and configure it.

The screenshot shows the 'New Billing Tier' configuration form in the PacketFence web interface. The form is titled 'New Billing Tier' and has a close button (X) in the top right corner. The form contains the following fields and options:

- Billing Tier:** Simple
- Name:** Simple access
- Description:** This tier will grant you basic access to the network for a duration of 24 hours.
- Price:** 1.99. Below the field is the text: 'The price that will be charged to the customer.'
- Role:** guest. Below the field is the text: 'The target role of the devices that use this tier.'
- Access Duration:** 24 hours. Below the field is the text: 'The access duration of the devices that use this tier.'
- Use Time Balance:** A toggle switch is currently turned off. Below the field is the text: 'Check this box to have the access duration be a real time usage. This requires a working accounting configuration.'

At the bottom of the form, there are two buttons: 'Create' (in blue) and 'Reset' (in white).

Where :

- **Billing tier** is the unique identifier of the billing tier.
- **Name** is the friendly name of the billing tier.
- **Description** is an extended description of the billing tier.
- **Price** is the amount that will be charged to the user.
- **Access duration** is the amount of time the user will be granted access to your network.
- **Role** is the target role the user should be in.
- **Use time balance** defines if the access duration should be computed on real-time access duration meaning if the user buys 24 hours of access he can use the network for 24 hours in different time blocks. This requires a valid RADIUS accounting configuration.

NOTE

If don't want to use all the billing tiers that are defined, you can specify the ones that should be active in the **Connection profile**.

13.5.3. Subscription based registration

PacketFence supports subscription based billing using Stripe as a billing provider.

Billing tier

When using subscription based billing, it is advised to configure the billing tier so it has an almost infinite access duration (e.g. 20 years) as the billing provider will be contacting the PacketFence server when the subscription is canceled.

You should configure a billing tier for each subscription plan you want to have. This example will

use the plan `simple` and `advanced` configured using the following parameters.

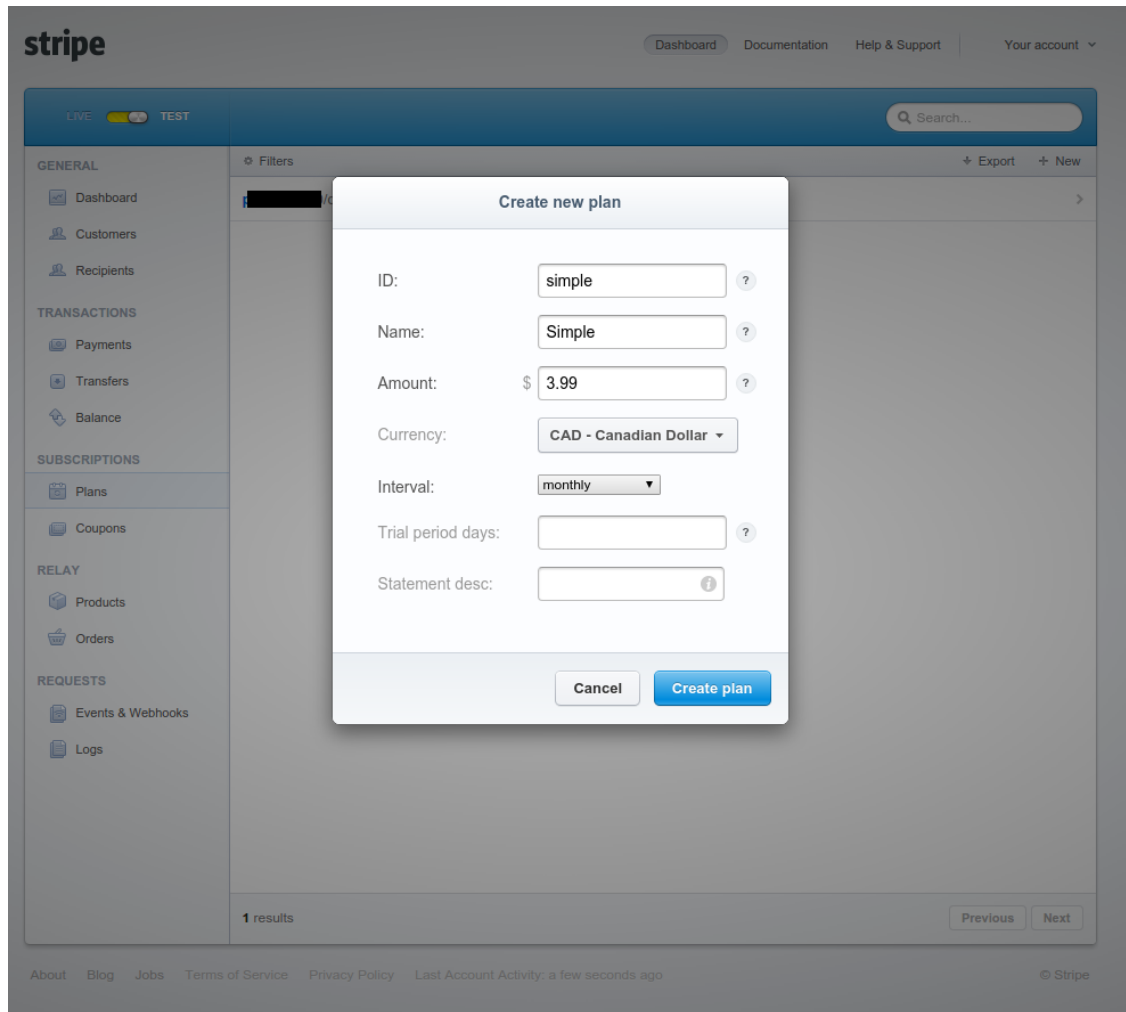
```
[simple]
name=Simple network access
description=Click here if you are poor
price=3.99
role=guest
access_duration=10Y
use_time_balance=disabled
```

```
[advanced]
name=Simple network access
description=Click here if you are poor
price=9.99
role=advanced_guest
access_duration=10Y
use_time_balance=disabled
```

Stripe configuration

Then in your Stripe dashboard, you should go in *Subscriptions* → *Plans*.

Then create a new plan.



Where :

- **ID** is the billing tier identifier. It is **important** that this matches the ID of the billing tier in PacketFence.
- **Amount** is the price of the plan. It is **important** that this matches the price of the billing tier in PacketFence.
- **Currency** is the currency that will be used in the transactions. It is **important** that this matches the currency of the Stripe source in PacketFence.
- **Interval** is the interval at which the customer should be billed. In the case of this example, it is monthly.

Now, following the same procedure, create the advanced plan.

Receiving updates from Stripe

As the subscription can be cancelled by a user, you need to setup your PacketFence installation to receive updates from Stripe.

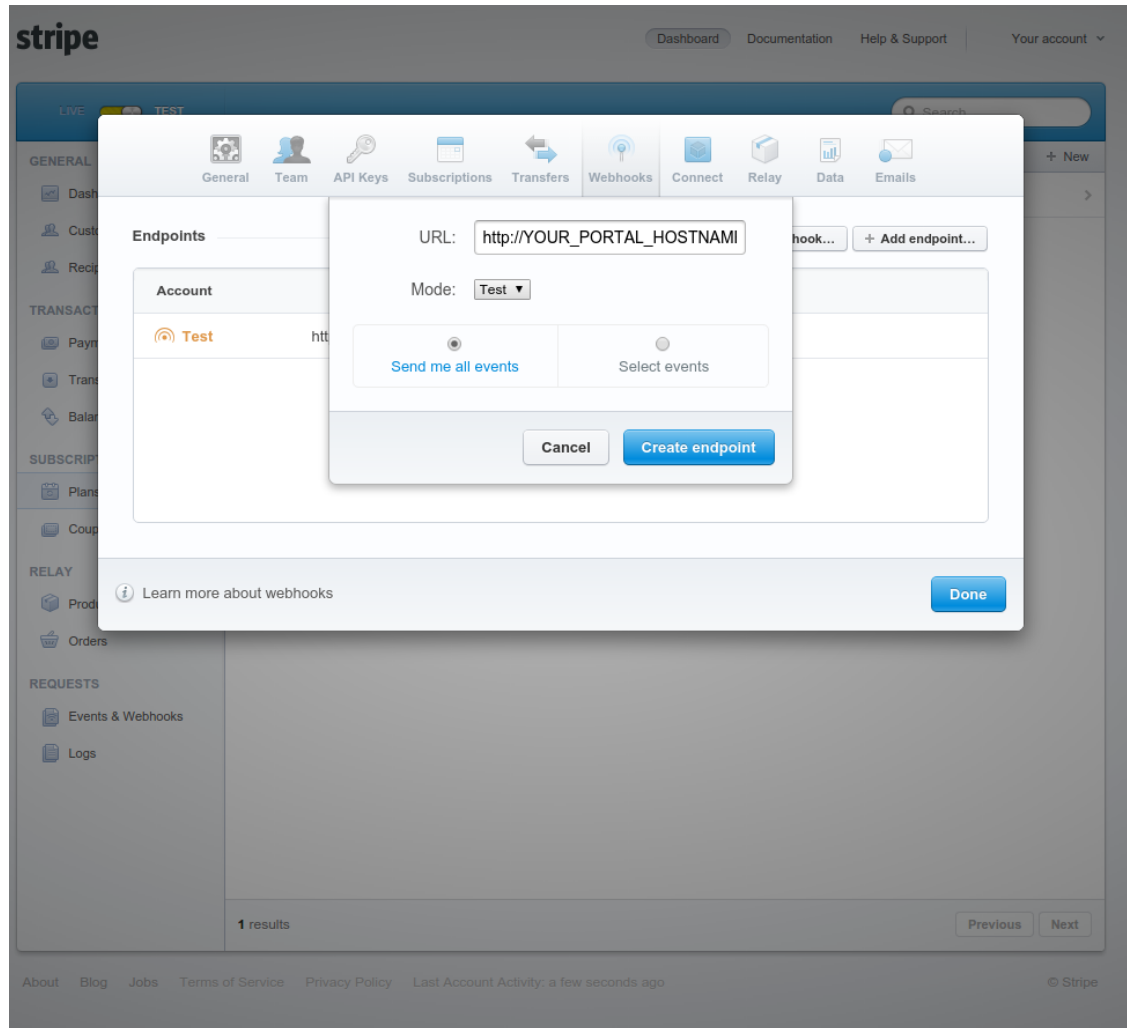
Updates are sent using HTTP requests on a public IP.

You need to make sure that your PacketFence server is available through a public IP on port 80

and that your PacketFence server hostname resolves on the public domain.

Then, in Stripe, configure a **Webhook** so Stripe informs PacketFence of any event that happens in this Stripe merchant account.

In order to do so go in *Your Account* → *Account Settings* → *Webhooks* and click **Add endpoint**.



Where :

- **URL** is the URL to the PacketFence server. This should be http://YOUR_PORTAL_HOSTNAME/hook/billing/stripe
- **Mode** is whether this webhook is for testing mode or live mode

Now every time a user unsubscribes from a plan, PacketFence will be notified and will unregister that device from your network.

13.5.4. Extending access before it ends

PacketFence allows users to extend their access before it has ended. In order to do so, you need to enable **Allow access to registration portal when registered** accessible via the **Captive Portal** tab of the **Connection Profiles**. Once this is activated, the users can reach

https://YOUR_PORTAL_IP/status and select **Extend your access** in order to be able to access the billing section after they have registered.

13.6. External API Authentication

PacketFence also supports calling an external HTTP API as an authentication source. The external API needs to implement an authentication action and an authorization action.

13.6.1. Authentication

This should provide the information about whether or not the username/password combination is valid

These information are available through the POST fields of the request

The server should reply with two attributes in a JSON response

- **result** : should be 1 for success, 0 for failure
- **message** : should be the reason it succeeded or failed

Example JSON response :

```
{"result":1,"message":"Valid username and password"}
```

13.6.2. Authorization

This should provide the actions to apply on a user based on it's attributes

The following attributes are available for the reply : **access_duration**, **access_level**, **sponsor**, **unregdate**, **category**.

Sample JSON response, note that not all attributes are necessary, only send back what you need.

```
{"access_duration":"1D","access_level":"ALL","sponsor":1,"unregdate":"2030-01-01","category":"default"}
```

NOTE | See /usr/local/pf/addons/example_external_auth for an example implementation compatible with PacketFence.

13.6.3. PacketFence Configuration

In PacketFence, you need to configure an HTTP source in order to use an external API.

Here is a brief description of the fields :

- **Host** : First, the protocol, then the IP address or hostname of the API and lastly the port to connect to the API.
- **API username and password** : If your API implements HTTP basic authentication (RFC 2617) you can add them in these fields. Leaving any of those two fields empty will make

PacketFence do the requests without any authentication.

- **Authentication URL** : URL relative to the host to call when doing the authentication of a user. Note that it is automatically prefixed by a slash.
- **Authorization URL** : URL relative to the host to call when doing the authorization of a user. Note that it is automatically prefixed by a slash.

13.7. Azure AD integration

PacketFence supports integrating with the Azure Active Directory for authenticating users on the captive portal, the admin interface and for 802.1X users using EAP-TTLS PAP. If your only goal is to authenticate users on the captive portal, using the OpenID implementation of Azure AD may be better suited. This section is aimed at providing username/password authentication through Azure AD.

13.7.1. Creating the PacketFence app

1. Open the 'Azure Active Directory' in your Azure portal
2. Go in 'Manage→App registrations→New registration'
3. Settings for the app
 - a. Name: PacketFence
 - b. Supported account types: Accounts in this organizational directory only - (Single tenant)
 - c. Redirect URI must be left blank
 - d. Save the app
4. Note down the 'Application (client) ID' and 'Directory (tenant) ID' for later usage
 - a. In your application, go in 'Certificates & secrets' and select 'New client secret'
 - i. Description: PacketFence
 - ii. Make sure you note down its expiry date so you can renew it before its expiration. Failure to do so will prevent authentication from working on PacketFence
 - iii. Save the secret
 - b. Note down the 'Value' of your client secret for later usage
 - c. Still in your application, go to 'API permissions'
 - i. Click on 'Add a permission'
 - A. Go to the 'Microsoft APIs' tab
 - B. Select 'Microsoft Graph'
 - C. Select 'Application permissions'
 - D. Add the permission **Directory.Read.All**
 - E. Click on 'Grant admin consent'
 - ii. Make sure **User.Read** is already there as a delegated permission

13.7.2. Disabling MFA

Currently, PacketFence requires that multi-factor authentication be disabled for the PacketFence app. If you use Azure AD premium, you can create a rule to exclude this only for the PacketFence

application. If you don't use Azure AD premium, this must be disabled for all your users.

Disabling MFA using Azure AD premium

1. Open the "Azure Active Directory" in your Azure portal
2. Go in 'Manage→Properties'
 - a. Click 'Manage Security defaults'
 - b. Disable the toggle 'Enable Security defaults' and save
3. Next, go in 'Manage→Security→Conditional Access'
 - a. Click 'New policy' and enter the following settings:
 - i. Name: 2FA policy
 - ii. Under 'Users and groups', select 'All users'
 - iii. Under 'Cloud apps or actions', go in the 'Exclude' section and select the 'PacketFence' app you created earlier in the 'Select excluded cloud apps'
 - iv. Under 'Grant', select 'Grant access' and check 'Require multi-factor authentication' and any other settings your organization requires.
 - v. At the bottom, make sure 'Enable policy' is set to 'On' and save your policy

Disabling MFA without Azure AD premium

WARNING

This will disable common recommended settings from Microsoft. Using Azure AD premium is the correct way to perform this. This option is only suggested for testing or when its impossible to have access to Azure AD premium.

1. Open the "Azure Active Directory" in your Azure portal
2. Under 'Manage', open 'Properties'
 - a. Click 'Manage Security defaults'
 - b. Disable the toggle 'Enable Security defaults' and save

13.7.3. Configuring PacketFence

1. Under 'Configuration→Policies and Access Control→Authentication Sources', create a new 'Azure Active Directory' internal source
 - a. Client ID: the 'Client ID' that was displayed while configuring the 'PacketFence' app inside Azure
 - b. Client Secret: the secret you created inside the 'PacketFence' app in Azure AD
 - c. Tenant ID: the 'Tenant ID' that was displayed while configuring the 'PacketFence' app inside Azure
 - d. Add any authentication or administration rules and then save the source

With this configuration, you can now use this source in your connection profiles to authenticate and authorize users on the captive portal and use it with EAP-TLS to authorize users (getting the role and access duration) as long as your EAP-TLS certificates use the distinguished name of the Azure AD users as their common name. Additionally, you can use this source for authenticating users in the admin interface and for VPN access.

Using Azure AD in 802.1X

You can perform 802.1X authentication of users using Azure AD but this will only work with supplicants configured to perform EAP-TTLS PAP which provides the RADIUS server with the plain-text password of the user. Support for this type of authentication is not as broad as EAP-PEAP MSCHAPv2 in the 802.1X supplicants but unfortunately Azure AD doesn't support MSCHAP authentication. Refer to the documentation of your operating system on how to configure EAP-TTLS PAP. This section will only focus on enabling EAP-TTLS PAP for your Azure AD users in PacketFence.

1. Under 'Configuration→Policies and Access Control→Realms', create a new realm
 - a. Realm: enter the realm of your Azure AD users. Example, if the usernames have the following format `bob@inverseinc.onmicrosoft.com`, then your realm is `inverseinc.onmicrosoft.com`
 - b. Go in the 'Stripping' tab of the realm and select your Azure AD source under 'Azure AD Source for TTLS PAP'
 - c. Still in the 'Stripping' tab, disable (uncheck), the following settings:
 - i. Strip on the portal
 - ii. Strip on the admin
 - iii. Strip in RADIUS authorization
 - d. Save the realm
2. Restart radiusd using `/usr/local/pf/bin/pfcmd service radiusd restart`
3. All the users matching this realm will now authenticate against Azure AD. Make sure you also have a connection profile with auto-registration enabled and the Azure AD source in it so that your users are correctly authorized when connecting.

13.8. Google Workspace LDAP Integration

1. Go to <https://admin.google.com/> and sign in as a Google Workspace domain administrator.
2. Go to Apps > LDAP > Add Client.
3. Provide an LDAP client name and an optional Description. Any descriptive values are acceptable. For example, the name could be 'PacketFence' and the description could be 'PacketFence LDAP Client'. Click the Continue button.
4. Set Access Permission according to your needs. You must choose either 'Entire domain (PacketFence)' or 'Selected organizational units' for both 'Verify user credentials' and 'Read user information'. Select 'Add LDAP Client'
5. Download the generated certificate. This is required for PacketFence to communicate with the Google Secure LDAP service. Save the downloaded certificates for later use. After downloading, click the Continue to Client Details button.
6. Expand the Service Status section and turn the LDAP client 'ON for everyone'. After selecting 'Save', click on the 'Service Status' bar again to collapse and return to the rest of the settings.
7. Expand the Authentication section and choose 'Generate New Credentials'. Copy/note these credentials for later use. After selecting 'Close', click on the 'Authentication' bar again to collapse and return to the rest of the settings.

13.8.1. Configuring PacketFence

1. Under 'Configuration→Policies and Access Control→Authentication Sources', create a new 'Google Workspace LDAP' internal source
 - a. The following are the configuration values obtained during the LDAP client configuration earlier:
 - i. Bind DN: The access credentials username
 - ii. Password: The access credentials password
 - iii. Client Certificate: The .crt file text from the downloaded certificate bundle
 - iv. Client Key: The .key file text from the downloaded certificate bundle

14. Advanced Portal Configuration

14.1. Portal Modules

The PacketFence captive portal flow is highly customizable. This section will cover the **Portal Modules** which are used to define the behavior of the captive portal.

NOTE

When upgrading from a version that doesn't have the portal modules, the PacketFence Portal Modules configuration already comes with defaults that will fit most cases and offers the same behavior as previous versions of PacketFence. Meaning, all the available Connection Profile sources are used for authentication, then the available provisioners will be used.

First, a brief description of the available Portal Modules:

- **Root:** This is where it all starts, this module is a simple container that defines all the modules that need to be applied in a chained way to the user. Once the user has completed all modules contained in the Root, he is released on the network.
- **Choice:** This allows to give a choice between multiple modules to the user. The 'default_registration_policy' is a good example of a choice that is offered to the user.
- **Chained:** This allows you to define a list of modules that a user needs to go through in the order that they are defined - ex: you want your users to register via Google+ and pay for their access using PayPal.
- **Message:** This allows you to display a message to the user. An example is available below in *Displaying a message to the user after the registration*
- **URL:** This allows you to redirect the user to a local or external URL which can then come back to the portal to continue. An example is available below in *Calling an external website*.
- **Authentication:** The authentication modules can be of a lot of types. You would want to define one of these modules, in order to override the required fields, the source to use, the template or any other module attribute.
 - **Billing:** Allows to define a module based on one or more billing sources
 - **Choice:** Allows to define a module based on multiple sources and modules with advanced filtering options. See the section *Authentication Choice module* below for a detailed explanation.
 - **Login:** Allows you to define a username/password based module with multiple internal sources (Active Directory, LDAP, ...)
 - **SelectRole:** Allows to define a module to override the role given when registering a device. For instance: an admin user is trying to register a device using the normal registration process, with this module the admin can choose which role to apply to the device while registering it. It will bypass authentication rules.
 - **Other modules:** The other modules are all based on the source type they are assigned to, they allow to select the source, the AUP acceptance, and mandatory fields if applicable.

14.1.1. Examples

This section will contain the following examples:

- Prompting for fields without authentication.
- Prompting additional fields during the authentication.
- Chained authentication.
- Mixing login and Secure SSID on-boarding on the portal.
- Displaying a message to the user after the registration.

Creating a custom root module

First, create a custom root module for our examples in order to not affect the default policy. In order to do so, go in *Configuration* → *Advanced Access Configuration* → *Portal Modules*, then click **Add Portal Module** and select the type **Root**. Give it the identifier **my_first_root_module** and the description **My first root module**, then hit save.

Next, head to *Configuration* → *Policies and Access Control* → *Connection Profiles*, select the connection profile you use (most probably **default**) and then under **Root Portal Module**, assign **My first root module** then save your profile. If you were to access the captive portal now, an error would display since the Root module we configured doesn't contain anything.

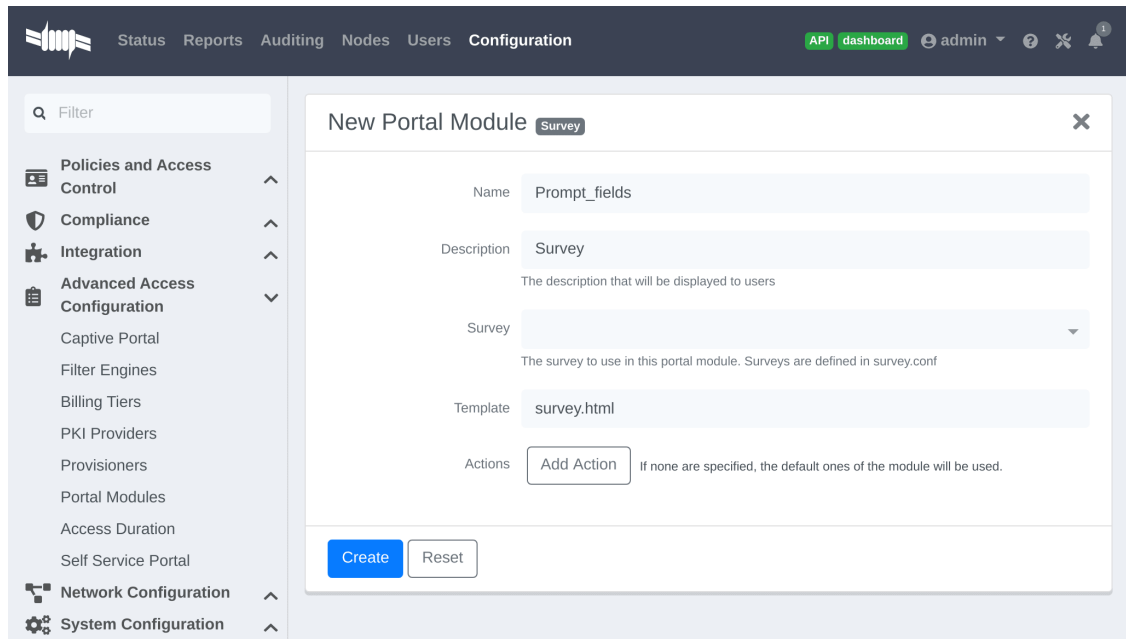
You could add some of the pre-configured modules to the new Root module you created and that would make the error disappear.

Prompting for fields without authentication

In order to prompt fields without authentication, you can use the Null source with the Null Portal Module.

PacketFence already comes with a Null source pre-configured. If you haven't modified it or deleted it, you can use it for this example. Otherwise, go in *Configuration* → *Policies and Access Control* → *Sources* and create a new Null source with a catchall rule that assigns a role and access duration.

Then go in *Configuration* → *Advanced Access Configuration* → *Portal Modules* and click **Add Portal Module** and select **Authentication Authentication::Null**. Set the **Identifier** to **prompt_fields** and configure the module with the **Mandatory fields** you want and uncheck **Require AUP** so that the user doesn't have to accept the AUP before submitting these fields.



Next, add the `prompt_fields` module in `my_first_root_module` (removing any previous modules) and save it. Now when visiting the portal, it should prompt you for the fields you define in the module. Then, submitting these information will assign you the role and access duration that you defined in the `null` source.

Prompting additional fields during the authentication

If you want to prompt additional fields during the authentication process for a module, you can define a Module based on that source that will specify the additional mandatory fields for this source.

You can also add additional mandatory fields to the default policies that are already configured.

This example will make the `default_guest_policy` require the user to enter a first name, last name and address so that guests have to enter these three information before registering.

Go in *Configuration* → *Advanced Access Configuration* → *Portal Modules* and click the `default_guest_policy`. Add `firstname`, `lastname` and `address` to the **Mandatory fields** and **Save**.

Next, add the `default_guest_policy` to `my_first_root_module` (removing any previous modules). Now when visiting the portal, any of the guest sources configured in your connection profile will require you to enter both the mandatory fields of the source (ex: phone + mobile provider) and the mandatory fields you defined in the `default_guest_policy`.

NOTE Not all sources support additional mandatory fields (ex: OAuth sources like Google, Facebook, ...).

Chained authentication

The portal modules allow you to chain two or more modules together in order to make the user accomplish all of the actions in the module in the desired sequence.

This example will allow you to configure a **Chained** module that will require the user to login via any configured OAuth source (Github, Google+, ...) and then validate his phone number using SMS registration.

For the OAuth login we will use the **default_oauth_policy**, so just make sure you have an OAuth source configured correctly and available in your Connection Profile.

Then, we will create a module that will contain the definition of our SMS registration.

Go in *Configuration* → *Advanced Access Configuration* → *Portal Modules* then click **Add Portal Module** and select **Authentication Authentication::SMS**.

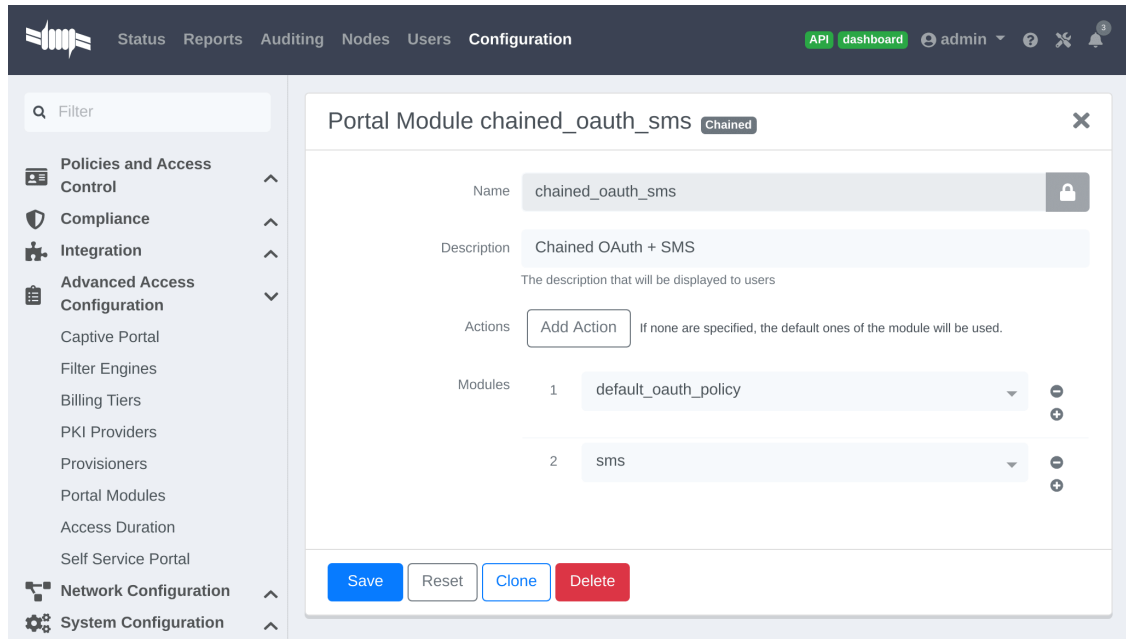
Configure the portal module so that it uses the **sms** source and uncheck the **Require AUP** option since the user will have already accepted the AUP when registering using OAuth.

The screenshot shows the 'New Portal Module' configuration form. The form is titled 'New Portal Module' with a sub-header 'Authentication::SMS'. It contains several fields:

- Name: sms
- Description: SMS registration (The description that will be displayed to users)
- PID field: telephone (Which field should be used as the PID.)
- Authentication Source: sms (The source to use in the module. If no source is specified, all the sources of the connection profile will be used.)
- Mandatory fields: (The additional fields that should be required for registration)
- Fields to save: (These fields will be saved through the registration process)
- Require AUP: (Require the user to accept the AUP)
- AUP template: aup_text.html (The template to use for the Acceptable Use Policy)
- Signup template: signin.html (The template to use for the signup)
- Actions: Add Action (If none are specified, the default ones of the module will be used.)

At the bottom of the form, there are 'Create' and 'Reset' buttons.

Then, add another Portal Module of type 'Chained'. Name it **chained_oauth_sms**, assign a relevant description and then add **default_oauth_policy** and **sms** to the **Modules** fields



Next, add the `chained_oauth_sms` module in `my_first_root_module` (removing any previous modules) and save it. Now when visiting the portal, you should have to authentication using an OAuth source and then using SMS based registration.

Note that if you add want to keep some fields the user previously filled you can add 'Saved fields' in the portal module. Per example the first module ask for the telephone number and the second too, then you can add 'telephone' as a 'Saved fields' and the second module will not ask for it.

Mixing login and Secure SSID on-boarding on the portal

This example will guide you through configuring a portal flow that will allow for devices to access an open SSID using an LDAP username/password but also give the choice to configure the Secure SSID directly from the portal.

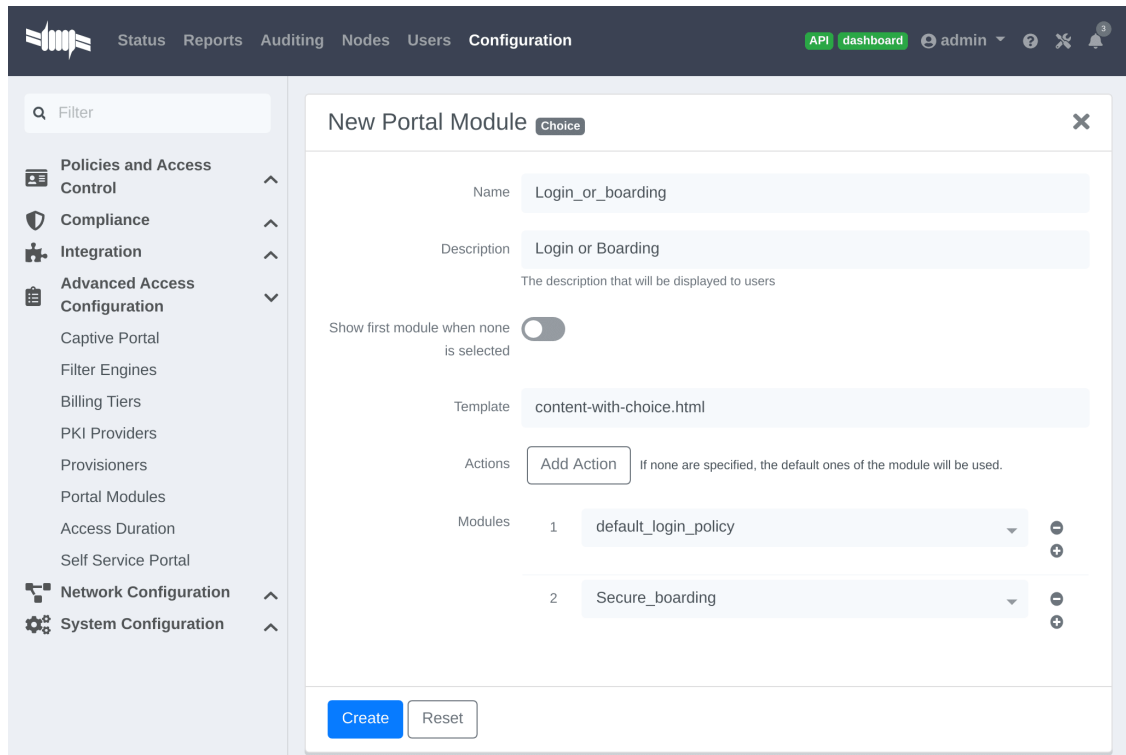
First, we need to configure the provisioners for the Secure SSID onboarding. Refer to section *Apple and Android Wireless Provisioning* of this guide to configure your provisioners and add them to the connection profile.

Create a provisioner of the type `Deny` and add it with your other provisioners (putting any other provisioner before it). This will make sure that if there is no match on the other provisioners, it will not allow the device through.

Also in the connection profile add your LDAP source to the available sources so its the only one available.

Next, create a 'Provisioning' portal module by going in *Configuration* → *Advanced Access Configuration* → *Portal Modules*. Set the 'Identifier' to `secure_boarding` and the description to `Board Secure SSID`. Also uncheck 'Skippable' so the user is forced to board the SSID should it choose this option.

Then, still in the Portal Modules, create a 'Choice' module. Set the 'Identifier' to `login_or_boarding` and description to 'Login or Boarding'. Add `secure_boarding` and `default_login_policy` to the 'Modules' field and save.



Next, add the `login_or_boarding` module in `my_first_root_module` (removing any previous modules) and save it. Now when visiting the portal, you will have the choice between login to the LDAP source and gain access to the network or directly use provisioning in order to configure your device for a Secure SSID.

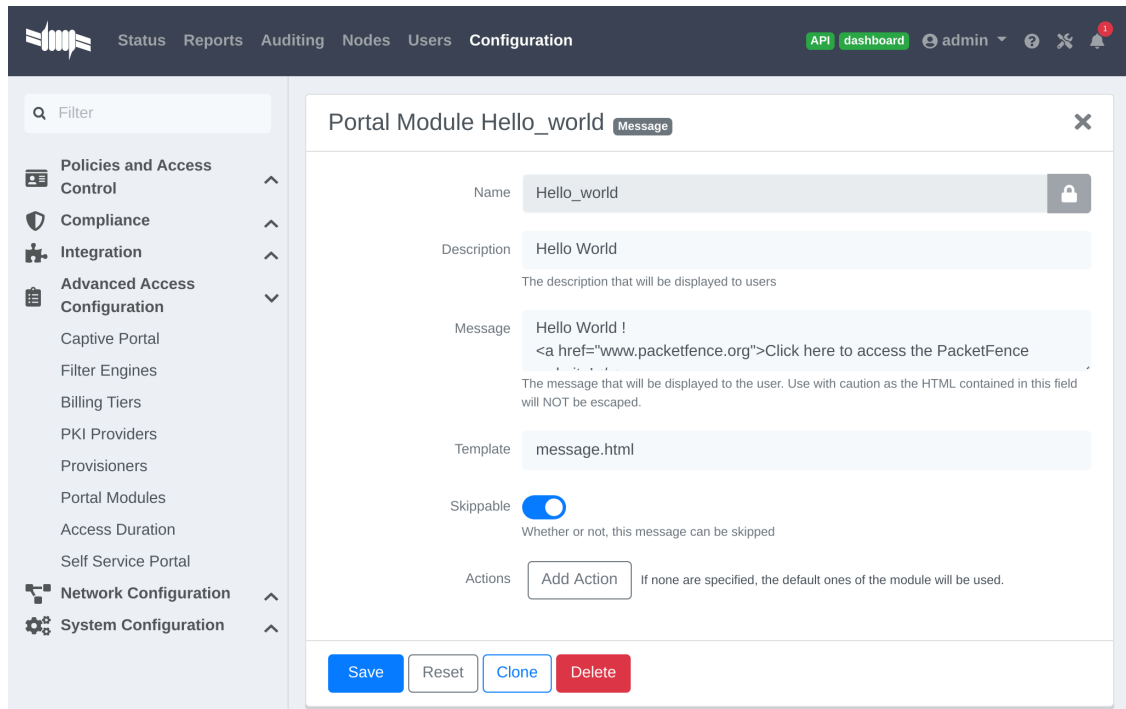
Displaying a message to the user after the registration

Using the 'Message' module you can display a custom message to the user. You can also customize the template to display in order to display a fully custom page.

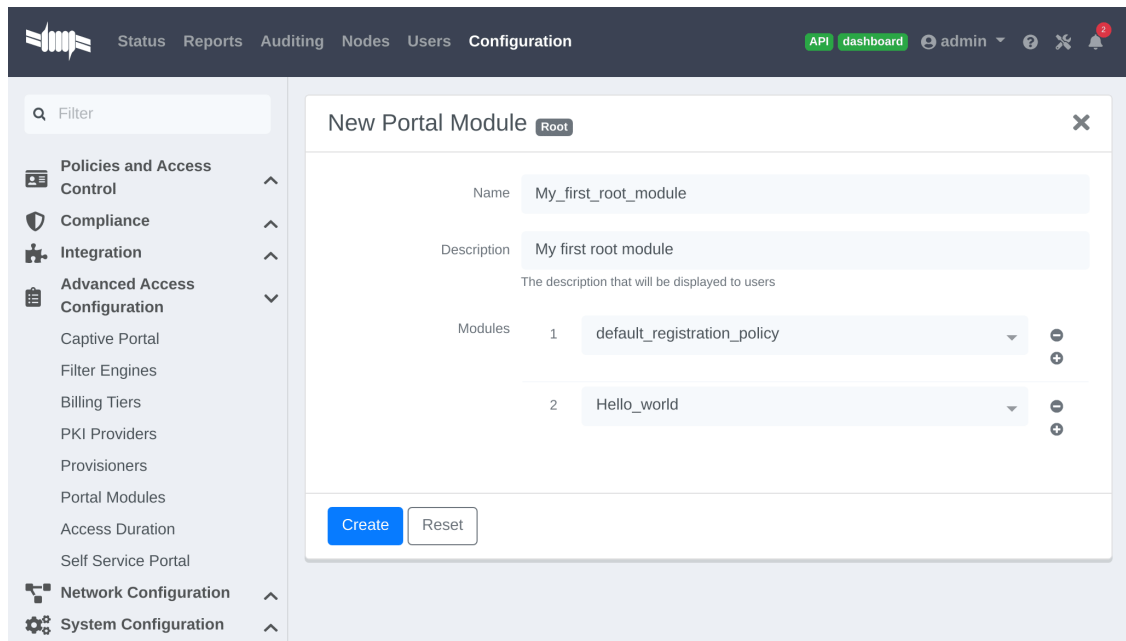
Go in *Configuration* → *Advanced Access Configuration* → *Portal Modules*, then click 'Add Portal Module' and select 'Message'. Set the 'Identifier' to `hello_world` and the description to `Hello World`.

Then put the following in the 'Message' field

```
Hello World !
<a href="www.packetfence.org">Click here to access the PacketFence website!</a>
```



Next, add `default_registration_policy` and `hello_world` in the 'Modules' of `my_first_root_module` (removing any previous modules) and save it. Now when visiting the portal, you should have to authenticate using the sources defined in your connection profile and you will then see the hello world message.



Calling an external website

Using the 'URL' module, you can redirect the user to a local or external URL (as long as it is in the passthroughs). Then you can make it so the portal accepts a callback in order for the flow to continue. Having a callback on local or external URL is a requirement to use this portal module.

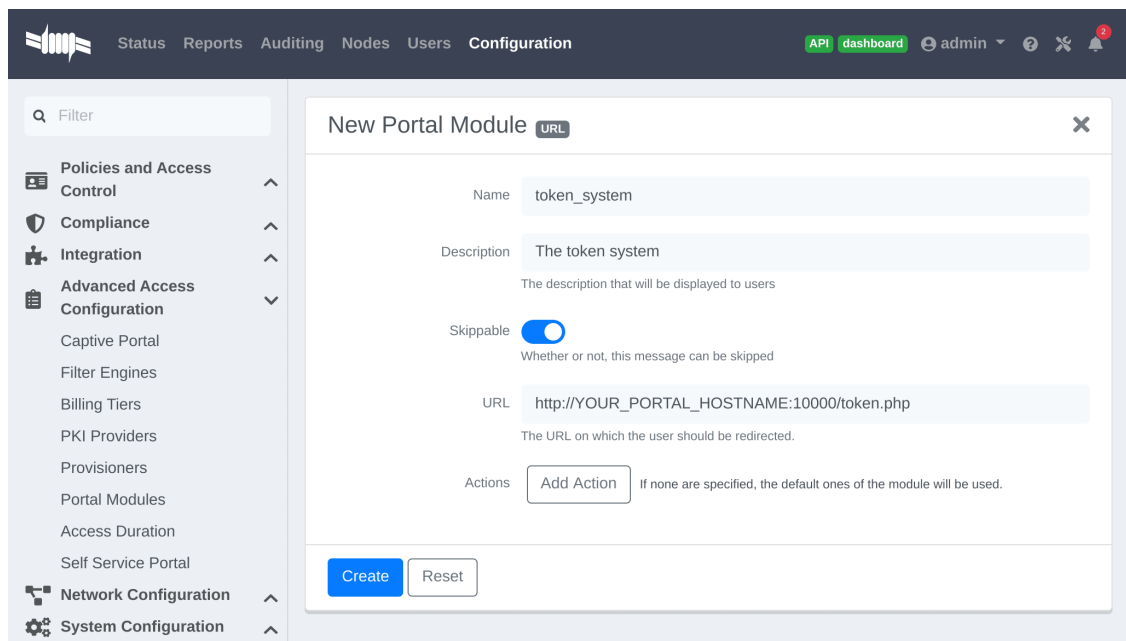
Otherwise, users will be **always** redirected to URL without any possibility to continue the registration process.

In this example, the portal will redirect to an externally hosted PHP script that will give a random token to the user and then callback the portal to complete the registration process.

The example script is located in `addons/example_external_auth/token.php` and a README is available in that directory to set it up.

Once you have the script installed and working on URL: http://YOUR_PORTAL_HOSTNAME:10000/token.php, you can configure what you need on the PacketFence side.

Go in *Configuration* → *Advanced Access Configuration* → *Portal Modules*, then click **Add Portal Module** and select **URL**. Set the 'Identifier' to `token_system`, the 'Description' to `Token system` and the 'URL' to http://YOUR_PORTAL_HOSTNAME:10000/token.php.



Next, add `default_registration_policy` and `token_system` in the 'Modules' of `my_first_root_module` (removing any previous modules) and save it. Now when visiting the portal, you should have to authenticate using the sources defined in your connection profile and then you will be redirected to example token system. Clicking the continue link on that system will bring you back to the portal and complete the registration process.

14.1.2. Authentication Choice module (advanced)

The Authentication Choice module allows to define a choice between multiple sources using advanced filtering rules, manual selection of the sources and selection of Portal Modules.

All the sources that are defined in the 'Sources' field will be available for usage by the user. Same goes for the modules defined in 'Modules'.

You can also define which mandatory fields you want to prompt for these authentication choices. Although you can still configure them on any 'Authentication Choice' module, they will only be shown if they are applicable to the source.

In addition to the manual selection above you can dynamically select sources part of the Connection Profile based on their object attribute (Object Class, Authentication type, Authentication Class).

NOTE | You can find all the authentication objects in [lib/pf/Authentication/Source](#)

- Sources by class: Allows you to specify the perl class name of the sources you want available
 - ex: `pf::Authentication::Source::SMSSource` will select all the SMS sources.
`pf::Authentication::Source::BillingSource` will select all the billing sources (Paypal, Stripe, ...)
- Sources by type: Allows you to filter out sources using the `type` attribute of the Authentication object
- Sources by Auth Class: Allows you to filter our sources using the `class` attribute of the Authentication object.

You can see the 'default_guest_policy' and 'default_oauth_policy' for examples of this module.

14.1.3. SelectRole

The SelectRole module allows to define specific roles manually when registering a device. This is useful if for instance you ask your technical crew to register new devices.

The configuration is simple, you have a role which is the 'Admin role' the one allowed to select the role while registering a new device and the 'Role List' which is the list of roles that can be chosen from while registering a device.

For instance; techs are in the AD group tech support and get the role 'tech support' while registering, let's put 'tech support' as the 'Admin role'. They are allowed to register new devices with the roles 'default', 'voice' and 'guest'. Every time someone with the role 'tech support' will try to register a device on a connection profile where this portal module is active, then the crew memeber will be asked to choose which role to assign to this device.

14.1.4. Onfailure Onsuccess

The `on_failure` and `on_success` actions allow you to create a more complex workflow and will permit to change the root portal module based on the result of the authentication. Let's say you have a root portal module linked to a `Authentication::Login` module associated to a connection profile and you want to present a Guest authentication if the login failed. Then you need to configure another root portal module "Guest" linked with a `Authentication::SMS` module and in the previous `Authentication::Login` add and action `on_failure` `Guest`.

14.2. Portal Surveys

PacketFence has the ability to perform surveys via the captive portal and store the results in dedicated tables in the database.

14.2.1. Setup

In order for the survey tables to be created automatically based on the definition of your surveys, you must grant create and alter rights to the database user defined in `pf.conf`. By default this user is 'pf'. On your database, connect to the MariaDB CLI as root and execute the following:

```
MariaDB> GRANT CREATE,ALTER ON pf.* TO 'pf'@'%';  
MariaDB> GRANT CREATE,ALTER ON pf.* TO 'pf'@'localhost';
```

14.2.2. Configuring your survey

Next, you will have to configure your survey in `/usr/local/pf/conf/survey.conf`. Here is an example of a survey:

```
[survey1]  
description=Mustard Turkey Sandwich Brothers  
  
[survey1 field gender]  
label=What is your gender?  
type=Select  
choices=<<EOT  
M|Male  
F|Female  
EOT  
required=yes  
  
[survey1 field firstname]  
label=What is your firstname?  
type=Text  
required=yes  
  
[survey1 field lastname]  
label=What is your lastname?  
type=Text  
required=yes  
  
[survey1 field sandwich_quality]  
label=On a scale of 1 to 5, how good was your sandwich today?  
type=Scale  
minimum=1  
maximum=5  
required=yes  
  
[survey1 field preferred_sandwich]  
label=What is your preferred sandwich?  
type=Select  
choices= <<EOT  
Classic|Classic  
Extra Turkey|Sandwich with extra turkey  
Extra Mustard|Sandwich with extra mustard  
EOT  
required=yes
```

```
[survey1 field comments]
label=Enter any additionnal comments here
type=TextArea
required=no
```

```
[survey1 data ssid]
query=node.last_ssid
```

```
[survey1 data ip]
query=ip
```

NOTE | Make sure you reload the configuration after setting it up by doing `/usr/local/pf/bin/pfcmd configreload hard`

In the example above, some of the data is being collected via fields on the captive portal directly (ex: `survey1 field firstname`) and some are collected via contextual data (ex: `survey1 data ssid`).

Fields are defined the following way:

- **label**: The label/question that goes with the field which will be displayed on the portal
- **table**: The table in which the survey data should be stored. If its not defined, it will use the ID of the survey. Tables are always prefixed with `survey_` even when this attribute is set.
- **type**: The type of input that should be displayed to the user. The following types are available:
 - **Select**: makes the user select a choice in a pre-defined list
 - **Text**: a simple small text input
 - **TextArea**: a bigger text input
 - **Scale**: a numeric scale. The `minimum` and `maximum` attributes control the range of numbers the user can select
 - **Checkbox**: a simple checkbox
 - **Email**: a text field with email validation (only validates the format)
- **required**: Whether or not the field is mandatory in the survey

Data fields are defined using a query and has access to node and person informations. Here are a few examples:

- `node.last_ssid`: The SSID the device is connected to, if applicable
- `node.device_class`: The Fingerbank device class
- `node.last_switch`: The switch/controller/access point the device is connected to
- You can get all the available node fields by executing the following command: `perl -I/usr/local/pf/lib -I/usr/local/pf/lib_perl/lib/perl5 -Mpf::node -MData::Dumper -e 'print Dumper(node_view("00:11:22:33:44:55"))'`
- `person.source`: If there was authentication done in the portal, this will provide the source that was used
- `person.email`: If there was authentication done in the portal, this will provide the email that

was used

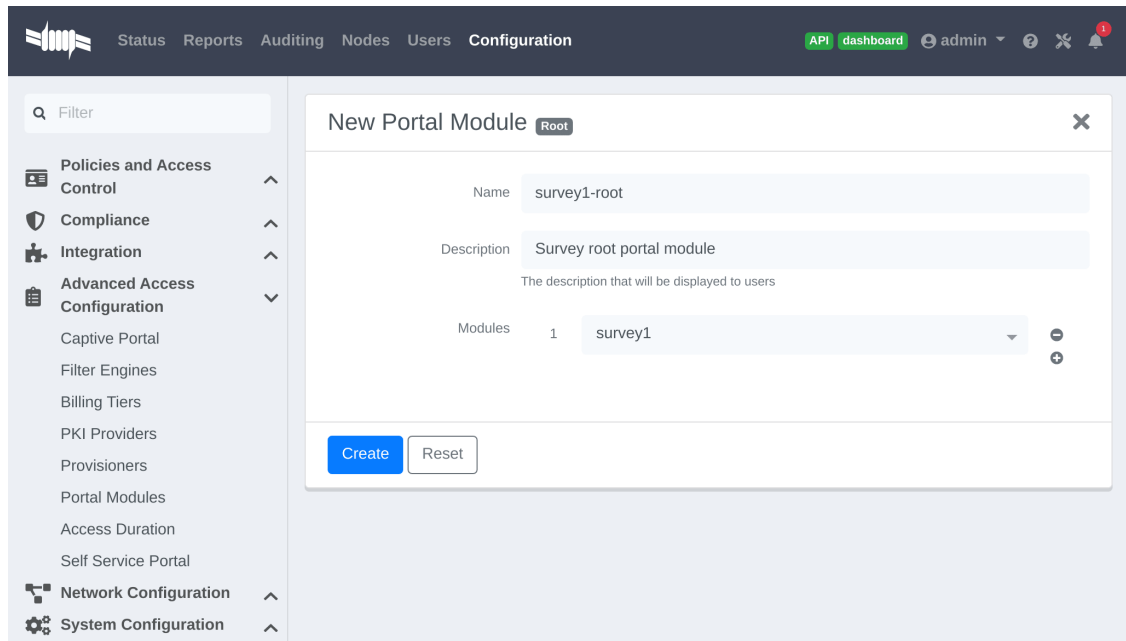
- You can get all the available person fields by executing the following command: `perl -I/usr/local/pf/lib -I/usr/local/pf/lib_perl/lib/perl5 -Mpf::person -MData::Dumper -e 'print Dumper(person_view("admin"))'`
- `ip`: The IP address of the client

14.2.3. Putting the survey on the portal

In order for your survey to be available on the portal, you will have to configure a portal module for it. In order to do so, go in *Configuration* → *Advanced Access Configuration* → *Portal Modules* and create a new Survey portal module with the following settings:

The screenshot shows a web interface for configuring a new portal module. The top navigation bar includes 'Status', 'Reports', 'Auditing', 'Nodes', 'Users', and 'Configuration'. The 'Configuration' section is active, showing 'API dashboard' and user 'admin'. A sidebar on the left lists categories: Policies and Access Control, Compliance, Integration, Advanced Access Configuration, Network Configuration, and System Configuration. The main content area is a 'New Portal Module' form with a 'Survey' tab. The form fields are: Name (survey1), Description (My first survey), Survey (survey1), and Template (survey.html). There is an 'Add Action' button and a 'Create' button at the bottom.

Then either add your survey to another portal module (Choice, Chained or Root) or create a Root portal module dedicated to the survey:



Once that is configured, make sure you have the right Root portal module on the applicable connection profile in *Policies and Access Control* → *Connection Profiles* → *Name of the profile* → *Root Portal Module*.

14.2.4. Exploring the collected data

All the data that is collected in the example survey will be stored in a table named `survey_survey1`. You can create *Reports* on your survey tables via `/usr/local/pf/conf/report.conf`. Here is an example for the survey created above:

```
[survey1]
description=My first survey report
base_table=survey_survey1
columns=firstname as "Firstname", lastname as "Lastname", preferred_sandwich as
"Preferred Sandwich", gender as "Gender"
```

NOTE Make sure you reload the configuration after setting it up by doing `/usr/local/pf/bin/pfcmd configureload hard`

Refer to the [Reports](#) section of this document for advanced configuration.

14.2.5. Cleaning up

When you are happy with the structure of your surveys, it is recommended to remove the `CREATE` and `ALTER` rights to the `pf` database user from a security perspective. In order to do so, execute the following commands. This step is optional and should only be done once the structure of your survey is set in stone.

```
MariaDB> REVOKE CREATE,ALTER ON pf.* FROM 'pf'@'%';  
MariaDB> REVOKE CREATE,ALTER ON pf.* FROM 'pf'@'localhost';
```

14.3. Devices Registration

Users have the possibility to **register** their devices right from a special portal page. When accessing this page, users will be prompted to login as if they were registering themselves. Once logged in, the portal will ask them to enter the device MAC address that will then be matched against the Fingerbank database to match authorized devices list. The device will be registered with the user's id and can be assigned into a specific category for easier management.

In order to configure this, you can configure a self service portal policy in *Configuration* → *Advanced Access Configuration* → *Self Service Portal*. Either by modifying the default policy or creating a new one, you'll be able to define the behavior of the device registration page. The portal page can be accessed by the following URL: https://YOUR_PORTAL_HOSTNAME/device-registration This URL is accessible from within the network, in any VLAN that can reach the PacketFence server on a 'portal' interface (see note below).

First, you can decide which role to assign to the devices registered through this self service portal. If left empty, the role of the user who is registering the device will be used.

You can also select which operating systems can be registered through this portal. This is useful for example, if you wish to only allow gaming devices to be registered through this portal.

Once you have configured your self service portal policy, you need to assign it to the appropriate connection profile in *Configuration* → *Policies and Access Control* → *Connection Profiles*.

After this, the page will be accessible at https://YOUR_PORTAL_HOSTNAME/device-registration.

WARNING

You may also have to add the 'portal' listening daemon on your management interface in order for this self service portal to be accessible to your users.

14.4. Status page

Users can have access to a self service portal that allows them to **manage** their devices. Using this portal, they can unregister devices they own and report them as stolen (triggering the **Lost or Stolen** violation).

Users of the local PacketFence database can also change their password through this portal.

By default all users can manage their devices through this self service portal. You can specify which roles can manage their devices on this page by configuring a self service portal policy in *Configuration* → *Advanced Access Configuration* → *Self Service Portal*. Then, make sure you assign this policy to the appropriate connection profile in *Configuration* → *Policies and Access Control* → *Connection Profiles*.

You can also prevent this page from being served in the PacketFence managed networks (registration, isolation, inline) by enabling the parameter **Status URI only on production network** in *Configuration* → *Advanced Access Configuration* → *Captive Portal*.

After this, the page will be accessible at https://YOUR_PORTAL_HOSTNAME/status.

Once you have configured your self service portal policy, you need to assign it to the appropriate connection profile in *Configuration → Policies and Access Control → Connection Profiles*.

WARNING

You may also have to add the 'portal' listening daemon on your management interface in order for this self service portal to be accessible to your users.

14.5. Passthroughs

Passthroughs are used to allow access to certain resources that are outside of the registration confinement process for the users that are in it. A good example would be when you want to allow access to a password reset server even for clients that are currently on the captive portal.

There are two solutions for passthroughs - one using DNS resolution and iptables and the other one using Apache's mod_proxy module. Note that non-HTTP (including HTTPS) protocols cannot use the mod_proxy approach. You can use one of them or both but for if a domain is configured in both, DNS passthroughs have a higher priority.

In order to use the passthroughs feature in PacketFence, you need to enable it from the GUI in *Configuration → Network Configuration → Networks → Fencing*, enable **Passthrough** and then **Save**.

14.5.1. DNS passthroughs

NOTE

In active-active cluster, **pfdns** needs to listen on VIP only. *Configuration → System Configuration → Cluster → pfdns on VIP only*

If you just enabled the passthroughs, you should restart the iptables services after configuring the parameter (`/usr/local/pf/bin/pfcmd service iptables restart`).

Then add passthroughs in *Configuration → Network Configuration → Networks → Fencing → Passthroughs*. They can be of the following format:

- **example.com**: opens ports 80 and 443 in TCP for example.com
- **example.com:1812**: opens the port 1812 in TCP and UDP for example.com
- **example.com:tcp:1812**: opens the port 1812 in TCP for example.com
- **example.com:udp:1812**: opens the port 1812 in UDP for example.com

In addition to the options above, you can prefix the domain with `.` (.example.com) to white list all the subdomains of example.com (ex: **www.example.com**, **my.example.com**).

Should you combine multiple times the same domain with different ports (**example.com**, **example.com:udp:1812**, **example.com:udp:1813**) in the passthroughs, it will open all ports specified in all entries. In the previous example that would open ports 80, 443 in TCP as well as 1812 and 1813 in UDP.

Now when pfdns receives a request for one of these domains, it will reply with the real DNS records for the FQDN instead of a response that points to the captive portal. At the same time, it will add the entry to a special ipset which will allow access to the real IP address attached the FQDN via iptables based routing.

14.5.2. Apache mod_proxy passthroughs

The proxy passthroughs can be configured in *Configuration* → *Network Configuration* → *Networks* → *Fencing* → *Proxy Passthroughs*. Add a new FQDN (can also be a wildcard domain like *.google.com). Port specific passthroughs cannot be used as these only apply to port 80 in TCP. Then for this FQDN, pfdns will still answer with the IP address of the captive portal and when a device hits the captive portal, PacketFence will detect that this FQDN has a passthrough configured in PacketFence and will forward the traffic to mod_proxy.

14.6. Proxy Interception

PacketFence enables you to intercept proxy requests and forward them to the captive portal. It only works on layer-2 networks because PacketFence must be the default gateway. In order to use the Proxy Interception feature, you need to enable it from the GUI in *Configuration* → *Network Configuration* → *Networks* → *Fencing* and check *Proxy Interception*.

Add the port you want to intercept (like 8080 or 3128) and add a new entry in the `/etc/hosts` file to resolve the fully qualified domain name (fqdn) of the captive portal to the IP address of the registration interface. This modification is mandatory in order for Apache to receive the proxy requests.

14.7. Parked Devices

In the event that you are managing a large registration network with devices that stay there (ex: Students that can't register in your environment), these devices consume precious resources and generate useless load on the captive portal and registration DHCP server.

Using the parking feature, you can make these devices have a longer lease and hit an extremely lightweight captive portal so that the amount of resources they consume is minimal. In that captive portal, they will see a message explaining that they haven't registered their device for a certain amount of time, and will let them leave the **parked** state by pressing a link.

The **parked** vs **unparked** state is controlled through security event `1300003` which gets triggered according to the `parking.threshold` setting (*Configuration* → *Network Configuration* → *Networks* → *Device Parking*).

So, in order to activate the parking, go in *Configuration* → *Network Configuration* → *Networks* → *Device Parking* and set the threshold to a certain amount of seconds. A suggested value would be `21600` which is 6 hours. This means that if a device stays in your registration network for more than 6 hours in a row, it will trigger security event `1300003` and place that device into the **parked** state.

In that same section, you can define the lease length of the user when he is in the **parked** state.

NOTE

Parking is detected when a device asks for DHCP, if PacketFence is not your DHCP server for the registration network, this feature will not work. Also, if the device goes into the parked state with a lease time of 1 hour and the user immediately releases himself from the parking state, it will take 1 hour before the next detection takes place even if you set `parking.threshold` to a lower value.

14.7.1. Security Event 1300003

This security event controls what happens when a user is detected doing parking.

Here are the main settings:

- You can add actions to the predefined ones (like 'Email admin' or 'External action') in *Definition* → *Actions*
- The amount of time a user can **unpark** their device is controlled through the *Remediation* → *Max enable* setting.
- The amount of grace time between two parking security events is controlled by the *Remediation* → *Grace* setting. This means, once a user release himself from the **parked** state, he will have at least this amount of time to register before the parking triggers again.
- The destination role (thus VLAN) of the user is controlled by *Advanced* → *Role*. You should leave the user in the registration role, but should you want to dedicate a role for parking, you can set it there.
- The **Template** attribute will only be used when the user is on the normal PacketFence portal and not the one dedicated for parking. If you want the user to access the non-parking portal, disable **Show parking portal** in *Configuration* → *Network Configuration* → *Networks* → *Device Parking*

15. Advanced Access Configuration

15.1. Connection Profiles

PacketFence comes with a default connection profile. The follow parameters are important to configure no matter if you use the default connection profile or create a new one:

- Redirect URL under *Configuration* → *Policies and Access Control* → *Connection Profile* → *Profile Name*

For some browsers, it is preferable to redirect the user to a specific URL instead of the URL the user originally intended to visit. For these browsers, the URL defined in `redirecturl` will be the one where the user will be redirected. Affected browsers are Firefox 3 and later.

- IP under *Configuration* → *Advanced Access Configuration* → *Captive portal*.

This IP is used as the web server who hosts the `common/network-access-detection.gif` which is used to detect if network access was enabled. It cannot be a domain name since it is used in registration or quarantine where DNS is black-holed. It is recommended that you allow your users to reach your PacketFence server and put your LAN's PacketFence IP. By default we will make this reach PacketFence's website as an easier and more accessible solution.

In some cases, you may want to present a different captive portal (see below for the available customizations) according to the SSID, the VLAN, the switch IP/MAC or the URI the client connects to. To do so, PacketFence has the concept of connection profiles which gives you this possibility.

When configured, connection profiles will override default values for which it is configured. When no values are configured in the profile, PacketFence will take its default ones (according to the "default" connection profile).

Here are the different configuration parameters that can be set for each connection profiles. The only mandatory parameter is "filter", otherwise, PacketFence won't be able to correctly apply the connection profile. The parameters must be set in `conf/profiles.conf`:

`conf/profiles.conf`

```
[profilename1]
description = the description of your connection profile
filter = the name of the SSID for which you'd like to apply the profile, or the
VLAN number
sources = comma-separated list of authentications sources (IDs) to use
```

Connection profiles should be managed from PacketFence's Web administrative GUI - from the *Configuration* → *Policies and Access Control* → *Connection Profiles* section. Adding a connection profile from that interface will correctly copy templates over - which can then be modified as you wish.

- Filters under *Configuration* → *Policies and Access Control* → *Connection Profile* → *Profile Name* → *Filters*

PacketFence offers the following filters: Connection Type, Network, Node Role, Port, realm, SSID, Switch, Switch Port, URI, VLAN and Time period.

Example with the most common ones:

- **SSID:** Guest-SSID
- **VLAN:** 100
- **Time period:** wd {Mon Tue} hr {1pm-3pm} — See <http://search.cpan.org/~pryan/Period-1.20/Period.pm>
- **Switch Port:** <SwitchId>-<Port>
- **Network:** Network in CIDR format or an IP address

CAUTION

Node role will take effect only with a 802.1X connection or if you use VLAN filters.

- Advanced filter under *Configuration* → *Policies and Access Control* → *Connection Profile* → *Profile Name* → *Advanced Filter*

In this section you are able to define an advanced filter in order to match specific attributes.

You have access to the following attributes:

From the database (mean that it comes from a previous connection):

```
autoreg
status
bypass_vlan
bandwidth_balance
regdate
bypass_role
device_class
device_type
device_version
device_score
pid
machine_account
category
mac
last_arp
lastskip
last_dhcp
user_agent
computername
dhcp_fingerprint
detect_date
voip
notes
time_balance
sessionid
dhcp_vendor
unregdate
fingerbank_info.device_name
fingerbank_info.device_fq
fingerbank_info.device_hierarchy_names
fingerbank_info.device_hierarchy_ids
fingerbank_info.score
fingerbank_info.version
fingerbank_info.mobile
radius_request.User-Name
radius_request.NAS-IP-Address
radius_request.NAS-Port-Id
```

From the current connection:

```
connection_sub_type
connection_type
switch
port
vlan
ssid
dot1x_username
realm
machine_account
```

Operator:

```
&& and
|| or
!= is not equal
== equal
()
```

Special value

```
__NULL__ the value is NULL in the database
```

Examples (Match machine authentication on secure wireless ssid)

- `machine_account != "" && connection_type == Wireless-802.11-EAP`

Examples (Match a device that did machine authentication in a previous connection and connect on ssid Secure)

- `machine_account != "" && ssid == Secure`

Examples (Match a device that does user authentication and did machine authentication on a secure ssid)

- `last_connection_type == "Wireless-802.11-EAP" && machine_account != "" && last_dot1x_username !~ "^host/"`

Examples (Match a device that does user authentication and never did machine authentication on a secure ssid)

- `last_connection_type == "Wireless-802.11-EAP" && (machine_account == "" || machine_account == __NULL__) && last_dot1x_username !~ "^host/"`

Examples (Match a device that never did a machine authentication (BYOD))

- `machine_account == __NULL__`

Here and example of attributes that can be tested:

```

'radius_request' => {
    'NAS-Port-Type' => 15,
    'Service-Type' => 2,
    'State' =>
'0x7cfd15627dba0f5a45baee16526652a6',
    'Called-Station-Id' => '00:8e:73:5d:f6:9e',
    'FreeRADIUS-Proxied-To' => '127.0.0.1',
    'Realm' => 'null',
    'EAP-Type' => 26,
    'NAS-IP-Address' => '172.30.255.13',
    'NAS-Port-Id' => 'GigabitEthernet1/0/30',
    'SQL-User-Name' => 'gwten',
    'Calling-Station-Id' => '00:11:22:33:44:55',
    'PacketFence-Domain' => 'ZAYM',
    'Cisco-AVPair' => 'service-type=Framed',
    'User-Name' => 'zaym',
    'Event-Timestamp' => 'Aug 15 2019 17:10:03
BST',
    'EAP-Message' => '0x024700061a03',
    'Framed-IP-Address' => '172.30.250.149',
    'NAS-Port' => 50130,
    'Stripped-User-Name' => 'gwten',
    'Framed-MTU' => 1500
},
'autoreg' => 'yes',
'last_port' => '37',
'device_class' => 'Windows OS',
'bandwidth_balance' => undef,
'bypass_role' => undef,
'device_type' => 'Windows OS',
'pid' => 'gwten',
'dhcp6_enterprise' => '',
'last_seen' => \[
    'NOW()'
],
'dhcp6_fingerprint' => '',
'category' => 'Wire',
'mac' => '00:11:22:33:44:55',
'portal' => 'Wire',
'lastskip' => '0000-00-00 00:00:00',
'eap_type' => 26,
'last_dhcp' => '0000-00-00 00:00:00',
'user_agent' => 'ccmhttp',
'computername' => 'zamtop',
'dhcp_fingerprint' => '1,15,3,6,44,46,47,31,33,121,249,43',
'detect_date' => '2019-08-15 15:33:30',
'last_vlan' => '0',
'last_connection_sub_type' => 26,

```

```

'fingerbank_info' => {
    'device_fq' => 'Operating System/Windows OS',
    'device_name' => 'Windows OS',
    'version' => '',
    'score' => '73',
    'mobile' => 0,
    'device_hierarchy_names' => [
        'Windows OS',
        'Operating System'
    ],
    'device_hierarchy_ids' => [
        1,
        16879
    ]
},
'bypass_role_id' => undef,
'last_role' => 'Wire',
'dhcp_vendor' => 'MSFT 5.0',
'unregdate' => '2019-08-15 20:10:04',
'last_switch' => '172.20.20.1',
'auto_registered' => 1,
'__from_table' => 1,
'source' => 'Wire',
'last_ifDesc' => 'GigabitEthernet1/0/30',
'device_version' => '',
'status' => 'reg',
'bypass_vlan' => undef,
'regdate' => '2019-08-15 17:10:04',
'last_dot1x_username' => 'zayme',
'tenant_id' => '1',
'category_id' => '166',
'machine_account' => '',
'last_connection_type' => 'Ethernet-EAP',
'last_ssid' => '',
'realm' => 'null',
'last_ip' => '172.20.20.2',
'device_score' => '73',
'last_arp' => '0000-00-00 00:00:00',
'last_start_timestamp' => '1565885356',
'stripped_user_name' => 'zayme',
'__old_data' => {
    'autoreg' => 'yes',
    'device_class' => 'Windows OS',
    'bandwidth_balance' => undef,
    'bypass_role' => undef,
    'device_type' => 'Windows OS',
    'pid' => 'gwten',
    'dhcp6_enterprise' => '',

```



```

        'last_seen' => '2019-08-15 16:09:16',
        'dhcp6_fingerprint' => '',
        'category' => 'Wire',
        'mac' => '00:11:22:33:44:55',
        'lastskip' => '0000-00-00 00:00:00',
        'last_dhcp' => '0000-00-00 00:00:00',
        'user_agent' => 'ccmhttp',
        'dhcp_fingerprint' =>
'1,15,3,6,44,46,47,31,33,121,249,43',
        'computername' => 'zamtop',
        'detect_date' => '2019-08-15 15:33:30',
        'bypass_role_id' => undef,
        'dhcp_vendor' => 'MSFT 5.0',
        'unregdate' => '2019-08-15 20:09:16',
        'device_version' => '',
        'status' => 'reg',
        'bypass_vlan' => undef,
        'regdate' => '2019-08-15 17:09:16',
        'category_id' => '166',
        'tenant_id' => '1',
        'machine_account' => undef,
        'last_arp' => '0000-00-00 00:00:00',
        'device_score' => '73',
        'voip' => 'no',
        'device_manufacturer' => 'Toshiba',
        'notes' => 'AUTO-REGISTERED',
        'time_balance' => undef,
        'sessionid' => undef
    },
    'voip' => 'no',
    'device_manufacturer' => 'Toshiba',
    'notes' => 'AUTO-REGISTERED',
    'time_balance' => undef,
    'last_switch_mac' => '00:8e:73:5d:f6:9e',
    'sessionid' => undef,
    'last_start_time' => '2019-08-15 16:09:16'

```

PacketFence relies extensively on Apache for its captive portal, administrative interface and Web services. The PacketFence Apache configuration is located in </usr/local/pf/conf/httpd.conf.d/>.

In this directory you have three important files: [httpd.admin](#), [httpd.portal](#), [httpd.webservices](#), [httpd.aaa](#).

- [httpd.admin](#) is used to manage PacketFence admin interface
- [httpd.portal](#) is used to manage PacketFence captive portal interface
- [httpd.webservices](#) is used to manage PacketFence webservices interface
- [httpd.aaa](#) is use to manage incoming RADIUS request

These files have been written using the Perl language and are completely dynamic - so they activate services only on the network interfaces provided for this purpose.

The other files in this directory are managed by PacketFence using templates, so it is easy to modify these files based on your configuration. SSL is enabled by default to secure access.

Upon PacketFence installation, self-signed certificates will be created in `/usr/local/pf/conf/ssl/` (`server.key` and `server.crt`). Those certificates can be replaced anytime by your 3rd-party or existing wild card certificate without problems. Please note that the CN (Common Name) needs to be the same as the one defined in the PacketFence configuration file (`pf.conf`).

15.1.1. Reuse 802.1X credentials

Under certain circumstances, for example to show an AUP after a successful 802.1X connection, it might be interesting to have the ability to use an "SSO emulation" in the sense that the user does not need to re-enter his credentials on the portal after having entered them during the 802.1X EAP process. The 'Reuse 802.1X credentials' connection profile option will address this purpose. The same username as the one used during the 802.1X connection will be used against the different connection profile authentication sources to recompute the role from the portal.

As a security precaution, this option will only reuse 802.1X credentials if there is an authentication source matching the provided realm. This means, if users use 802.1X credentials with a domain part (`username@domain`, `domain\username`), the domain part needs to be configured as a realm under the RADIUS section and an authentication source needs to be configured for that realm. If users do not use 802.1X credentials with a domain part, only the NULL realm will be match IF an authentication source is configured for it.

15.2. Filter Engine Macro

In each filter engines you have the possibility to use some macro in the text field:

```
uc
lc
join
substr
macToEUI48
random_from_range
log
replace
BuildFromMatch
```

15.2.1. uc

This macro return the upper case of a string

Example:

```
PacketFence-UserName = ${uc($radius_request.Calling-Station-Id)}
```

will assign to PacketFence-UserName the upper case value of Calling-Station-Id.

```
Calling-Station-Id = "00:10:7f:38:89:9d" -> PacketFence-UserName =  
"00:10:7F:38:89:9D"
```

15.2.2. lc

This macro return the lower case of a string

Example:

```
PacketFence-UserName = ${lc($radius_request.User-Name)}
```

will assign to PacketFence-UserName the lower case value of User-Name.

```
User-Name = "ZAMMIT" -> PacketFence-UserName = "zammit"
```

15.2.3. join

This macro return the join of a string

Example:

```
PacketFence-UserName = ${join(":",$radius_request.User-Name,"Super")}
```

will assign to PacketFence-UserName a join of the separator and the values.

```
User-Name = "bobey" -> PacketFence-UserName = "bobey:Super"
```

15.2.4. substr

This macro will return a substring

```
PacketFence-UserName = ${substr($radius_request.User-Name,0, 5)}
```

will assign to PacketFence-UserName the substring

```
User-Name = "ZammitLudovic" -> PacketFence-UserName = "Zammit"
```

15.2.5. macToEUI48

This macro will return the EUI48 format of a MAC address

```
PacketFence-UserName = ${macToEUI48($radius_request.Calling-Station-Id)}
```

will assign to PacketFence-UserName the EUI48 MAC address

```
Calling-Station-Id = "00:10:7f:38:89:9d" -> PacketFence-UserName = "00-10-7F-38-89-9D"
```

15.2.6. random_from_range

This macro will return a random number from a range

```
Session-Timeout = ${random_from_range("10620..12600")}
```

will assign a random number between 10620 and 12600 to Session-Timeout

```
Session-Timeout = 11343
```

15.2.7. log

This macro will log a message in the packetfence.log file

```
PacketFence-UserName = ${log($radius_request.User-Name." logged")}
```

will log the value of the RADIUS request attribute User-Name and logged

```
User-Name = "zammit" -> "Zammit logged"
```

15.2.8. replace

This macro will replace a string by another one

```
PacketFence-UserName = ${replace($radius_request.User-Name,"z","r")}
```

will replace the letter z by the letter r in the string

```
User-Name = "zabbit" -> PacketFence-UserName = "rabbit"
```

15.2.9. BuildFromMatch

This macro will apply a regular expression on the attribute value to be able to build another value

```
TLS-Stripped-UserName = ${BuildFromMatch($radius_request.TLS-Client-Cert-Common-Name,"^[^@]+","$0")}
```

will extract the value before the @ sign and assign this value

```
TLS-Client-Cert-Common-Name = "zammit@packetfence.org" -> TLS-Stripped-UserName = "zammit"
```

15.3. VLAN Filter Definition

We added the ability to specify filters directly in the portion of code that re-evaluates the VLAN or do a call to the API when we receive a RADIUS request. These filters can be defined in *Configuration* → *Advanced Access Configuration* → *Filter engines*.

These rules are available in different scopes:

```
IsolationRole  
RegistrationRole  
RegisteredRole  
InlineRole  
AutoRegister  
NodeInfoForAutoReg
```

And can be defined using different criteria like:

```
node_info.attribute (like node_info.status)  
switch  
ifIndex  
mac  
connection_type  
username  
ssid  
time  
owner.attribute (like owner.pid)  
radius_request.attribute (like radius_request.Calling-Station-Id)
```

There are some default VLAN filters defined in the configuration you can use to achieve the following goal:

EXAMPLE_Reject_between_11am_2pm

will revert a device from connecting when its role is "default", the SSID is "SECURE", the current time is between 11am and 2pm, from Monday to Friday and is a registered device

EXAMPLE_Trigger_event_if_user

will create a security event if the SSID is OPEN and the owner is igmout (the security event

needs to have a custom trigger with the value 12345)

EXAMPLE_Autoregister_if_user

will autoregister the device and assign the role staff to each device where the username is igmout.

EXAMPLE_Autoregister_windows_devices

will autoregister all Windows devices and assign them the default role.

EXAMPLE_Reject_specific_MAC

will filter a MAC address and reject it by assigning the REJECT role.

EXAMPLE_Detect_VOIP

will automatically set Avaya and Polycom as phones by matching vendor MAC and set to default role

EXAMPLE_Reject_User_Unless_Machine

will refuse user authentication without prior machine authentication

EXAMPLE_Autoregister_Printer_Scanner

will autoregister printers and scanners and add a note.

You can have a look in the file `/usr/local/pf/conf/vlan_filters.conf`, there are some examples on how to use and define filters.

15.4. RADIUS Filter Definition

We added the ability to specify filters directly in the portion of code that return RADIUS attributes or do a call to the API. These filters can be defined in *Configuration → Advanced Access Configuration → Filter engines*.

These rules are available in thoses scopes:

```
returnRadiusAccessAccept is when you return the answer for a device access
returnAuthorizeRead is when you return the answer for the switch read login
access
returnAuthorizeWrite is when you return the answer for the switch write login
access
returnAuthorizeVoip is when you return the answer for a VoIP device
preProcess is when you want to manipulate the RADIUS context (like adding
custom attributes to the request)
```

```

packetfence.authorize call the RADIUS filter in the packetfence authorize
section
packetfence.authenticate call the RADIUS filter in the packetfence authenticate
section
packetfence.pre-proxy call the RADIUS filter in the packetfence pre-proxy
section
packetfence.post-proxy call the RADIUS filter in the packetfence post-proxy
section
packetfence-tunnel.authorize call the RADIUS filter in the packetfence-tunnel
authorize section
packetfence.precacct call the RADIUS filter in the packetfence precacct section
packetfence.accounting call the RADIUS filter in the packetfence accounting
section
eduroam.authorize call the RADIUS filter in the eduroam accounting section
eduroam.pre-proxy call the RADIUS filter in the pre-proxy accounting section
eduroam.post-proxy call the RADIUS filter in the post-proxy accounting section
eduroam.precacct call the RADIUS filter in the eduroam precacct section

```

All the packetfence.* eduroam.* scopes are covered in the file radius_filters.conf, this is advanced configuration and you must know what you are doing.

And can be defined using different criteria like:

```

node_info.attribute (like node_info.$attribute)
switch
ifIndex
mac
connection_type
username
ssid
time
owner.attribute (like owner.$attribute)
radius_request.attribute (like radius_request.$attribute)
security_event
user_role
vlan

```

There are some default RADIUS filters defined in the configuration you can use to achieve the following goal:

EXAMPLE_Ethernet-EAP-Accept

will return Access-Accept (with Cisco-AVPair attribute) when the connection is Ethernet-EAP and when there is no security event.

EXAMPLE_Session-timeout_Idle-Timeout_Terminate_action

will filter on the switch IP addresses and add the Session-Timeout (with a value between 10620 and 12600), the Idle-Timeout and Terminate-Action RADIUS attributes.

EXAMPLE_ipad_by_name

will use Fingerbank to target a specific devices (Apple iPad) and will add a Cisco ACLs to them.

EXAMPLE_eap-tls-preProcess

will create internal RADIUS attributes that will be used internally (like in the authentication rules). This rule will add the TLS-Stripped-UserName RADIUS attribute in the request and you will be able to use it in the authentication/administrations rules.

You can have a look in the file `radius_filters.conf`, there are some examples on how to use and define filters.

15.5. Advanced LDAP Authentication

15.5.1. ldapfilter action

ldapfilter action overrides the internal LDAP filter that PacketFence creates internally (`uid=$username`) so you can create a custom filter that matches your needs.

For example something like this (search for the user and check to see if it's permitted based on some criteria):

```
(&(|(cn=${radius_request.Stripped-User-Name})(cn=${radius_request.User-Name}))( |(permitWifi=*)(grade=staff)(memberOf=CN=WifiGroup,OU=Security Groups,DC=ad,DC=acme,DC=com)))
```

15.5.2. Set_role_on_not_found action

`set_role_on_not_found` is a way to define a role if the rule doesn't match, let's take the `ldapfilter` example above.

If we add the action `set_role_on_not_found = REJECT` so it mean that the device will be rejected if the LDAP filter doesn't return anything. (if it matches then `set_role` action will be applied)

15.5.3. role_from_source action

`role_from_source` will check to see if the LDAP attribute exists and will add it in the `ldap_attribute` context (available in the radius filters)

So for example if you want to take the value of the LDAP attribute `customRadius` and add it in the RADIUS answer you need to do the following.

In the authentication rule, set an action "Role from source" to `customRadius`. Next create a RADIUS filter that will add the custom RADIUS attributes:


```
[IF_SET_ROLE_FROM_SOURCE]
status=enabled
answer.0=reply:Packetfence-Raw = $ldap_attribute.customRadius
top_op=and
description=If the role has been computed from the action set_role_from_source
then return the value of the role as a RADIUS attribute
scopes=returnRadiusAccessAccept
radius_status=RLM_MODULE_OK
merge_answer=no
condition=action == "set_role_from_source"
```

Note, this supports multiples LDAP attributes, like customRadius:Airespace-Interface-Name=internet , customRadius:Aruba-User-Vlan=666

15.5.4. Append search attributes LDAP filter

This option will add as a & condition to the LDAP filter generated by PacketFence, So for example the ldapfilter generated by PacketFence will be the following:

```
(&( |(sAMAccountName=%{User-Name})(sAMAccountName=%{Stripped-User-
Name})(cn=%{User-Name})(cn=%{Stripped-User-Name})(sAMAccountName=%{%{Stripped-
User-Name}}:-%{User-Name})))
```

and if you define a appended LDAP filter like:

```
( |(memberOf=CN=Staff,OU=Security Groups,DC=ad,DC=acme,DC=com)(wifi=enabled))
```

Then the filter will be generated like that:

```
(&( |(sAMAccountName=%{User-Name})(sAMAccountName=%{Stripped-User-
Name})(cn=%{User-Name})(cn=%{Stripped-User-Name})(sAMAccountName=%{%{Stripped-
User-Name}}:-%{User-Name}))) ( |(memberOf=CN=Staff,OU=Security
Groups,DC=ad,DC=acme,DC=com)(wifi=enabled)))
```

By doing that, event if you wont don't need the "Search Attributes" feature, you will be able to store the user's DN in the PacketFence-UserDN attribute.

15.5.5. basedn condition

This condition allow to override the default basedn in the LDAP source and it will permit to test if a object is in a specific ou.

15.6. Advanced Realm Configuration

In PacketFence you can define multiple realms to select on which domain you want to

authenticate the users.

You can define a Realm with a regex in order to match multiple formats.

For example in the ACME realm we define the regex like this:

```
.*\.acme\.com$
```

It means that if you have a user coming with this username mickey@la.acme.com , PacketFence will define the realm as la.acme.com (it will be included in the RADIUS request) and PacketFence will map the user to the ACME realm.

16. Advanced RADIUS Configuration

This section presents the FreeRADIUS configuration steps. In some occasions, a RADIUS server is mandatory in order to give access to the network. For example, the usage of WPA2-Enterprise (Wireless 802.1X), MAC authentication and Wired 802.1X all require a RADIUS server to authenticate the users and the devices, and then to push the proper roles or VLAN attributes to the network equipment.

16.1. Local Authentication

Add your user's entries at the end of the `/usr/local/pf/raddb/users` file with the following format:

```
username Cleartext-Password := "password"
```

16.2. Authentication against Active Directory (AD)

To perform EAP-PEAP authentication using Microsoft Active Directory, please refer to the Active Directory documentation from the Authentication Mechanism section.

16.3. EAP Authentication against OpenLDAP

To authenticate 802.1X connection against OpenLDAP you need to define the LDAP connection in `/usr/local/pf/raddb/modules/ldap` and be sure that the user password is define as a NTHASH or as clear text.

```

pfcron
  ldap openldap {
    server = "ldap.acme.com"
    identity = "uid=admin,dc=acme,dc=com"
    password = "password"
    basedn = "dc=district,dc=acme,dc=com"
    filter = "(uid=%{mschap:User-Name})"
    ldap_connections_number = 5
    timeout = 4
    timelimit = 3
    net_timeout = 1
    tls {
    }
    dictionary_mapping = ${confdir}/ldap.attrmap
    edir_account_policy_check = no

    keepalive {
      # LDAP_OPT_X_KEEPALIVE_IDLE
      idle = 60

      # LDAP_OPT_X_KEEPALIVE_PROBES
      probes = 3

      # LDAP_OPT_X_KEEPALIVE_INTERVAL
      interval = 3
    }
  }
}

```

Next in `/usr/local/pf/raddb/sites-available/packetfence-tunnel` add in the authorize section:

```

authorize {
  suffix
  ntdomain
  eap {
    ok = return
  }
  files
  openldap
}

```

16.4. EAP Guest Authentication on Email, Sponsor and SMS Registration

This section will allow local credentials created during guest registration to be used in 802.1X

EAP-PEAP connections.

NOTE

Be sure to select `plaintext` or `ntlm` as the "Database passwords hashing method" to make it work.

First create a guest SSID with the guest access you want to use (Email, Sponsor or SMS, ...) and activate 'Create local account' on that source.

At the end of the guest registration, PacketFence will send an email with the credentials for Email and Sponsor and SMS.

NOTE

This option doesn't currently work with the **Reuse dot1x credentials** option of the captive portal.

To enable this feature, go in 'Configuration→System Configuration→RADIUS→General' and enable 'Authenticate against local users database'. Once saved, restart the `radiusd` service.

16.5. EAP Local User Authentication

The goal here is to use the local user to authenticate 802.1X device.

To enable this feature, go in 'Configuration→System Configuration→RADIUS→General' and enable 'Authenticate against local users database'. Once saved, restart the `radiusd` service.

CAUTION

Take care of the "Database passwords hashing method" that has been configured in *Configuration → System Configuration → Main Configuration → Advanced* or in the authentication source configuration (when you enabled "create local account") , the hash method must be `plaintext` or `ntlm` to be able to work.

16.6. Limit Brute Force EAP Authentication

This section will allow you to limit a brute force attack and prevent the locking of Active Directory accounts.

Edit `/usr/local/pf/conf/radiusd/packetfence-tunnel`

```
# Uncomment the following lines to enable this feature
packetfence-control-ntlm-failure
packetfence-cache-ntlm-hit
```

By default it will reject for 5 minutes a device that has been rejected twice in the last 5 minutes. Feel free to change the default values in `raddb/policy.d/packetfence` and in `raddb/mods-enabled/cache_ntlm`

16.7. Testing

Test your setup with `radtest` using the following command and make sure you get an **Access-Accept** answer:

```
# radtest dd9999 Abcd1234 localhost:18120 12 testing123
Sending Access-Request of id 74 to 127.0.0.1 port 18120
  User-Name = "dd9999"
  User-Password = "Abcd1234"
  NAS-IP-Address = 255.255.255.255
  NAS-Port = 12
rad_recv: Access-Accept packet from host 127.0.0.1:18120, id=74, length=20
```

16.8. RADIUS Accounting

RADIUS Accounting is usually used by ISPs to bill clients. In PacketFence, we are able to use this information to determine if the node is still connected, how much time it has been connected, and how much bandwidth the user consumed.

PacketFence uses RADIUS Accounting to display Online/Offline status in webadmin in *Nodes* menu.

16.8.1. IP log updates

If you send the IP address of nodes in accounting data and want to update iplog entries of your nodes, you can enable 'Update the iplog using the accounting' setting from *Configuration* → *System configuration* → *Main configuration* → *Advanced*.

16.8.2. Security Events

Using PacketFence, it is possible to add security events to limit bandwidth abuse. The format of the trigger is very simple:

```
Accounting::[DIRECTION][LIMIT][INTERVAL(optional)]
```

Let's explain each chunk properly:

- **DIRECTION**: You can either set a limit to inbound(IN), outbound(OUT), or total(TOT) bandwidth
- **LIMIT**: You can set a number of bytes(B), kilobytes(KB), megabytes(MB), gigabytes(GB), or petabytes(PB)
- **INTERVAL**: This is actually the time window we will look for potential abuse. You can set a number of days(D), weeks(W), months(M), or years(Y).

Example triggers

- Look for Incoming (Download) traffic with a 50GB/month

```
Accounting::IN50GB1M
```

- Look for Outgoing (Upload) traffic with a 500MB/day

```
Accounting::OUT500MB1D
```

- Look for Total (Download + Upload) traffic with a 200GB limit in the last week

```
Accounting::TOT200GB1W
```

Grace Period

When using such security event feature, setting the grace period is really important. You don't want to put it too low (ie. A user re-enable his network, and get caught after 1 bytes is transmitted!) or too high. We recommend that you set the grace period to one interval window.

16.9. RADIUS Proxy

RADIUS Proxy is a way to proxy authentication and accounting requests to other radius server(s) based on the realm. Let's say you want to authenticate users on an Active Directory where there is a NPS server running and you don't want to join the PacketFence's server to this domain or in the case you want to integrate PacketFence in a Passpoint setup then this section is for you.

To do that in PacketFence you need first to define the target RADIUS server(s) in *Configuration* → *Policies and Access Control* → *Authentication Sources*, and create the RADIUS source(s) (ACME1 ACME2). In the Source configuration, fill the mandatory fields and add the options to define the home_server in FreeRADIUS. (<https://github.com/FreeRADIUS/freeradius-server/blob/v3.0.x/raddb/proxy.conf>)

Per example for the RADIUS Source ACME1:

The screenshot shows the 'New Authentication Source' configuration page in the FreeRADIUS web interface. The page has a dark header with navigation links: Status, Reports, Auditing, Nodes, Users, Configuration. The user is logged in as 'admin'. The left sidebar shows a navigation menu with categories like Policies and Access Control, Compliance, Integration, Advanced Access Configuration, Network Configuration, and System Configuration. The main content area is titled 'New Authentication Source' and contains the following fields:

- Name:** ACME1
- Description:** Radius Server 1
- Host:** 192.168.0.20
- Port:** 1812
- Secret:** [Redacted]
- Timeout:** 1
- Monitor:** Disabled (toggle)
- Options:**

```

type = auth+acct
response_window = 6
status_check = status-server
revive_interval = 120
check_interval = 30
num_answers_to_alive = 3
src_ipaddr = $src_ip

```
- Associated Realms:** [Dropdown menu]
- Authentication Rules:** [Add Rule button]
- Administration Rules:** [Add Rule button]

At the bottom of the form are 'Create' and 'Reset' buttons.

\$src_ip is a way to dynamically use the correct source ip address of the system in case of multiples network interfaces.

Next go in *Configuration* → *Policies and Access Control* → *REALMS*, and add a new realm.

[Status](#)
[Reports](#)
[Auditing](#)
[Nodes](#)
[Users](#)
[Configuration](#)

[API](#)
[dashboard](#)

- Policies and Access Control** ▼
- Roles
- Domains
 - Active Directory Domains
 - Realms
- Authentication Sources
- Network Devices
 - Switches
 - Switch Groups
- Connection Profiles
- Compliance** ^
- Integration** ^
- Advanced Access Configuration** ^
- Network Configuration** ^
- System Configuration** ^

New Realm ✕

Realm

NTLM Auth Configuration

Domain

The domain to use for the authentication in that realm.

Freeradius Proxy Configuration

Realm Options

You can add FreeRADIUS options in the realm definition.

RADIUS AUTH

The RADIUS Server(s) to proxy authentication.

Type

Home server pool type.

Authorize from PacketFence

Should we forward the request to PacketFence to have a dynamic answer or do we use the remote proxy server answered attributes?

RADIUS ACCT

The RADIUS Server(s) to proxy accounting.

Type

Home server pool type.

Freeradius Eduroam Proxy Configuration

Eduroam Realm Options

You can add Eduroam FreeRADIUS options in the realm definition.

Eduroam RADIUS AUTH

The RADIUS Server(s) to proxy authentication.

Type

Home server pool type.

Authorize from PacketFence

Should we forward the request to PacketFence to have a dynamic answer or do we use the remote proxy server answered attributes?

Eduroam RADIUS ACCT

The RADIUS Server(s) to proxy accounting.

Type

Home server pool type.

Stripping Configuration

Strip on the portal

Should the usernames matching this realm be stripped when used on the captive portal.

Strip on the admin

Should the usernames matching this realm be stripped when used on the administration interface.

Strip in RADIUS authorization

Should the usernames matching this realm be stripped when used in the authorization phase of 802.1x. Note that this doesn't control the stripping in FreeRADIUS, use the options above for that.

Custom attributes

Allow to use custom attributes to authenticate 802.1x users (attributes are defined in the source).

LDAP source

The LDAP Server to query the custom attributes.

(type definition can be found here <https://wiki.freeradius.org/features/Proxy>)

Authorize from PacketFence will send the request to PacketFence to compute the role and access duration of the device.

In this case the easiest way to achieve that is to create a Authorization source (with rules), assign this source to a connection profile where you enabled "Automatically register devices" and where you defined a filter like Realm = acme.com .

Click on **Save** and restart radiusd service.

```
/usr/local/pf/bin/pfcmd service radiusd restart
```

Now when a device connect with the username [bob@acme.com](#) then the authentication and accounting requests will be forwarded to one of the ACME RADIUS servers.

16.10. RADIUS EAP Profiles

RADIUS EAP Profiles allow you to select a specific EAP profile in PacketFence based on the realm of the user.

In this EAP profile you can define: Certificates configuration. OCSP configuration EAP-Fast configuration TLS Configuration

And link all these configuration together.

For example the realm ACME.COM needs to use the CA certificate from ACME CA and the other realms need to use the default one.

To do that go in *Configuration* → *System Configuration* → *RADIUS* → *SSL Certificates* and create a new profile. Next go in *Configuration* → *System Configuration* → *RADIUS* → *TLS Profiles* and create a new TLS profile and select the Certificate profile created just before. Then create the EAP profile in *Configuration* → *System Configuration* → *RADIUS* → *EAP Profiles* and create a new EAP profile and select the TLS profile created before (PEAP Profile for exemple)

The last thing to do is to link the EAP profile with your realm configuration, to achieve that go in *Configuration* → *Policies and Access Control* → *Domains* → *REALMS* and edit the ACME.COM realm (create it if it's not already the case) then choose the EAP profile you created before in the EAP configuration parameter.

Restart packetfence-radiusd-auth.service to generate the new RADIUS configuration. (systemctl restart packetfence-radiusd-auth.service)

17. Fingerbank Integration

Fingerbank, a great device profiling tool developed alongside of PacketFence, now integrates with it to power-up the feature set allowing a PacketFence administrator to easily trigger security events based on different device types, device parents, DHCP fingerprints, DHCP vendor IDs, MAC vendors and browser user agents.

The core of that integration resides in the ability for a PacketFence system, to interact with the Fingerbank upstream project, which then allow a daily basis fingerprints database update, sharing unknown data so that more complex algorithms can process that new data to integrate it in the global database, querying the global upstream database in the case of an unknown match and much more.

Since the Fingerbank integration is now the "de facto" device profiling tool of PacketFence, it was a requirement to make it as simple as possible to configure and to use. From the moment a working PacketFence system is in place, Fingerbank is also ready to be used, but only in a "local" mode, which means, no interaction with the upstream Fingerbank project.

17.1. Onboarding

To benefit from all the advantages of the Fingerbank project, the onboarding step is required to create an API key that will then allow interaction with the upstream project. That can easily be done only by going in the "Settings" menu item under the "Fingerbank" section of the PacketFence "Configuration" tab. From there, an easy process to create and save an user/organization specific API key can be followed. Once completed, the full feature set of Fingerbank can be used.

17.2. Update Fingerbank Database

Updating the Fingerbank data can't be easier. The only requirement is the onboarding process which allows you to interact with upstream project. Once done, an option to "Update Fingerbank DB" can be found on top of every menu item sections under "Fingerbank". Process may take a minute or two, depending on the size of the database and the Internet connectivity, after which a success or error message will be show accordingly. "Local" records are NOT being modified during this process.

17.3. Submit Unknown Data

Saying that we don't know everything is not false modesty. In that sense, the "Submit Unknown/Unmatched Fingerprints" option is made available (after onboarding) so that unknown fingerprinting data going in and out on your network can easily be submitted to the upstream Fingerbank project for further analysis and integration the in the global database.

17.4. Upstream Interrogation

By default, PacketFence is configured to interrogate the upstream Fingerbank project (if

onboarding has been completed) to fulfill a query with unmatched local results. Unmatched local results can result of an older version of the Fingerbank database or a requirement for a more complex algorithm due to the data set. That behavior is completely transparent and can be modified using the "Settings" menu item under the "Fingerbank" section of the PacketFence "Configuration" tab.

17.5. Local Entries

It is possible for an administrator who wants to customize an existing record (or create a new one) to do so using the "Local" entries. An upstream record (DHCP Fingerprint, DHCP Vendor, MAC Vendor, User Agent, Device type, even a Combination) can be cloned and then modified on a local basis if needed. Local records are always matched first since their purpose is to 'override' an existing one. A local combination can be created to match either "Local" or "Upstream" or both entries to allow identification of a device.

17.6. Settings

Fingerbank settings can easily be modified from the "Settings" menu item under the "Fingerbank" section of the PacketFence "Configuration" tab. There's documentation for each and every parameter that allow easier understanding.

17.7. Device change detection

Using Fingerbank, you can perform detection of potential MAC spoofing by seeing if a device changes from a device class to another (ex: a device goes from Windows to a printer) and trigger a security event and potentially isolate the endpoint. An example security event using this trigger is available (security event ID 1300006, name "Fingerbank device class change").

This feature is disabled by default, in order to configure it, go in *Configuration* → *Compliance* → *Fingerbank Profiling* → *Device Change Detection*.

You should then check **Enabled** to activate this feature. You will then have the choice between triggering the security event on any device class change or on a specific set of changes.

17.7.1. Triggering on any device class change

NOTE | You should perform non-enforcing actions in the security event when initially deploying the feature to see if some corner cases may require whitelisting some device class transitions

The easiest method for performing this detection is to trigger on any device class change which will trigger the security event whenever the device is detected transitioning from any device class to another. Since some of these transitions may be normal in your environment, you can add whitelisting of transitions via the "Device class change whitelist" parameter which allows you to list valid transitions (ex: "Windows OS" to "Mac OS X or macOS").

17.7.2. Manual triggers

Instead of detecting all transitions, you can perform detection and security event triggering on specific device class transitions. In order to do so, declare all the transitions that should trigger the security event in the "Manual device class change triggers".

18. Network Devices Anomaly Detection

Starting with version 10, PacketFence integrates network devices anomaly detection capabilities. This means that PacketFence can detect abnormal network activities from devices - that is, if they are talking to a compromised host, if they are deviating from their pristine network profile and more. These capabilities come from the integration of the Fingerbank technology. That is, the Fingerbank Cloud API is responsible for producing pristine network device profiles while the Fingerbank Collector, included in PacketFence, does consume the pristine profiles and does anomaly detection based on its templating engine.

18.1. Creating Network Behavior Policies

A network behavior policy is a template, used by the Fingerbank Collector, to determine if the devices matching the criterias defined in the template ultimately deviate from a normal network usage pattern. You can create new templates from *Configuration* → *Compliance* → *Network Anomaly Detection*.

Network behavior policies can be consumed from PacketFence's Security Events module.

18.2. Integration with Security Events

After creating a network behavior policy, you can use it from the Security Events module of PacketFence. From *Configuration* → *Compliance* → *Security Events*, click on **New Security Event**.

You can use your policy by first adding a new trigger. The network behavior policy can be selected after defining an internal event on the following attributes:

- **fingerbank_blacklisted_ips_threshold_too_high** - Fingerbank Collector detected traffic to blacklisted IPs
- **fingerbank_blacklisted_ports** - Fingerbank Collector detected traffic to blacklisted ports
- **fingerbank_diff_score_too_low** - Fingerbank Collector detected a network behavior that doesn't match the known profile

Once done, the appropriate policy can be selected. If you want your entire network policy to be checked in the Security Events module, you must create three triggers - one with each of the attribute listed above together with your appropriate policy selected. You can look at the default security events Fingerbank profile anomaly (1300007) and Fingerbank detected blacklisted communication (1300008) for some examples on how to create customized security events to fulfill your requirements.

19. Tenants

PacketFence supports multi-tenancy but by default, it is in single-tenant mode.

19.1. General concepts

19.1.1. Built-in tenants

Tenant **1** is the default tenant. A default PacketFence installation will use this tenant transparently.

Tenant **0** is the global tenant. In a multi-tenant configuration, you can modify configuration only in that tenant.

All items configuration are global and shared between tenants. If you want to manage a multi-tenant configuration, you will need to deploy a specific configuration that will rely on tenant IDs.

19.1.2. Domain name and portal domain name of a tenant

A tenant can be created with two **optional** parameters:

- a domain name: to create users in database using following convention: `username@DOMAIN_NAME_OF_TENANT`
- a portal domain name: to assign nodes to a specific tenant when they reach PacketFence on this captive portal address

These two parameters are used to assign tenants to PacketFence objects.

When possible, we highly recommend you to create tenant with these two parameters.

19.1.3. Network devices (switches)

A network device is attached to one tenant at creation.

When PacketFence receives a RADIUS request from a network device, it will instantiate a network device in this tenant and create node in this tenant. Then you can use **Tenant ID** field in connection profiles filters to apply specific configuration for each tenant.

In some WebAuth scenarios, PacketFence doesn't receive RADIUS requests from network devices (only HTTP/s requests). Consequently, it can't assign tenant ID to nodes based on network devices' tenant ID. In that case, you need to define a portal domain name in your tenant and filter on **FQDN** field in connection profiles.

19.2. Getting started

19.2.1. Creating your tenant

Following command will create a new tenant in database, mapped to `example.com` domain and with a specific captive portal address `portal.example.com`:

```
/usr/local/pf/bin/pfcmd tenant add example example.com portal.example.com
```

In order to apply changes, you need to restart PacketFence services (on each server in a cluster configuration):

```
systemctl restart packetfence-config  
/usr/local/pf/bin/pfcmd service pf restart
```

After that, if you try to connect on web admin using the default `admin` account, you will not be able to access any configuration items. Now you need to do configuration actions using the `system` account which has a specific password defined in `/usr/local/pf/conf/unified_api_system_pass` in cleartext. Or create an admin account in global tenant to manage configuration.

20. Intrusion Detection System Integration

20.1. Regex Syslog Parser

You are now able to create syslog parser using regex. This will allow you complex filters and rules to work on data receive via syslog.

Configuring a Regex Syslog Parser

- Enabled - You can enable/disable the parser from running
- Alert Pipe - A previously created alert pipe (FIFO)
- Rules - The list of rules that defines how to match log file entries and what action(s) to take when matching

Regex Syslog Parser Rule

- Name - The name of the rule
- Regex - The regex to match against a log entry. The regex may have [named captures](#) which can be used for parameter replacement start a '\$'.
- Actions - A list of actions to take when the regex matches
- IP to MAC - Perform automatic translation of IPs to MACs and the other way around
- Last if matches - Stop processing the other rules if this rule matched

Defining Actions

An action have two parts

- method - The name of the action you want to take
- parameter list - The list of parameters you want to provide to the method. Each parameter is separated by a comma. The parameters that are to be replaced by a named capture.

Example Action

Regex -

```
mac\s*:\s*(?P<mac>[a-zA-Z0-9]{2}(:[a-zA-Z0-9]{2}){5}),
notes\s*:\s*(P?<notes>.*)
```

Action -

```
modify_node: mac, $mac, notes, $notes
```


20.2. Suricata IDS

PacketFence already contains a syslog parser for Suricata. This is an example to raise a security event from a syslog alert on the Suricata SID.

The first step is to create the syslog regex parser and then create the security event.

20.2.1. Syslog regex parser configuration

To create the syslog regex parser you will need to go to *Configuration* → *Integration* → *Syslog Parsers* → *Add a Syslog Parser* → *regex*

Here is the configuration of the syslog regex parser:

```
Detector *: Suricata
Enabled: checked
Alert pipe: /usr/local/pf/var/suricata (To create the fifo file, do: mkfifo
/usr/local/pf/var/suricata)
```

Rules:

Rule - New:

```
Name *: ET P2P Kaaza Media desktop p2pnetworking.exe
Regex *: (?P<date>\d{2}\/\d{2}\/\d{4}-\d{2}:\d{2}:\d{2}.*) \[\*\*\]
\[\d+:(?P<sid>\d+):\d+\] (?P<message>.*) \[\*\*\].*
(?P<srcip>\d{1,3}(\.\d{1,3}){3}):(?P<srcport>\d+) ->
(?P<ip>\d{1,3}(\.\d{1,3}){3}):(?P<port>\d+)
Action: trigger_security_event mac, $mac, tid, $sid, type, detect
Last if match: unchecked
IP to MAC: checked
```

Save the regex rule.

You can directly test your rule. In the previous example the parser expect a syslog string like this:

```
02/26/2017-14:29:00.524309 [\*\*] [1:2000340:10] ET P2P Kaaza Media desktop
p2pnetworking.exe Activity [\*\*] [Classification: Potential Corporate Privacy
Violation] [Priority: 1] {UDP} 173.194.7.75:443 -> 1.2.3.4:46742
```

In order to have a correct match in the rule, you will need to have a valid iplog entry in the database. Put the string in the test box and then click on the **RUN TEST** button, you should get:

```
Click to see actions for - 02/26/2017-14:29:00.524309 [**] [1:2000340:10] ET
P2P Kaaza Media desktop p2pnetworking.exe Activity [**] [Classification:
Potential Corporate Privacy Violation] [Priority: 1] {UDP} 173.194.7.75:443 ->
1.2.3.4:46742
```

- ET P2P Kaaza Media desktop p2pnetworking.exe : trigger_security_event('mac', '00:11:22:33:44:55', 'tid', '2000340', 'type', 'detect')

We can see that PacketFence will execute the security event on the MAC address 00:11:22:33:44:55.

20.2.2. Security Event Creation

Now you will need to create the security event with the trigger id '2000340' in order to isolate the device. In order to do so, go to *Configuration* → *Compliance* → *Security Events* → *New Security Event*

Definition:

```
Enabled: ON
Identifier: 1500001
Description: ET P2P Kaaza Media
Action: Reevaluate Access Action; Log message
Priority: 1
```

Triggers:

- Click on the (+) button
- Look for 'detect' in the dropdown list
- Add the trigger ID: 2000340 and click the ADD button
- Click on the < button next to 'Select Some Options'

Remediation:

```
Auto Enable: checked
Max Enables: 2
Grace: 5 minutes
Template: p2p.html
```

Click on the SAVE button.

Now you will need to restart the pfqueue and the pfdetect services.

```
/usr/local/pf/bin/pfcmd service pfqueue restart
```

```
/usr/local/pf/bin/pfcmd service pfdetect restart
```

Make sure that you have your pipe file otherwise the process won't start.

20.3. Security Onion

20.3.1. Installation and Configuration

Security Onion is a Ubuntu-based security suite. The latest installation instructions are available directly from the Security Onion website, <https://github.com/Security-Onion-Solutions/security-onion/wiki/Installation>

Since a security suite consists of multiple pieces of software tied together, you may be prompted for different options during the installation process. A detailed "Production Deployment" guide can also be found directly from the Security Onion website: <https://github.com/Security-Onion-Solutions/security-onion/wiki/ProductionDeployment>

20.3.2. PacketFence Integration

Once Security Onion is installed and minimally configured, integration with PacketFence is required to be able to raise security events based on sensor(s) alerts. syslog is used to forward sensor(s) alerts from Security Onion to the PacketFence detection mechanisms.

The simplest way is as follow (based on <https://github.com/Security-Onion-Solutions/security-onion/wiki/ThirdPartyIntegration>);

On the Security Onion server:

NOTE | Must be done on the master server running 'sguild'.

Configure `/etc/syslog-ng/syslog-ng.conf` by adding the following to enable sending sguild log entries to PacketFence:

```

### PacketFence / IDS integration
# This line specifies where the sguild.log file is located
# -> Make sure to configure the right path along with the right filename (on a
Security Onion setup, that should be pretty much standard)
source s_sguild { file("/var/log/nsm/securityonion/sguild.log"
program_override("securityonion_ids")); };
# This line filters on the string "Archived Alert"
filter f_sguild { match("Archived Alert"); };
# This line tells syslog-ng to send the data read to the PacketFence management
IP address using UDP 514
# -> Make sure to configure the right PacketFence management interface IP
address
destination d_packetfence_alerts { udp("PACKETFENCE_MGMT_IP" port(514)); };
# This line indicates syslog-ng to use the s_sguild source, apply the f_sguild
filter and send it to the d_packetfence_alerts destination
log { source(s_sguild); filter(f_sguild); destination(d_packetfence_alerts); };

```

NOTE | Ensure you change PACKETFENCE_MGMT_IP to the management IP address of your PacketFence server

Sending sguild alert output to syslog requires DEBUG to be changed from 1 to 2 under [/etc/sguild/sguild.conf](#)

```
set DEBUG 2
```

A restart of the sguild daemon is then required

```
sudo nsm_server_ps-restart
```

A restart of the syslog-ng daemon is then required

```
service syslog-ng restart
```

On the PacketFence server:

Modify rsyslog configuration to allow incoming UDP packets by uncommenting the following two lines in [/etc/rsyslog.conf](#):

```
$ModLoad imudp
$UDPServerRun 514
```

Configure [/etc/rsyslog.d/securityonion_ids.conf](#) so it contains the following which will redirect Security Onion sguild log entries and stop further processing of current matched message:

```
if $programname == 'securityonion_ids' then /usr/local/pf/var/securityonion_ids
& ~
```

Make sure the receiving alert pipe (FIFO) exists

```
mkfifo /usr/local/pf/var/securityonion_ids
```

Restart the rsyslog daemon

```
service rsyslog restart
```

At this point, Security Onion should be able to send detected alerts log entries to PacketFence.

A configuration of a new 'syslog parser' as well as some security events are the only remaining steps to make full usage of the Security Onion IDS integration.

Configuration of a new 'syslog parser' should use the followings:

```
Type: security_onion
Alert pipe: the previously created alert pipe (FIFO) which is, in this case,
/usr/local/pf/var/securityonion_ids
```

Configuration of a new security event can use the following trigger types:

```
Type: detect
Triggers ID: The IDS triggered rule ID
```

```
Type: suricata_event
Trigger ID: The rule class of the triggered IDS alert
```

20.4. Security Onion 2.3.10

This documentation is based on Security Onion v2.3. You can review its documentation at: <https://docs.securityonion.net/en/2.3>

All commands are done through the SSH CLI.

20.4.1. Suricata configuration on SO

First we need to modify the Suricata configuration to output the alerts into a fast.log file.

```
sudo vim /opt/so/saltstack/default/salt/suricata/defaults.yaml
```

Locate the outputs section and modify the fast options as follow:

```
outputs:
  - fast:
    enabled: "no"
    filename: /nsm/fast.log
    append: "yes"
  - eve-log:
    enabled: "yes"
    filetype: regular
    filename: /nsm/eve-%Y-%m-%d-%H:%M.json
    rotate-interval: hour
    #prefix: "@cee: "
    #identity: "suricata"
    #facility: local5
    #level: Info
    #redis:
    # server: 127.0.0.1
```

Reload the configuration on all minions with (it will take few minutes to apply):

```
sudo salt '*' state.highstate
```

You can verify the configuration done under:

```
sudo vim /opt/so/conf/suricata/suricata.yaml
```

20.4.2. Rsyslog configuration on SO

Now we need to send the alerts from the /nsm/fast.log to PacketFence.

```
sudo vim /etc/rsyslog.d/SO.conf
```

Replace the PACKETFENCE_MGMT_IP with your PacketFence management IP interface.

```
$ModLoad imfile
$InputFileName /nsm/suricata/fast.log
$InputFileTag suricata
$InputFileStateFile stat-suricata
$InputFileSeverity error
$InputFileFacility local3
$InputRunFileMonitor
local3.* @PACKETFENCE_MGMT_IP:514
```

Restart Rsyslog:

```
sudo systemctl restart rsyslog
```

20.4.3. Configure PacketFence to process the syslog traffic

On the PacketFence server:

Modify rsyslog configuration to allow incoming UDP packets by uncommenting the following two lines in `/etc/rsyslog.conf`:

```
$ModLoad imudp
$UDPServerRun 514
```

Configure `/etc/rsyslog.d/securityonion_ids.conf` so it contains the following which will redirect Security Onion sguild log entries and stop further processing of current matched message:

```
if $programname == 'suricata' then /usr/local/pf/var/securityonion_ids
& ~
```

Make sure the receiving alert pipe (FIFO) exists

```
mkfifo /usr/local/pf/var/securityonion_ids
```

Restart the rsyslog daemon

```
service rsyslog restart
```

At this point, Security Onion should be able to send detected alerts log entries to PacketFence.

A configuration of a new 'syslog parser' as well as some security events are the only remaining steps to make full usage of the Security Onion IDS integration.

Configuration of a new 'syslog parser' should use the followings:

```
Type: suricata
Alert pipe: the previously created alert pipe (FIFO) which is, in this case,
/usr/local/pf/var/securityonion_ids
```

Configuration of a new security event can use the following trigger types:

```
Type: detect
Triggers ID: The IDS triggered rule ID
```

```
Type: suricata_event
Trigger ID: The rule class of the triggered IDS alert
```

20.5. ERSPAN

ERSPAN permits to mirror a local port traffic (low bandwidth) to a remote IP, E.G: your Security Onion already deployed box. ERSPAN encapsulates port traffic into ERSPAN then GRE and send that traffic to one/multiple destination(s). ERSPAN is a Cisco technology which is available only on some platforms, including: Catalyst 6500, 7600, Nexus, and ASR 1000.

One way of accessing encapsulated traffic at the destination host is through a software called RCD CAP, which is a daemon that creates a virtual interface if not existing, on which both GRE and ERSPAN headers are decapsulated prior to the traffic being injected to the previous interface. Security Onion can then feed on that interface like it would on any other, and if the RCD CAP daemon dies, continue to listen to that interface even though decapsulated traffic won't be available anymore.

Assumptions for the example: The switch is at IP 172.16.0.1, the monitored switch port is GigabitEthernet0/10 and the Security Onion monitoring destination IP is 10.10.10.10 on eth2, eth2 ideally being a dedicated interface.

On Security Onion:

- Enable Inverse repository for Security Onion:

```
sudo bash -c 'cat << EOL >/etc/apt/sources.list.d/securityonion-inverse.list
deb http://inverse.ca/downloads/PacketFence/securityonion trusty trusty
EOL'
```

```
gpg --keyserver keyserver.ubuntu.com --recv 19CDA6A9810273C4
gpg --export --armor 19CDA6A9810273C4 | sudo apt-key add -
```

- Install RCD CAP

```
sudo apt-get update
sudo apt-get install rcdcap
```

- Modify network file (/etc/network/interfaces) so that eth2 has an IP and a proper MTU. Decapsulated traffic will be injected on mon1. Make sure that the configuration is similar to the following:


```

auto eth2
iface eth2 inet static
    address 10.10.10.10
    netmask 255.255.255.240
    up ip link set $IFACE arp on up
    up ip link set dev $IFACE mtu 1900
    post-up ethtool -G $IFACE rx 4096; for i in rx tx sg tso ufo gso gro lro; do
ethtool -K $IFACE $i off; done
    post-up echo 1 > /proc/sys/net/ipv6/conf/$IFACE/disable_ipv6

auto mon1
iface mon1 inet manual
    pre-up rcdcap -i eth1 --erspan --tap-persist --tap-device $IFACE --expression
"host 172.16.0.1" -d
    up ip link set $IFACE promisc on arp off up
    down ip link set $IFACE promisc off down
    post-up ethtool -G $IFACE rx ; for i in rx tx sg tso ufo gso gro lro; do
ethtool -K $IFACE $i off; done
    post-up echo 1 > /proc/sys/net/ipv6/conf/$IFACE/disable_ipv6

```

- Rerun Security Onion wizard and make sure to skip network configuration step. Make sure that mon1 is selected for monitoring purposes, note that eth2 doesn't need to.

```
sudo ssetup
```

On the Switch:

```

monitor session 10 type erspan-source
description ERSPAN to 10.10.10.10
source interface GigabitEthernet0/10
destination
erspan-id 10
ip address 10.10.10.10
origin ip address 172.16.0.1
no shutdown ! Default is shutdown

```

20.6. StreamScan Comprromise Detection System (CDS)

This is an example to raise a security event from a syslog alert on a StreamScan alert ID.

The first step is to create the syslog regex parser and then create the security event.

20.6.1. Syslog regex parser configuration

To create the syslog regex parser you will need to go to *Configuration* → *Integration* → *Syslog*

Parsers → Add a Syslog Parser → regex

Here is the configuration of the syslog regex parser:

```
Detector *: StreamScan
Enabled: checked
Alert pipe: /usr/local/pf/var/cds
```

Rules:

Rule - New:

```
Name *: ET TROJAN
Regex *: CDS\[(<?<cds_id>\d+)\].*?type=(?<type>[^\ ]*).?*threat="(?"<threat>.*)"
direction=(?<direction>[^\ ]+) sourceip=(?<sourceip>\d+(\.\d+){3})
sourceport=(?<sourceport>\d+) destip=(?<ip>\d+(\.\d+){3})
destport=(?<destport>\d+) app=(?<app>[^\ ]*) timestamp=(?<timestamp>[^\ ]*)
sid=(?<sid>\d+)
Action: trigger_security_event mac, $mac, tid, $sid, type, detect
Last if match: unchecked
IP to MAC: checked
```

Save the regex rule.

You can directly test your rule. In the previous example the parser expect a syslog string like this:

```
Apr 24 16:50:41 ubuntu CDS[13423]: type=alert threat="ET TROJAN Likely Zbot
Generic Post to gate.php no accept headers" direction=outgoing
sourceip=192.168.254.194 sourceport=53252 destip=5.175.143.42 destport=80
app=HTTP timestamp=2017-04-24_16-50-41.832096 sid=2022985
```

In order to have a correct match in the rule, you will need to have a valid iplog entry in the database. Put the string in the test box and then click on the 'RUN TEST' button, you should get:

Results

```
Click to see actions for - Apr 24 16:50:41 ubuntu CDS[13423]: type=alert
threat="ET TROJAN Likely Zbot Generic Post to gate.php no accept headers"
direction=outgoing sourceip=192.168.254.194 sourceport=53252
destip=5.175.143.42 destport=80 app=HTTP timestamp=2017-04-24_16-50-41.832096
sid=2022985
- security event: trigger_security_event('mac', '00:11:22:33:44:55', 'tid',
'2022985', 'type', 'detect')
```

We can see that PacketFence will execute the security event on the MAC address 00:11:22:33:44:55.

20.6.2. Security Event Creation

Now you will need to create the security event with the trigger id '2000340' in order to isolate the device. In order to do so, go to *Configuration* → *Compliance* → *Security Events* → *New Security Event*.

Definition:

```
Enabled: ON
Identifier: 2022985
Description: ET Trojan
Action: Reevaluate Access Action; Log message
Priority: 1
```

Triggers:

- Click on the + button
- Look for 'detect' in the dropdown list
- Add the trigger ID: 2022985 and click the ADD button
- Click on the '<' button next to 'Select Some Options'

Remediation:

```
Auto Enable: checked
Max Enables: 2
Grace: 5 minutes
Template: generic.html
```

Click on the SAVE button.

Now you will need to restart the pfqueue and the pfdetect services.

```
/usr/local/pf/bin/pfcmd service pfqueue restart
```

```
/usr/local/pf/bin/pfcmd service pfdetect restart
```

Make sure that you have your pipe file otherwise the process won't start.

20.6.3. Rsyslog Configuration

You will need to create a rsyslog configuration to forward all the syslog messages sent by StreamScan to the pipe file `/usr/local/pf/var/cds`

```
mkfifo /usr/local/pf/var/cds
```

First you need to enable the syslog under the rsyslog configuration file located at `/etc/rsyslog.conf`, uncomment the two parameters:

```
# Provides UDP syslog reception
$ModLoad imudp
$UDPServerRun 514
```

Then create the forwarding configuration:

```
vim /etc/rsyslog.d/cds.conf
```

Put the follow configuration in it where the IP 1.2.3.4 is your syslog source server:

```
if ($fromhost-ip=='172.20.20.181') then /usr/local/pf/var/cds
&~
```

Restart rsyslog to apply the configuration

```
service rsyslog restart
```

21. Firewall SSO Integration

PacketFence is able to update some firewall based on device information, like the IP address, the username connected on it. Look below for integration guides to see how you can configure your firewall with PacketFence. By default PacketFence uses the DHCP traffic to trigger an update on the firewall but it's also possible to do it with the RADIUS accounting traffic.

In order to manage the way you want to update the firewall, go in *Configuration* → *System Configuration* → *Main Configuration* → *Advanced*, then there are two choices:

- Trigger Single-Sign-On on accounting.
- Trigger Single-Sign-On on DHCP

You can use both methods at the same time but this will result in duplicate SSO requests if you receive the DHCP and accounting of the same device which can cause unexpected load on your firewall.

21.1. Barracuda

21.1.1. Configuration of the Barracuda in PacketFence

Go to *Configuration* → *Integration* → *Firewall SSO* → *Add Firewall* → *Barracuda*.

- **Hostname or IP Address:** IP of your Barracuda
- **Firewall type:** Barracuda (Barracuda = SSH requests)
- **Password:** secret
- **Port:** 22
- **Roles:** add the roles that you want to do SSO

The screenshot displays the PacketFence configuration interface for setting up a new Firewall SSO. The interface is divided into a sidebar and a main configuration area.

Sidebar:

- Filter
- Policies and Access Control
- Compliance
- Integration
 - Firewall SSO
 - Cisco Mobility Services Engine
 - Web Services
 - Syslog Parsers
 - Syslog Forwarding
 - WRIX
- Advanced Access Configuration
- Network Configuration
- System Configuration

Main Configuration Area: New Firewall SSO (BarracudaNG)

- Hostname or IP Address:** 192.168.100.3
- Username:** root
- Secret or Key:** [Masked]
- Port of the service:** 22
- Roles:** Staff
- Networks on which to do SSO:** [Empty field]
- Cache updates:** [Toggle Off]
- Cache timeout:** [Empty field]
- Username format:** \$pf_username
- Default realm:** [Empty field]

Buttons: Create, Reset

21.1.2. Step 2: Verification

For our example, when the user registers on the portal it will be registered and the role staff will be assigned. The PacketFence server will send a request to the Barracuda database.

If you want to see if it's working, open an SSH access to your Barracuda and run this command

following commands:

```
acpfctl auth show
```

You will get that:

```
[root@baracudafw:~]# acpfctl auth show
1 entries
172.20.20.152/0
origin=PacketFence
service=PacketFence
user=Jdoe
```

21.2. Checkpoint

21.2.1. Enabling Identity Awareness on the Security Gateway

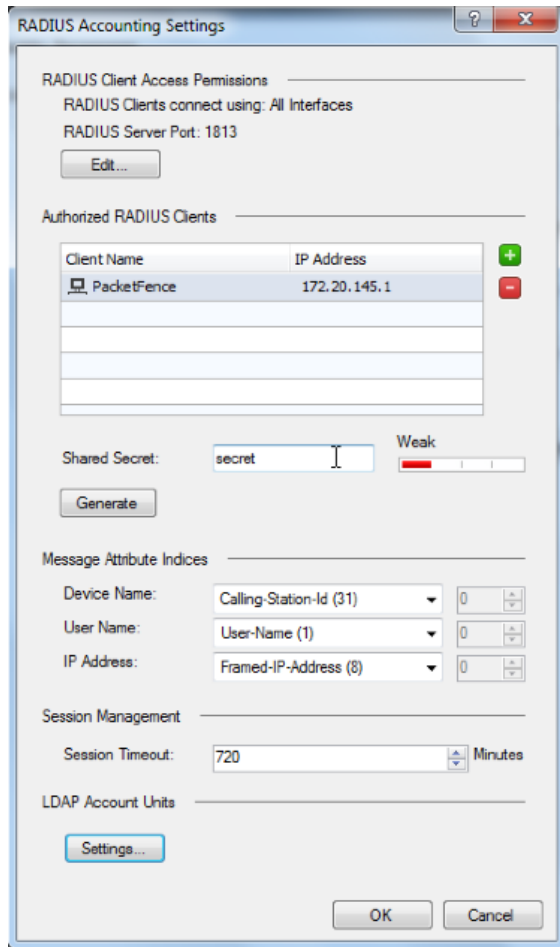
To enable Identity Awareness:

1. Log in to 'SmartDashboard'.
2. From the 'Network Objects tree', expand the 'Check Point branch'.
3. Double-click the 'Security Gateway' on which to enable 'Identity Awareness'.
4. In the 'Software Blades' section, select 'Identity Awareness' on the 'Network Security tab'. The 'Identity Awareness Configuration' wizard opens.
5. Select 'one or more options'. These options set the methods for acquiring identities of managed and unmanaged assets.
6. Select 'AD Query - Lets the Security Gateway seamlessly identify Active Directory users and computers' and click Next. The 'Integration With Active Directory' window opens.
7. Select the Active Directory to configure from the list that shows configured LDAP account units or create a new domain. If you have not set up Active Directory, you need to enter a domain name, username, password and domain controller credentials.
8. Enter the Active Directory credentials and click Connect to verify the credentials. (Important - For AD Query you must enter domain) administrator credentials.
9. Click Finish.

21.2.2. Enabling RADIUS Accounting on a Security Gateway

To enable RADIUS Accounting for a Security Gateway: 1. In the 'SmartDashboard Network Objects tree', open the Security Gateway. 2. On the 'General Properties' page, make sure that the Identity Awareness Blade is enabled. 3. On the 'Identity Awareness' page, select RADIUS Accounting.

21.2.3. Configuring RADIUS Accounting



1. In the 'Check Point Gateway' window > 'Identity Awareness' panel, click 'Settings' (to the right of the RADIUS Accounting option).
2. In the 'RADIUS Accounting Settings' window, configure the 'Message Attribute Indices' like this:
 - **Device Name:** Calling-Station-Id (31) (MAC Address of the device)
 - **User Name:** User-Name (1) (Username put on the PacketFence Portal)
 - **Device Name:** Framed-IP-Address (8) (IP Address of the device in the production network)

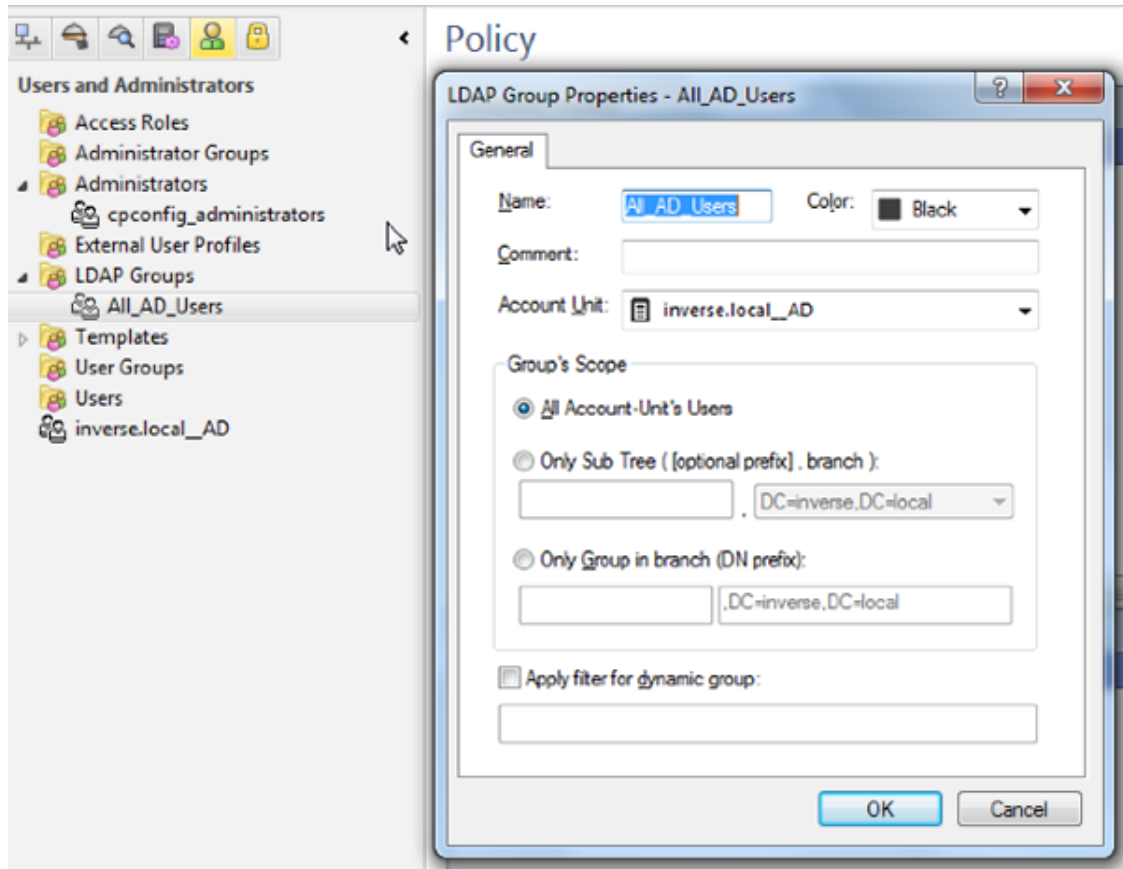
21.2.4. RADIUS Client Access Permissions

Gateway interfaces must be authorized to accept connections from PacketFence RADIUS Accounting.

To select gateway interfaces: 1. In the 'RADIUS Client Access Permissions' section, click Edit. 2. Select 'All Interfaces - All Security Gateway interfaces can accept connections from RADIUS Accounting clients'. 3. Leave the default port to 1813. 4. Click OK on both windows to submit the configuration. 5. Select 'Policy' > 'Install' from the SmartDashboard menu.

21.2.5. LDAP Groups

Make sure that you have the correct LDAP Objects created on the Checkpoint.



21.2.6. SSO Configuration in PacketFence

Go to '*Configuration' → 'Firewall SSO' → 'Add Firewall' → 'Checkpoint' *

- **Hostname or IP Address:** IP of your Checkpoint firewall
- **Secret or Key:** secret (radius shared secret)
- **Port:** 1813
- **Roles:** add the roles that you want to do SSO with

The screenshot displays the 'New Firewall SSO' configuration window in the PacketFence management console. The interface includes a top navigation bar with 'Configuration' selected, and a left sidebar with a search filter and a tree view of configuration categories. The main configuration area contains the following fields and options:

- Hostname or IP Address:** 192.168.100.2
- Secret or Key:** A masked field with a toggle to show/hide the characters.
- Port of the service:** 1813, with a note: "If you use an alternative port, please specify."
- Roles:** A dropdown menu currently showing 'Staff', with a note: "Nodes with the selected roles will be affected."
- Networks on which to do SSO:** A text input field with a note: "Comma delimited list of networks on which the SSO applies. Format : 192.168.0.0/24"
- Cache updates:** A toggle switch that is currently turned off. A note explains: "Enable this to debounce updates to the Firewall. By default, PacketFence will send a SSO on every DHCP request for every device. Enabling this enables 'sleep' periods during which the update is not sent if the informations stay the same."
- Cache timeout:** A text input field with a note: "Adjust the 'Cache timeout' to half the expiration delay in your firewall. Your DHCP renewal interval should match this value."
- Username format:** \$pf_username, with a detailed note: "Defines how to format the username that is sent to your firewall. \$username represents the username and \$realm represents the realm of your user if applicable. \$pf_username represents the unstripped username as it is stored in the PacketFence database. If left empty, it will use the username as stored in PacketFence (value of \$pf_username)."
- Default realm:** A text input field with a note: "The default realm to be used while formatting the username when no realm can be extracted from the username."

At the bottom of the configuration window, there are two buttons: 'Create' (in blue) and 'Reset' (in white).

21.2.7. Verification

You can check the correct log in with the SmartView Tracker under 'Network & Endpoint Queries' → 'Predefined' → 'Identity Awareness Blade' → 'Login Activity'

21.3. Cisco ISE-PIC

21.3.1. Preliminary steps

First, attach ISE-PIC to Active Directory and set it up as an Identity Provider as described here: https://www.cisco.com/c/en/us/td/docs/security/ise/2-6/pic_admin_guide/PIC_admin26/PIC_admin26_chapter_010.html

21.3.2. Syslog template

Add a new Template and call it **PacketFence**. Make it match the following:

Syslog Template X

Name *

Mapping Operations

New Mapping *

Removed Mapping

User Data

IP Address *

User Name

Domain

MAC address

Test Template

Paste one line of syslog

Data Identified

- User name
- IP Address
- Domain
- MAC Address

- The new mapping should be set to: **assigned to session**
- The regular expression for the IP address is: **Address <([^\s]+)>|address ([^\s]+)**
- The regular expression for the username is: **User <([^\s]+)>**

21.3.3. Syslog provider

To add PacketFence as an identity provider, hover over "Providers" and click "Syslog Providers.", then click "Add".

Then add each of your PacketFence servers as Syslog providers using the syslog template you created above. In the case of a cluster, add each member management IP and the management virtual IP.

NOTE In your DNS servers, make sure the FQDN and reverse lookup entries match your PacketFence server FQDN.

Syslog Providers

Name *	<input type="text" value="MyPFInstance"/>
Description	<input type="text"/>
Status *	<input type="text" value="Enabled"/>
Host FQDN *	<input type="text" value="pf1.mydomain.com"/>
Connection Type *	<input type="text" value="UDP - Port 40514"/>
Template *	<input type="text" value="PacketFence"/> <input type="button" value="Edit"/> <input type="button" value="New"/>
Default Domain	<input type="text" value="mydomain.com"/>

Make sure your syslog header configuration matches this:

Syslog Custom Header

If some or all of the syslogs are not being accepted, it may be because they have an uncommon header format. Define a custom header here.

Paste sample syslog *

Separator *

Position of hostname in header *

Hostname Following analysis of sample syslog; the hostname will appear here. If correct, then save this custom header.

21.3.4. PacketFence configuration

Add a Cisco ISE-PIC firewall SSO entry in "Configuration→Integration→Firewall SSO"

New Firewall SSO Cisco ISE-PIC

Hostname or IP Address: 192.168.1.100

Port of the service: 40514
If you use an alternative port, please specify.

Roles: default
Nodes with the selected roles will be affected.

Networks on which to do SSO
Comma delimited list of networks on which the SSO applies.
Format : 192.168.0.0/24

Cache updates:
Enable this to debounce updates to the Firewall.
By default, PacketFence will send a SSO on every DHCP request for every device. Enabling this enables "sleep" periods during which the update is not sent if the informations stay the same.

Cache timeout:
Adjust the "Cache timeout" to half the expiration delay in your firewall.
Your DHCP renewal interval should match this value.

Username format: \$pf_username
Defines how to format the username that is sent to your firewall. \$username represents the username and \$realm represents the realm of your user if applicable. \$pf_username represents the unstripped username as it is stored in the PacketFence database. If left empty, it will use the username as stored in PacketFence (value of \$pf_username).

Default realm:
The default realm to be used while formatting the username when no realm can be extracted from the username.

- **Hostname or IP Address:** IP of your Cisco ISE-PIC instance
- **Port:** 40514
- **Roles:** add the roles that you want to do SSO with

You should then see User Sessions populating under "Live Logs" in ISE-PIC. The source should say "syslog"

21.4. FortiGate

21.4.1. Configuration of the RSSO Agent

Go to your FortiGate administration webpage in **User & Device** → **User** → **User Groups** → **Create New**.

- **Name:** RSSO_group
- **Type:** RADIUS Single Sign-On (RSSO)
- **RADIUS Attribute Value:** RSSO_Student (use the rolename of PacketFence, it's case sensitive)

FortiWiFi 60C FortINET

Edit User Group

Name: RSSO_Student

Type: Firewall Fortinet Single Sign-On (FSSO) Guest RADIUS Single Sign-On (RSSO)

RADIUS Attribute Value: RSSOStudent

System
Policy
Firewall Objects
Security Profiles
VPN
User & Device
 User
 User Definition
 User Groups
 Guest Management
 Device

You can also see that in the webpage at [User & Device → Monitor → Firewall](#)

21.4.2. Configure the endpoint attribute

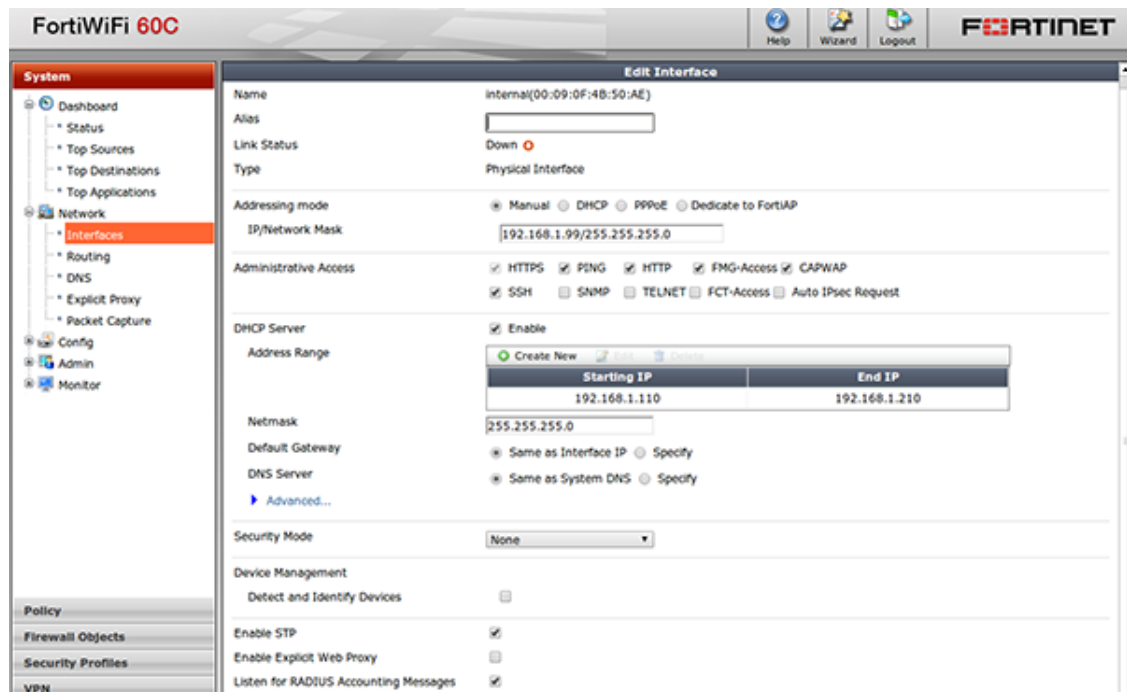
The default endpoint attribute is the Calling-Station-Id so the MAC address shows up under User Name, we can change that in CLI:

```
config user radius
edit RSSO_agent
set rso-endpoint-attribute User-Name
end
```

21.4.3. Activate the Accounting Listening

Go to [System → Network → Interfaces](#).

Select the interface that will communicate with PacketFence and check 'Listen for RADIUS Accounting Messages' then confirm.



21.4.4. SSO Configuration in PacketFence

Go to [Configuration → Integration → Firewall SSO → Add Firewall → FortiGate](#).

- **Hostname or IP Address:** IP of your firewall
- **Secret or Key:** secret (radius shared secret)
- **Port:** 1813
- **Roles:** add the roles that you want to do SSO

The screenshot shows the PacketFence configuration interface. The top navigation bar includes 'Status', 'Reports', 'Auditing', 'Nodes', 'Users', and 'Configuration'. The user is logged in as 'admin'. The left sidebar shows a navigation menu with categories like 'Policies and Access Control', 'Compliance', 'Integration', 'Advanced Access Configuration', 'Network Configuration', and 'System Configuration'. The main content area is titled 'New Firewall SSO' and contains the following configuration fields:

- Hostname or IP Address:** 192.168.100.2
- Secret or Key:** [Redacted]
- Port of the service:** 1813
- Roles:** Staff
- Networks on which to do SSO:** [Empty field]
- Cache updates:** [Enabled]
- Cache timeout:** [Empty field]
- Username format:** \$pf_username
- Default realm:** [Empty field]

At the bottom of the configuration form, there are 'Create' and 'Reset' buttons.

21.4.5. Verification

If you want to see if it's working, you can log into the firewall over SSH and run these following commands:

```
di debug enable
di debug application radiusd -1
```

21.5. iBoss

21.6. JSON-RPC

21.6.1. JSON-RPC interface

The JSONRPC module shipped with PacketFence is meant as a generic firewall SSO module to be used with Linux or BSD firewalls that do not by default ship with a vendor-specific interface to do SSO with.

A compatible server must implement the methods **Start** and **Stop**, both with the identical set of parameters provided below.

- **Protocol:** JSON-RPC 2.0 over HTTPS
- **Authentication:** HTTP Basic authentication
- **Methods:** **Start** and **Stop**
- **Parameters:**
 - **user** (*string*): Username that registered the device
 - **mac** (*string*): MAC address of the device
 - **ip** (*string*): IP address of the device
 - **role** (*string*): PacketFence role assigned to the device
 - **timeout** (*int*): Duration until the registration expires in seconds
- **Response:** Success must be indicated by **"result": ["OK"]**. Every string other than **OK** is taken as an error message.

A simple JSON-RPC server written in Python that is compatible with this specification and creates ipsets based on the SSO information provided by PacketFence can be found at <https://github.com/tribut/ipset-rpcd>.

21.6.2. SSO Configuration in PacketFence

Go to 'Configuration' → 'Integration' → 'Firewall SSO' → 'Add Firewall' → 'JSONRPC'.

- **Hostname or IP Address:** IP of your JSON-RPC server
- **Username and Password:** HTTP Basic credentials
- **Port of the service:** 9090
- **Roles:** Add the roles that you want to do SSO with

The screenshot displays the 'New Firewall SSO' configuration window in PacketFence. The interface includes a top navigation bar with 'Configuration' selected, and a left sidebar with categories like 'Policies and Access Control', 'Compliance', 'Integration', 'Advanced Access Configuration', 'Network Configuration', and 'System Configuration'. The main configuration area contains the following fields and options:

- Hostname or IP Address:** 192.168.100.1
- Username:** Jsonrpc-updater
- Password:** [Redacted]
- Port of the service:** 9090
- Roles:** gaming, guest
- Networks on which to do SSO:** [Empty field]
- Cache updates:** [Toggle off]
- Cache timeout:** [Empty field]
- Username format:** \$pf_username
- Default realm:** [Empty field]

At the bottom of the configuration window are 'Create' and 'Reset' buttons.

21.7. Juniper SRX

21.7.1. Configuration of the Juniper SRX in PacketFence

Go to **Configuration** → **Integration** → **Firewall SSO** → **Add Firewall** → **JuniperSRX**.

- **Hostname or IP Address:** IP of your JuniperSRX
- **Firewall type:** JuniperSRX (JuniperSRX = HTTPS requests)
- **Password:** secret
- **Port:** 8443
- **Roles:** add the roles that you want to do SSO

21.7.2. Step 1: webapi configuration

You need to setup webapi management as follows

```
set system services webapi user PF
set system services webapi user password YOURPASSWORD
set system services webapi client PF_MANAGEMENT_IP_ADDRESS
```

```
set system services webapi https port PORT_YOU_WANT_TO_USE i.e. 8443
set system services webapi https default-certificate
```

Next setup user entry settings

```
set services user-identification authentication-source aruba-clearpass
authentication-entry-timeout 120
set services user-identification authentication-source aruba-clearpass no-user-
query
set services user-identification device-information authentication-source
network-access-controller
```

Then you need to allow traffic from the PacketFence management interface to port you set up on webapi settings (i.e. 8443) on SRX device.

21.7.3. Step 2: Verification

For debugging the webapi set (disable it when you won't need it anymore):

```
set system services webapi debug-log api-log
set system services webapi debug-level notice
```

To check registered device entries on SRX use

```
show services user-identification authentication-table authentication-source
all ( extensive for more detailed informations)
```

or

```
run show services user-identification device-information table all extensive
```

to see more details about OS, device type etc.

21.8. Palo Alto

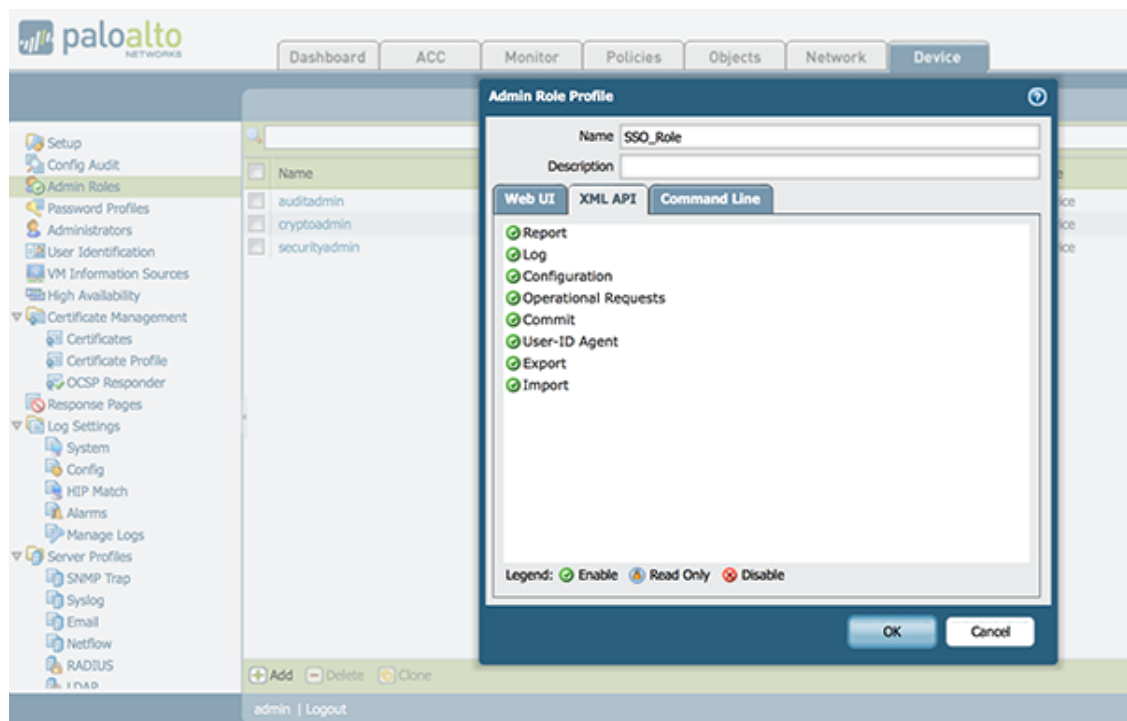
21.8.1. Installation using XMLAPI

Create a SSO role

You will first need to create an SSO role on the web interface on the PaloAlto firewall.

Go to **Device** → **Admin Roles** → **Add**.

Create the role name 'SSO_Role', under the 'XML API' tab, enable everything and validate it with 'OK'.



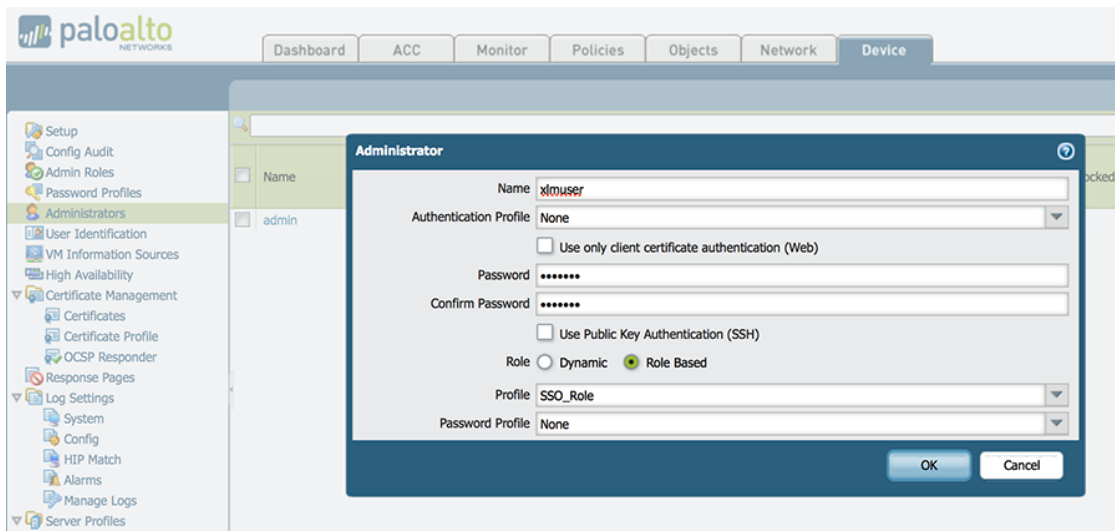
Create the account in PAN-OS

Now you have created the role, you will associate an user with it.

Go to **Device** → **Administrators** → **Add**.

- **Name:** xmluser
- **Authentication Profile:** None
- **Password:** xmluser
- **Role:** Role Based

- **Profile:** SSO_Role (Previously created)
- **Password Profile:** None



Get the XML Key

Go on this URL: <https://@IP-of-PaloAlto/api/?type=keygen&user=xmluser&password=xmluser>.

It should display:

```
<response status="success">
<result>
<key>
LUFRT1jeFV6SHd1QnJHaU55dnYvR1FNSkJNeTR6Uzg9TDgzNV1j0=
</key>
</result>
</response>
```

SSO Configuration in PacketFence

Now that we have the key, we will configure the PaloAlto firewall in PacketFence.

Go to **Configuration** → **Integration** → **Firewall SSO** → **Add Firewall** → **PaloAlto**.

- **Hostname or IP Address:** IP of your firewall
- **Transport:** HTTP
- **Secret or Key:** LUFRT1jeFV6SHd1QnJHaU55dnYvR1FNSkJNeTR6Uzg9TDgzNV1j0= (use the key previously generated)
- **Port of the service:** 443
- **Roles:** add the roles that you want to do SSO with

The screenshot displays the 'New Firewall SSO' configuration window for a Palo Alto firewall. The interface includes a sidebar on the left with categories such as 'Policies and Access Control', 'Compliance', 'Integration', 'Advanced Access Configuration', 'Network Configuration', and 'System Configuration'. The main configuration area contains the following fields and options:

- Hostname or IP Address:** 192.168.100.1
- Vsys:** 1 (Note: Please define the Virtual System number. This only has an effect when used with the HTTP transport.)
- Transport:** HTTP
- Port of the service:** 443 (Note: If you use an alternative port, please specify. This parameter is ignored when the Syslog transport is selected.)
- Secret or Key:** Masked with asterisks (Note: If using the HTTP transport, specify the password for the Palo Alto API.)
- Roles:** gaming, guest (Note: Nodes with the selected roles will be affected.)
- Networks on which to do SSO:** Empty field (Note: Comma delimited list of networks on which the SSO applies. Format : 192.168.0.0/24)
- Cache updates:** Toggle switch (Note: Enable this to debounce updates to the Firewall. By default, PacketFence will send a SSO on every DHCP request for every device. Enabling this enables "sleep" periods during which the update is not sent if the informations stay the same.)
- Cache timeout:** Empty field (Note: Adjust the "Cache timeout" to half the expiration delay in your firewall. Your DHCP renewal interval should match this value.)
- Username format:** \$pf_username (Note: Defines how to format the username that is sent to your firewall. \$username represents the username and \$realm represents the realm of your user if applicable. \$pf_username represents the unstripped username as it is stored in the PacketFence database. If left empty, it will use the username as stored in PacketFence (value of \$pf_username).)
- Default realm:** Empty field (Note: The default realm to be used while formatting the username when no realm can be extracted from the username.)

At the bottom of the configuration window, there are two buttons: 'Create' and 'Reset'.

Verification

Now we will check that PacketFence is sending information when a user registers on the portal. If the process worked, you will see the entry in the PaloAlto database.

Use SSH on the PaloAlto firewall and run this command:

```
admin@PA-VM> show user ip-user-mapping all
```

IP	Vsys	From	User	IdleTimeout(s)
192.168.100.10	vsys1	XMLAPI	domain\user1	Never

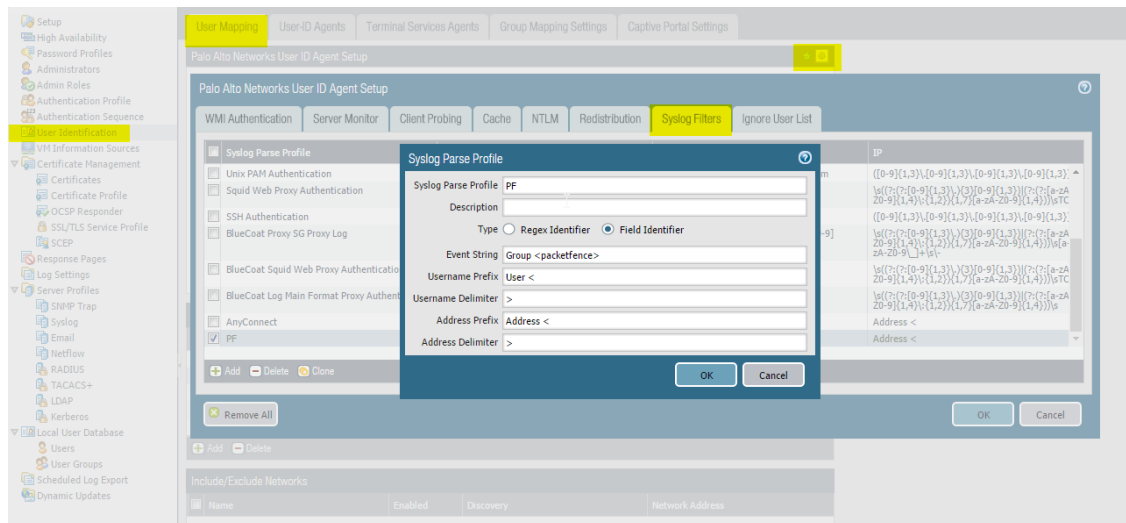
21.8.2. Installation using syslog

NOTE

This installation mode is not suggested unless you use the SSO for informational purposes (no enforcement). PacketFence will use easily spoofable UDP packets to communicate with the Palo Alto firewall. If you require encryption and origin validation of the SSO messages, please use the XML API.

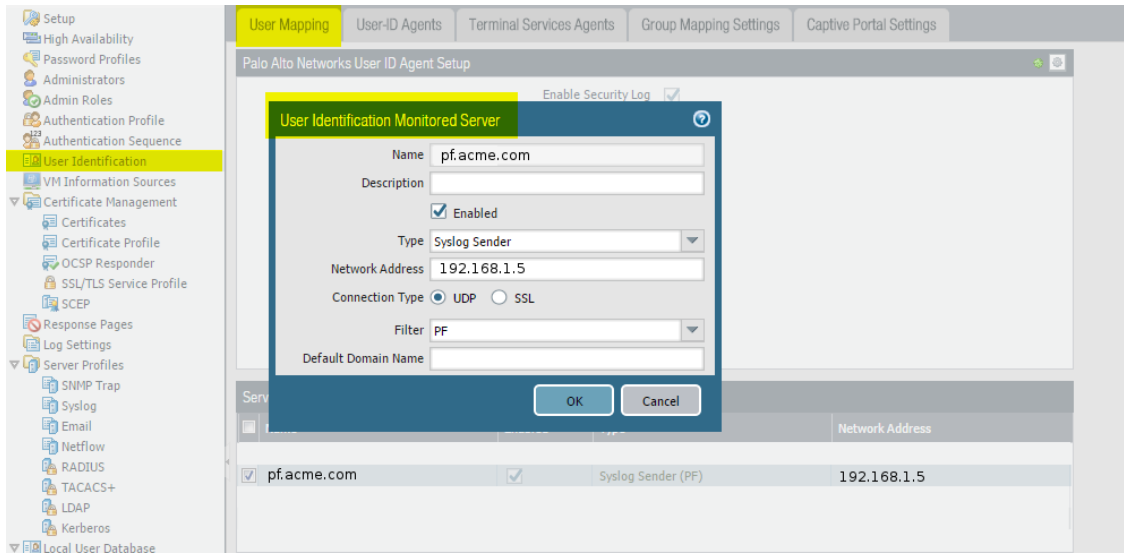
Create a filter

You will first need to create a filter to parse the SSO line that PacketFence will send. This can be done in 'User Identification→User Mapping'



Assign the filter to a 'Monitored Server'

Next, configure the filter to be used in a syslog receiver on the Palo Alto. In order to do so, go in 'User Identification→User Mapping' and configure a syslog sender.



SSO Configuration in PacketFence

Next you need to configure the firewall in PacketFence.

Go to **Configuration** → **Integration** → **Firewall SSO** → **Add Firewall** → **PaloAlto**.

- **Hostname or IP Address:** IP of your firewall
- **Transport:** Syslog
- **Secret or Key:** Ignore this parameter
- **Port of the service:** Ignore this parameter
- **Roles:** add the roles that you want to do SSO with

Verification

Now we will check that PacketFence is sending information when a user registers on the portal. If the process worked, you will see the entry in the PaloAlto database.

Use SSH on the PaloAlto firewall and run this command:

```
admin@PA-VM> show user ip-user-mapping all
```

IP	Vsys	From	User	IdleTimeout(s)
192.168.100.10	vsys1	syslog	domain\user1	Never

NOTE

If the process is not working and you get the following error **Usage: Socket::inet_ntoa(ip_address_sv)**, check that the hostname of your PacketFence server can be resolved correctly on the server itself. If its not, make sure you adjust your hosts file or your DNS server.

22. Performing Compliance Checks

PacketFence supports either Nessus, OpenVAS and WMI as a scanning engine for compliance checks. Since PacketFence v5.1 you are now able to create multiples scan engines configuration and assign them on specific captive portals. It mean per example that you are now able to active a scan for specific Operating System only on a specific SSID.

22.1. Installation

22.1.1. Nessus

Please visit <https://www.tenable.com/downloads/nessus> to download Nessus v7 and install the Nessus package for your operating system. You will also need to register for the HomeFeed (or the ProfessionalFeed) in order to get the plugins.

After you installed Nessus, follow the Nessus documentation for the configuration of the Nessus Server, and to create a user for PacketFence.

NOTE | You may run into some issue while using Nessus with the Net::Nessus::XMLRPC module (which is the default behavior in PacketFence). Please refer to the [bug tracking system](#) for more information.

22.1.2. OpenVAS

Requirements

You will first need to install OpenVAS along with XYZ and ABC in order to manage OpenVAS remotely via the `omp` command line.

In order to validate proper connectivity from PacketFence to OpenVAS for remote management, execute the following command (replacing admin by the user you wish to use for PacketFence to communicate with OpenVAS):

```
# omp -u admin -p 9390 -X "<get_version/>"
```

The output of the above command should provide you the version of OpenVAS. Otherwise, ensure all the necessary components are in place for management through the `omp` command line client and that PacketFence is able to communicate with OpenVAS on port 9390.

Configuring the alert

You will need to configure an alert policy in OpenVAS to inform PacketFence of the completion of a task. The `httpd.portal` daemon takes care of handling this callback so you'll want to make sure that you have "portal" in your additionnal listening daemons on your management interface in PacketFence.

In order to create the alert policy, go in the Greenbone Security Assistant, then in "Configuration → Alerts" and create a new alert with the following configuration

The screenshot shows the 'New Alert' configuration window. The 'Name' field contains 'Alert PacketFence'. The 'Event' section has 'Task run status changed to' selected with a dropdown set to 'Done'. The 'Condition' section has 'Always' selected. The 'Report Result Filter' is set to '..'. The 'Method' is 'HTTP Get'. The 'HTTP Get URL' is 'http://192.168.1.5/hook/openvas?task=\$n'. A 'Create' button is at the bottom right.

Where:

- Name is of the value of your choosing
- Ensure the event is set to "Task run status changed to: Done"
- Ensure the condition is set to "Always"
- Method is set to "HTTP Get"
- HTTP Get URL is set to: [http://PF_MANAGEMENT_IP/hook/openvas?task=\\$n](http://PF_MANAGEMENT_IP/hook/openvas?task=$n)
 - In the URL above, only change PF_MANAGEMENT_IP to the management IP of your PacketFence server. Leave the rest of the URL untouched as this exact URL and format is expected by PacketFence.

Collecting the identifiers

Once you have connectivity working between PacketFence and OpenVAS, use the Greenbone Security Assistant to obtain the following information for configuring PacketFence

Alert ID

Navigate to *Configuration* → *Alerts*, then click on the alert you've configured above to view it, and note down the ID of the alert.

Greenbone Security Assistant | Logged in as Admin **admin** | Logout
Tue Nov 27 17:37:46 2018 UTC

Dashboard Scans Assets SecInfo Configuration Extras Administration Help

Alert: Alert PacketFence

ID: **de0b6876-0554-41e4-befc-344619ee7b4f**
 Created: Tue Nov 27 17:06:08 2018
 Modified: Tue Nov 27 17:06:08 2018
 Owner: admin

Comment:
 Condition: Always
 Event: Task run status changed (to Done)
 Method: HTTP Get
 URL: http://192.168.1.5/hook/openvas?task=\$n
 Filter: None

Scan config ID

Navigate to *Configuration* → *Scan Configs* and then select the scan configuration you would like to use to scan the hosts. In this scan config view, note down the ID.

Greenbone Security Assistant | Logged in as Admin **admin** | Logout
Tue Nov 27 17:39:27 2018 UTC

Dashboard Scans Assets SecInfo Configuration Extras Administration Help

Scan Config: Discovery

ID: **8715c877-47a0-438d-98a3-27c7a6ab2196**
 Created: Tue Aug 21 18:53:13 2018
 Modified: Tue Aug 21 18:53:13 2018

Comment: Network Discovery scan configuration.

Report format ID

Navigate to *Configuration* → *Report Formats* and then select the **CSV Results** report format. In this view, note down the ID.

Greenbone Security Assistant | Logged in as Admin **admin** | Logout
Tue Nov 27 17:41:13 2018 UTC

Dashboard Scans Assets SecInfo Configuration Extras Administration Help

Report Format: CSV Results

ID: **c1645568-627a-11e3-a660-406186ea4fc5**
 Created: Tue Aug 21 18:53:13 2018
 Modified: Tue Aug 21 18:53:13 2018
 Owner:

Extension: csv
 Content Type: text/csv
 Trust: yes
 Active: yes
 Summary: CSV result list.

Description:
 List of results.

22.1.3. WMI

You just have to enable WMI on each Microsoft Windows devices with a GPO from Active Directory.

22.2. Configuration

In order for the compliance checks to correctly work with PacketFence (communication and generate security events inside PacketFence), you need to configure these sections:

22.2.1. Scanner Definition

First go in Configuration and Scanner Definition: *Configuration* → *Compliance* → *Scan Engines*

Then add a [New Scan Engine](#)

Scan Engine

Name

Scan Engine

Add Scan ▾

Nessus

OpenVAS

wmi

There are common parameters for each scan engines:

Name: the name of your scan engine

Roles: Only devices with these role(s) will be affected (Optional)

OS: Only devices with this Operating System will be affected (Optional)

Duration: Approximate duration of scan (Progress bar on the captive portal)
Scan before registration: Trigger the scan when the device appear on the registration vlan

Scan on registration: Trigger the scan just after registration on the captive portal

Scan after registration: Trigger the scan on the production network (pfdhcp listener must receive production dhcp traffic)

Specific to Nessus:

Hostname or IP Address: Hostname or IP Address where Nessus is running

Username: Username to connect to Nessus scan

Password: Password to connect to Nessus scan

Port of the service: port to connect (default 8834)

Nessus client policy: the name of the policy to use for the scan (Must be define on the Nessus server)

Specific to OpenVAS:

Hostname or IP Address: Hostname or IP Address where OpenVAS is running
Username: Username to connect to OpenVAS scan
Password: Password to connect to OpenVAS scan
Port of the service: port to connect (default 9390)
Alert ID: the ID of the alert configuration on the OpenVAS server
Scan config ID: the ID of the scanning configuration on the OpenVAS server
Report format ID: the ID of the report format for the "CSV Results"

Specific to WMI:

Username: A username from Active Directory that is allowed to connect to wmi
Domain: Domain of the Active Directory
Password: Password of the account
WMI Rules: Ordered list of WMI rules you defined in Configuration -> Compliance
-> Scans -> WMI Rules

22.2.2. WMI Rules Definition

If you have configured a WMI scan engine then you need to define WMI Rules. WMI is a sort of database on each windows devices, to retrieve information on the device you need to know the SQL request. In order to help you to find and make a rule you can use a third party tool like WMI Explorer.

Some example rules are defined in `/usr/local/pf/conf/wmi.conf.example` with their description. You can browse these rules in *Configuration* → *Compliance* → *WMI Rules*.

Rules syntax

The syntax of the rules are simple to understand and use same syntax as [VLAN filters](#).

- *Request* is the SQL request you will launch on the remote device, you must know what the request will return to write the test.

Inside the *Rules Actions* field we define 2 sorts of blocs:

- The test bloc (i.e. `[explorer]`)
- The action bloc (i.e. `[1:explorer]`)

The test bloc is a simple test based on the result of the request:

- attribute is the attribute you want to test
- operator can be:
 - is
 - is_not
 - match
 - match_not
 - advance

- value is the value you want to compare

You can define multiple test blocs.

The action bloc is where you will define your logic. All actions available are identical to [VLAN filters](#). Take a look at `/usr/local/pf/conf/vlan_filters.conf.example` for all available actions.

22.2.3. WMI tab

It is possible to have the result of a WMI scan in the node section. To have this, go into the rule configuration and check the box *On Node tab*. Now configure your WMI scanner as you would usually do and you will be able to have the results in the tab *WMI Rules* under Node.

22.2.4. Security Events Definition

You need to create a new security event section and have to specify:

Using Nessus:

```
trigger=Nessus::
```

Using OpenVAS:

```
trigger=OpenVAS::
```

Where **security event ID** is either the ID of the Nessus plugin or the OID of the OpenVAS plugin to check for. Once you have finished the configuration, you need to reload the security event related database contents using:

```
pfcmd reload security_events
```

NOTE | Security events will trigger if the plugin is higher than a low severity vulnerability.

22.2.5. Assign Scan definition to connection profiles

The last step is to assign one or more scanner you configured to one or more connection profiles. Go in *Configuration* → *Policies and Access Control* → *Connection Profiles* → *Edit a Profile* → *Add Scan*

Hosting Nessus / OpenVAS remotely

Because of the CPU intensive nature of an automated vulnerability assessment, we recommend that it is hosted on a separate server for large environments. To do so, a couple of things are required:

- PacketFence needs to be able to communicate to the server on the port specified by the vulnerability engine used
- The scanning server need to be able to access the targets. In other words, registration VLAN access is required if scan on registration is enabled.

If you are using the OpenVAS scanning engine:

- The scanning server need to be able to reach PacketFence's Admin interface (on port 1443 by default) by its DNS entry. Otherwise PacketFence won't be notified of completed scans.
- You must have a valid SSL certificate on your PacketFence server

If you are using the Nessus scanning engine:

- You just have to change the host value by the Nessus server IP.

22.3. Rapid7 integration

PacketFence supports integration with Rapid7 to start scans automatically when a device connects to the network and also to receive the Rapid7 alerts via syslog.

22.3.1. Rapid7 installation

- Install the InsightVM application
 - <https://insightvm.help.rapid7.com/docs/installing-in-linux-environments#section-installing-in-red-hat>
- Run the application
 - <https://insightvm.help.rapid7.com/docs/running-the-application#section-managing-the-application-in-linux>
- Logon to the server: <https://YourRapid7ServerIP:3780>

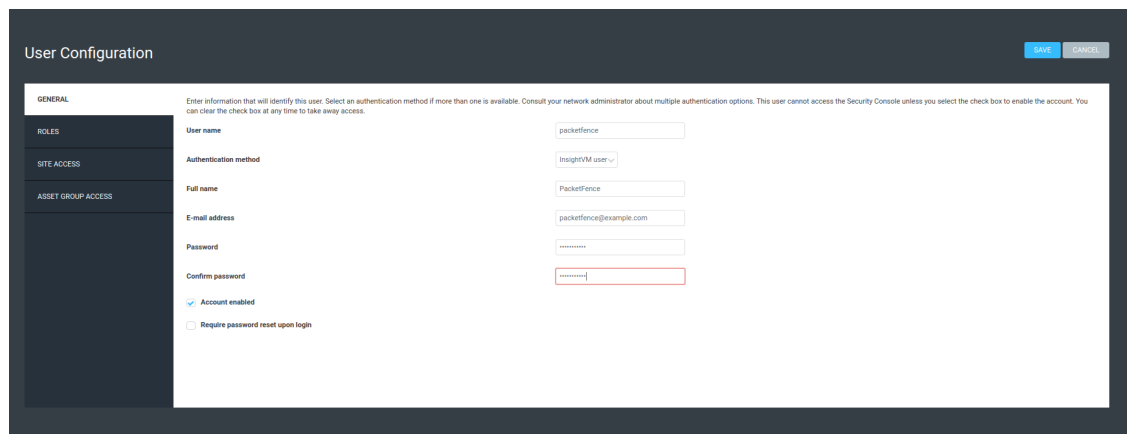
NOTE

Make sure that you create a site for the devices you want to manage in Rapid7, you will need to reference it in the PacketFence configuration

22.3.2. Configuring the scan engine

Rapid7 PacketFence user

First, you will need to create credentials for PacketFence so that it can perform API calls on Rapid7. In order to do so, on Rapid7, go in *Administration* → *Users* and click on **Create**. Then configure the appropriate username and password and make sure the account is enabled.



The screenshot shows the 'User Configuration' page in PacketFence. The 'GENERAL' tab is selected, and the 'Account enabled' checkbox is checked. The form fields are filled with the following values:

Field	Value
User name	packetfence
Authentication method	InsightVM user ✓
Full name	PacketFence
E-mail address	packetfence@example.com
Password	*****
Confirm password	*****
Account enabled	<input checked="" type="checkbox"/>
Require password reset upon login	<input type="checkbox"/>

Next, in the roles of that user, select the "Custom" role and assign at least the following privileges to the new user:

- Manage Sites
- Manage Scan Engines/pfcrn
- View Site Asset Data
- Specify Scan Targets
- View Group Asset Data

The screenshot shows the 'User Configuration' interface with the 'ROLES' tab selected. The 'Role' is set to 'Custom'. Under 'GLOBAL PERMISSIONS', the following permissions are checked:

- All Security Console Permissions: Manage all functions related to static and dynamic sites, asset groups, scans, reports, tickets, and vulnerability exceptions. Implicitly have access to all static and dynamic sites, asset groups, and reports. Implicitly own all reports. Manage all functions related to user accounts. Manage configuration, maintenance, and diagnostic operations for the Security Console. Manage vConnections. Manage shared scan credentials.
- Manage Sites: Create, delete, and configure all attributes of static and dynamic sites, except for user access. Manage shared scan credentials. Implicitly have access to all static and dynamic sites. Perform vAsset discovery.
- Manage Scan Templates: Create, delete, and configure all attributes of scan templates.
- Manage Report Templates: Create, delete, and configure all attributes of report templates.
- Manage Scan Engines: Create, delete, and configure all attributes of Scan Engines. Pair scan engines with the Security Console.
- Agree on Ticket and Report Lists: Agree on user lists in order to be assigned remediation tickets and view reports.
- Configure Global Settings: Configure settings that are applied throughout the entire Security Console environment, such as risk scoring and exclusion of assets from all scans.
- Manage Policies: Copy existing policies, edit and delete custom policies.
- Manage Tags: Create tags and configure their attributes. Delete tags except for built-in criticality tags. **Implicitly have access to all sites**

Under 'SITE PERMISSIONS', the following permissions are checked:

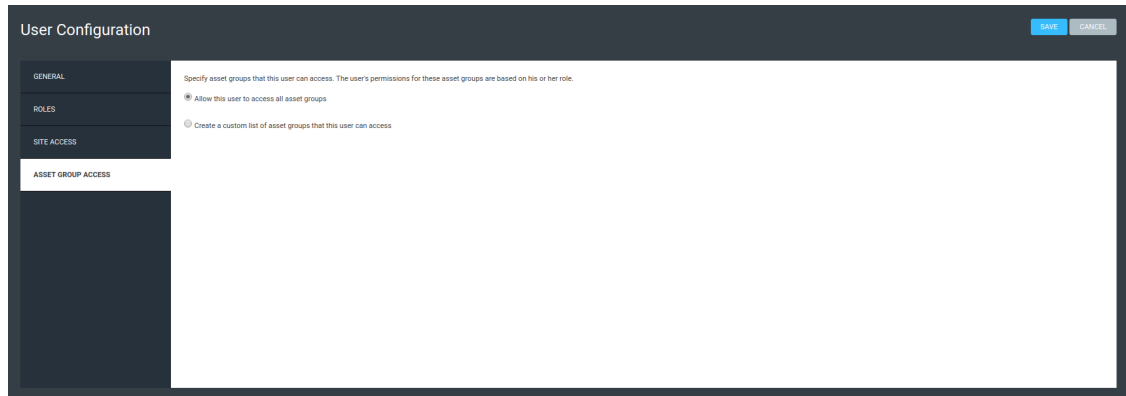
- View Site Asset Data: View discovered information about all users in accessible sites, including IP address, installed software, and vulnerabilities.
- Specify Site Metadata: Enter site descriptions, importance settings, and organization data.
- Specify Scan Targets: Add or remove IP addresses, address ranges, and host names for site scans.
- Assign Scan Engine: Assign a scan engine to sites.
- Assign Scan Template: Assign a scan template to sites.
- Manage Scan Alerts: Create, delete, and configure all attributes of alerts to notify users about scan-related events.
- Manage Site Credentials: Provide the Security Console with login credentials for deeper scanning capability on password-protected assets.
- Schedule Automatic Scans: Create and edit site scan schedules.
- Start Unscheduled Scans: Manually start one-off scans of accessible sites. This does not include ability to configure scan settings.
- Purge Site Asset Data: Manually remove asset data from accessible sites.
- Manage Site Access: Grant and remove user access to sites.

Under 'ASSET GROUP PERMISSIONS', the following permissions are checked:

- Manage Dynamic Asset Groups: Create dynamic asset groups. Delete and configure all attributes of accessible dynamic asset groups except for user access. **Implicitly have access to all sites.**
- Manage Static Asset Groups: Create static asset groups. Delete and configure all attributes of accessible static asset groups except for user access. It requires the **View Group Asset Data and Manage Group Assets** permissions.
- View Group Asset Data: View discovered information about all assets in accessible asset groups, including IP addresses, installed software, and vulnerabilities.

Next, in "Site access" and "Asset group access", ensure you provide access to this user to all the assets and sites it needs to manage. When in doubt, grant access to all sites and asset groups.

The screenshot shows the 'User Configuration' interface with the 'SITE ACCESS' tab selected. The 'Allow this user to access all sites' option is selected.



Configure the scan engine in PacketFence

Once you have the user created, you need to create the scan engine by going in *Configuration* → *Compliance* → *Scan Engines* and creating a **New Scan Engine** of the type **Rapid7**

Notes on the configuration:

- 172.20.20.230 is the IP address (hostname can also be configured) of your Rapid7 server
- Verify Hostname must be disabled unless you have a valid SSL certificate configured for the configured Rapid7 hostname
- Roles and OS represents the roles and operating systems for which you want to apply this scan engine. Leaving them empty will apply the policy to all devices.
- Scan before/on/after registration controls when the automated scans are started for the devices PacketFence sees. If you only want to start the scans manually, leave those unchecked.
- You will not be able to select a scan template, site and scan engine when initially configuring the engine. First configure the access and credentials and edit the engine again to be able to select those from the available values in Rapid7.

The screenshot shows the 'New Scan Engine' configuration form in a web application. The form is titled 'New Scan Engine' with a 'rapid7' badge. The left sidebar contains navigation menus for 'Policies and Access Control', 'Compliance', 'Scans', 'Security Events', 'Integration', 'Advanced Access Configuration', 'Network Configuration', and 'System Configuration'. The main form fields are:

- Name: MyRapid7Scan
- Hostname or IP Address: 172.20.20.230
- Username: packetfence
- Password: [masked]
- Port of the service: 3780
- Verify Hostname: [toggle off]
- Scan Engine: [dropdown]
- Scan Template: [dropdown]
- Site: [dropdown]
- Roles: [dropdown]
- OS: Type to search. [dropdown]
- Duration: 20 seconds
- Scan before registration: [toggle off]
- Scan on registration: [toggle off]
- Scan after registration: [toggle off]

At the bottom of the form, there are 'Create' and 'Reset' buttons.

Assign the engine to a connection profile

With the scan engine now created, you need to assign it to the connection profile that your endpoints use. In order to do so, go in *Configuration* → *Connection Profiles*, select your connection profile and add your scan engine there.

Automatically deregister devices on accounting stop This activates automatic deregistration of devices for the profile if PacketFence receives a RADIUS accounting stop.

VLAN pool technique: The algorithm used to calculate the VLAN in a VLAN pool.

Filters:

Filter: With no filter specified, an advanced filter must be specified.

Advanced filter:

Sources: With no source specified, all internal and external sources will be used.

Billing Tiers: With no billing tiers specified, all billing tiers will be used.

Provisioners: With no provisioners specified, the provisioners of the default profile will be used.

Scanners: With no scan specified, the scan engine will not be triggered.

Self service policy:

Viewing data on endpoints

With the scan engine integration completed, PacketFence will now automatically start scans on the endpoints it sees DHCP for and you will be able to view the Rapid7 information of the endpoints by going in the *Nodes* tab in PacketFence and then viewing a node and browsing its Rapid7 tab.

MAC 00:0c:29:30:17:84 x

[Info](#)
[Fingerbank](#)
[IPv4 Address](#)
[IPv6 Address](#)
[Location](#)
[Violations](#)
[WMI Rules](#)
[Option82](#)
[Rapid7](#)

[Summary](#)
[Device Profiling](#)
[Top Vulnerabilities](#)
[Last Scan](#)

Assessed For Policies	true
Assessed For Policies	false
OS Profiling	CentOS Linux
Risk Score	9993.353515625
Exploits Found	8
Critical Vulnerabilities Found	3
Severe Vulnerabilities Found	44
Moderate Vulnerabilities Found	4
Malware Kits Found	0

22.3.3. Configuring the syslog integration

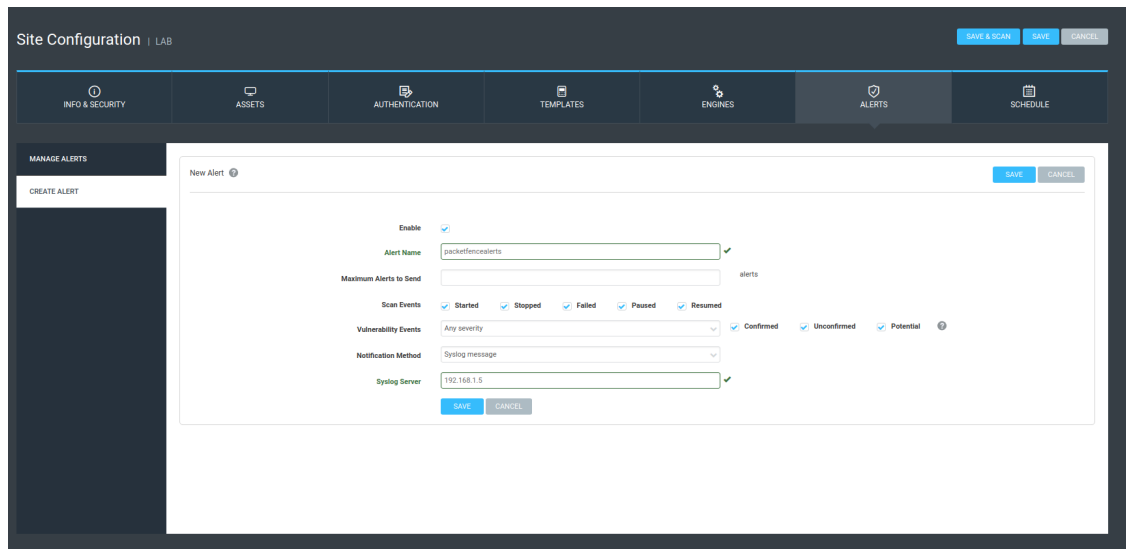
PacketFence also supports integration with the syslog forwarding of Rapid7 (with or without the scan engine integration) in order to receive vulnerability alerts from Rapid7.

Sending syslog information to PacketFence

In Rapid7:

- First select the site you want to have alerts for and click on *Manage Site*
- In the site management tabs select **Alerts**, then create a new alert

Enable: Must be checked. **Alert Name:** Rsyslog to PacketFence or else. **Maximum Alerts to send:** blank (none) **Scan events:** Check all. **Vulnerability Events:** Any severity ; Check as well *Confirmed, Unconfirmed, Potential* **Notification Method:** Select *Syslog message* **Syslog Server:** PacketFence cluster VIP or server IP for a standalone



Creating the alert pipe on PacketFence

WARNING

If you are using a PacketFence cluster, you will need to do these steps on all your PacketFence servers.

First, logon to PacketFence Server with a ssh terminal, then create the fifo pipe file that PacketFence will use to get data from Rapid7.

```
mkfifo /usr/local/pf/var/run/nexpose_pipe
```

Create a new file named /etc/rsyslog.d/nexpose-log.conf with the following content

```
# rsyslog conf for Rapid7 Nexpose server logs reception
if $programname == 'Nexpose' then /usr/local/pf/var/run/nexpose_pipe
& ~
```

Next, modify /etc/rsyslog.conf to accept syslogs data on 'udp 514' by uncommenting the following two lines:

```
$ModLoad imudp
$UDPServerRun 514
```

Restart the 'rsyslog' service

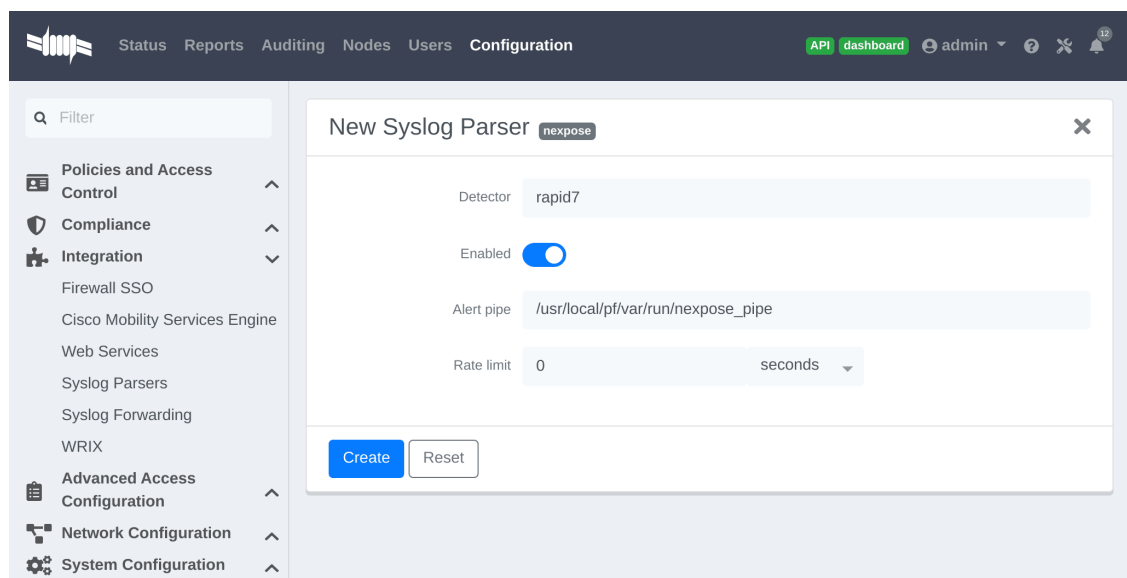
```
service rsyslog restart
```

At this point PacketFence must be able to get the Rapid7 audit results via syslog.

TIP You can see if the Nexpose server is sending to the right server by monitoring the traffic using `tcpdump -i any dst host YOUR_PACKETFENCE_SERVER_IP` on your Rapid7 Nexpose server and `tcpdump -i any src host YOUR_RAPID7_IP` on the PacketFence server.

Creating the syslog parser

In the Packetfence administration interface, go to *Configuration* → *Integration* → *Syslog parsers* and add a new Nexpose syslog parser



The screenshot shows the PacketFence administration interface. The top navigation bar includes 'Status', 'Reports', 'Auditing', 'Nodes', 'Users', and 'Configuration'. The 'Configuration' menu is expanded, showing 'Policies and Access Control', 'Compliance', 'Integration', 'Firewall SSO', 'Cisco Mobility Services Engine', 'Web Services', 'Syslog Parsers', 'Syslog Forwarding', 'WRIX', 'Advanced Access Configuration', 'Network Configuration', and 'System Configuration'. The 'Syslog Parsers' menu item is selected. The main content area displays the 'New Syslog Parser' form for a 'nexpose' parser. The form fields are: 'Detector' set to 'rapid7', 'Enabled' set to 'ON', 'Alert pipe' set to '/usr/local/pf/var/run/nexpose_pipe', and 'Rate limit' set to '0' seconds. There are 'Create' and 'Reset' buttons at the bottom of the form.

- As Detector, put the name of your choice for this parser.
- In Alert pipe, put the 'absolute' path to our nexpose pipe (`/usr/local/pf/var/run/nexpose_pipe` if you used the same name as above)

Once done, restart the following services

```
/usr/local/pf/pfcmd service pfdetect restart  
/usr/local/pf/pfcmd service pfqueue restart
```

Now that PacketFence is properly configured to receive information from Nexpose, we can configure it to perform some actions on the alerts it receives. In the PacketFence GUI, go to *Configuration* → *Compliance* → *Security Events* and create a new security event.

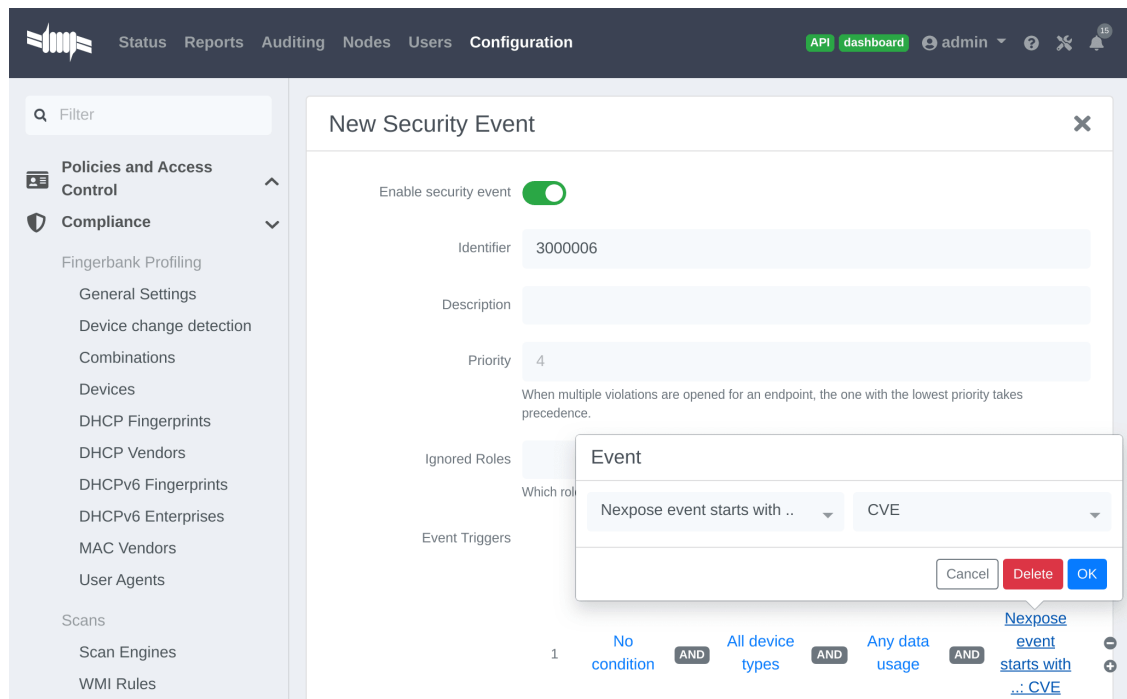
Make sure you set the following parameters in the 'Definition' tab:

- Enable: Set it to **ON**
- Action: This is where you put what you want PacketFence to do, refer to the security events

documentation in this guide for details on these.

Next, in the 'Triggers' tab:

- Click on the plus (+), on the right side of the page.
- On the second line, choose the appropriate trigger between "nexpose_event_contains" or "nexpose_event_start_with"
- Choose "nexpose_event_contains" if you know, for example, the "Common Vulnerabilities and Exposures" you want to take action on.
- For "nexpose_event_contains": You can put there the CVE or the vulnerability name you are looking for.
- For "nexpose_event_start_with": Put there the full vulnerability name you can find in the Nexpose report, on the Nexpose GUI
- Click on **ADD**, then **SAVE**



For more info on security event actions, go to the *Blocking malicious activities with security events* section of this guide.

23. Integrating Provisioning Agents

23.1. PacketFence Apple, Android and Windows Wireless Provisioning

23.2. PacketFence Apple, Android and Windows Wireless Provisioning

Provisioners allow devices to automatically configure themselves to connect to the proper SSID (if applicable), use the proper authentication method (e.g. EAP-TLS) and trust the CA certificate and any certificate signed by it.

Apple devices such as iPhones, iPads, iPods and Mac OS X (10.7+) support wireless profile importation using a special XML file format (mobileconfig). Android is also able to support this feature by importing the wireless profile with the Android PacketFence Agent. In fact, installing such file on your Apple device will automatically configure the wireless settings for a given SSID. This feature is often used when the SSID is hidden, and you want to ease the configuration steps on the mobile device (because it is often painful to configure manually). In PacketFence, we are going further, we generate the profile according to the administrator's preference and we pre-populate the file with the user's credentials (without the password). The user simply needs to install its generated file and he will be able to use the new SSID.

The Windows agent will import and apply the provisioned profile so that the user only needs to enter his username and password.

23.2.1. Configure the feature

NOTE

If EAP-TLS provisioning is desired, you have to configure a PKI before going any further. Two sections exist to assist you: [PacketFence PKI](#), which covers PacketFence's implementation, or [PacketFence MSPKI](#) which covers Microsoft's.

First of all, you need to configure the SSID that your devices will use after they go through the authentication process.

In the administration interface, go in *Configuration* → *Advanced Access Configuration* → *Provisioners*. Then select 'android' / 'ios' / 'Windows' provisioner. Enter the SSID information and roles for which the provisioner applies. Repeat for all desired provisioners. Note that the default RADIUS certificate path is `/usr/local/pf/raddb/certs/server.crt`.

After, you simply need to add the 'Android', 'iOS' and 'Windows' provisioners to your 'Connection Profile' configuration. If no connection profile is defined, configure the 'default' connection profile to use the provisioners created.

NOTE

If you use two different connection profiles for the open and secure networks, make sure you configure the provisioners on both profiles.

To add a new provisioner for another class of devices to be supported, click on the **Add Provisioner** button, and fill out the form, choosing a different Provisioning ID per provisioner.

- **Roles:** this field defines which devices will be affected by the provisioning item. If empty, all devices for this class will be affected.
- **SSID:** this field defines which SSID will be configured on the device using the authentication profile.
- **EAP-Type:** this field defines the authentication method supported and should be set to EAP-TLS to integrate with the PacketFence PKI.
- **Security type:** this field should be set to WPA2-Enterprise to integrate with the PacketFence PKI.
- **PKI Provider:** this field should match the provider you configured in the PKI provider section.

We also advise you to configure a SSID for provisioning, for instance: **OnBoarding-PF**, open with MAC Authentication, pointing to PacketFence. Create a **New Portal Profile**, add a **filter SSID** with this **SSID name**, add the source you want the users to authenticate from and add your provisioners to this Portal Profile. From there, users who logged in will have to follow the captive portal instruction to get provided their certificate.

Android specifications

For Android provisioning support, you must activate and adjust the passthroughs. You might need to adapt them depending on your geolocality.

NOTE | Please refer to the 'Passthroughs' section of this guide if needed.

In the administration interface, go in *Configuration* → *Network Configuration* → *Networks* → *Fencing*. Activate 'Passthrough' and make sure the following passthroughs domains are present:

```
*.ggpht.com,*.googleusercontent.com,android.clients.google.com,*.googleapis.com
,*.android.clients.google.com,*.gvt1.com,*.l.google.com,play.google.com,*.gstatic.com
```

Then run the following commands so that passthroughs become effective:

```
/usr/local/pf/bin/pfcmd configreload hard
/usr/local/pf/bin/pfcmd service iptables restart
/usr/local/pf/bin/pfcmd service pfdns restart
```

Next, make sure you are using a valid SSL certificate on your captive portal since Android devices will only be able to be provisioned on a captive portal that uses valid HTTPS

NOTE | Some Android devices may use their cellular connection when running the PacketFence agent during the onboarding process. If that is the case, enable the airplane mode on the Android device and then only enable WiFi during the onboarding process.

iOS specifications

Mac OS X/iOS require the provisioning profile to be signed if you want to remove the **untrusted** warning when installing the profile. For more information, please refer to the PKI guides referred earlier in 'Configure the feature' above.

Other Corporate Devices

Let's say that you now need to add some 'Linux computers' as 'corporate' devices.

Those devices cannot be authenticated via Machine Authentication, so we will need to use EAP-TLS and provide those devices with a certificate.

First of all make sure that your RADIUS certificate from the PacketFence server and the certificates that you will be provided are delivered from the same CA, else your authentication will not work. To enable EAP-TLS you will need to reconfigure the new RADIUS server certificate in the file `conf/radiusd/eap.conf`.

While creating the RADIUS server certificate make sure to have the **Extended key usage: servAuth**.

Under the section `tls-config tls-common`, search for ``private_key_file'`, ``certificate_file'` and ``ca_file'`. Those should contain respectively the path of:

- the private key for your PacketFence server,
- the server certificate issued by your CA for your PacketFence server,
- the public key of your CA.

If you have an **OCSP** capable PKI you can configure it in the section `OCSP` in the `eap.conf` file.

Lastly you will need to restart RADIUS to ensure the use of the new configuration and certificates. Please do the following:

```
/usr/local/pf/bin/pfcmd configureload hard
/usr/local/pf/bin/pfcmd service radiusd restart
```

Make sure everything happens without errors.

Now that your RADIUS is ready to handle EAP-TLS, configure your SSID connection profile on the **corporate** device using this method. Generate a client certificate for your device and install it on.

Please configure an EAPTLS source which can be found while adding a new sources under *Configuration → Policies and Access Control → Authentication Sources* **New internal Source EAPTLS**, simply give it a name, a description and a catch-all rule. This will allow you to validate the authentication via EAP-TLS.

You can now create a new Portal Profile for EAP-TLS. Under the tab configuration, section *Configuration → Policies and Access Control → Connection Profiles*, **New Connexion Profile** and select as a filter the Sub Connection Type as EAP-TLS, add your source EAP-TLS. Check the box "Automatically register devices".

You now have a full flow working for your corporate devices.

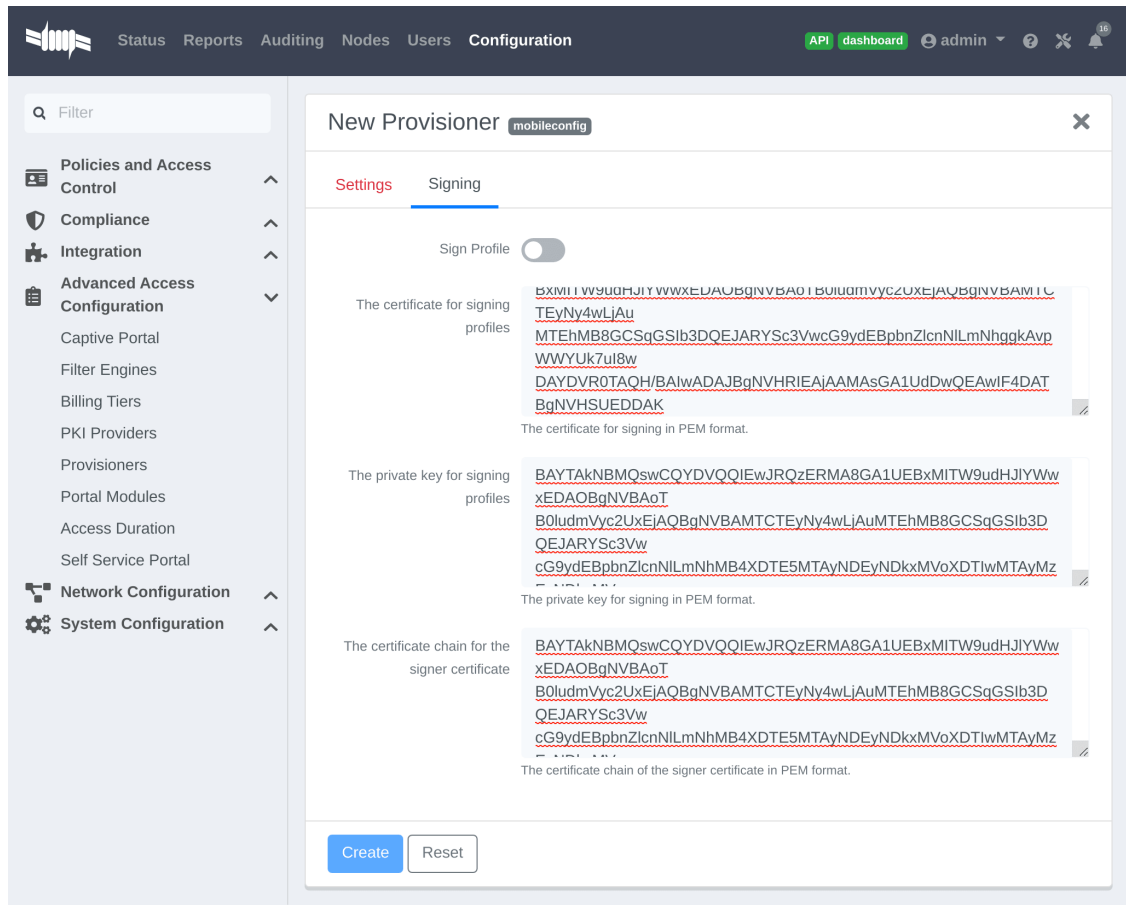
The following is an example on how to configure an EAP-TLS connection for Windows/Android/Mac OS X/iOS

The screenshot shows a web-based configuration interface for a network management system. The top navigation bar includes 'Status', 'Reports', 'Auditing', 'Nodes', 'Users', and 'Configuration'. The 'Configuration' section is active, and the user is logged in as 'admin'. The main content area is titled 'New Provisioner' and is currently set to 'windows'. The form contains the following fields:

- Provisioning ID:** EAPTLS
- Description:** Windows EAP-TLS
- Roles:** default (with a close icon)
- SSID:** PF-Secure
- Broadcast network:** (with a note: 'Uncheck this box if you are using a hidden SSID.')
- Security type:** WPA2 (with a note: 'Select the type of security applied for your SSID.')
- EAP type:** EAP-TLS (with a note: 'Select the EAP type of your SSID. Leave empty for no EAP.')
- PKI Provider:** MS-SCEP

At the bottom of the form, there are two buttons: 'Create' (in blue) and 'Reset' (in white).

Mac OS X/iOS require the provisioning profile to be signed if you want to remove the **untrusted** warning when installing the profile. You need to sign it with a Certification Authority already trusted by the device such as e.g. VeriSign. Configuring this has to be done in the 'Signing' tab in the "Apple devices".



23.2.2. Profile generation

Upon registration, instead of showing the default release page, the user will be showing another version of the page saying that the wireless profile has been generated with a clickable link on it. To install the profile, Apple user owner simply need to click on that link, and follow the instructions on their device. Android user owner simply click to the link and will be forwarded to Google Play to install PacketFence agent. Simply launch the application and click to configure will create the secure SSID profile. It is that simple.

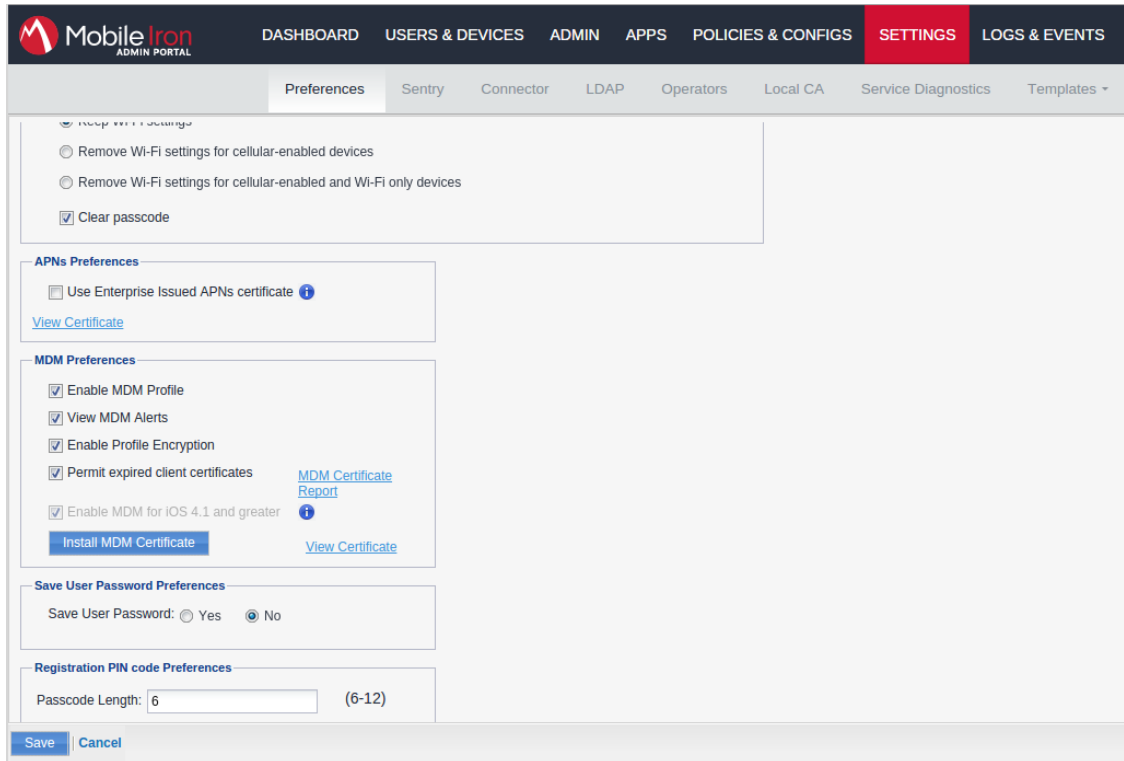
23.3. MobileIron

23.3.1. Configure MobileIron

First of all you will need to configure the basic functionality of MobileIron using their documentation.

MDM profile

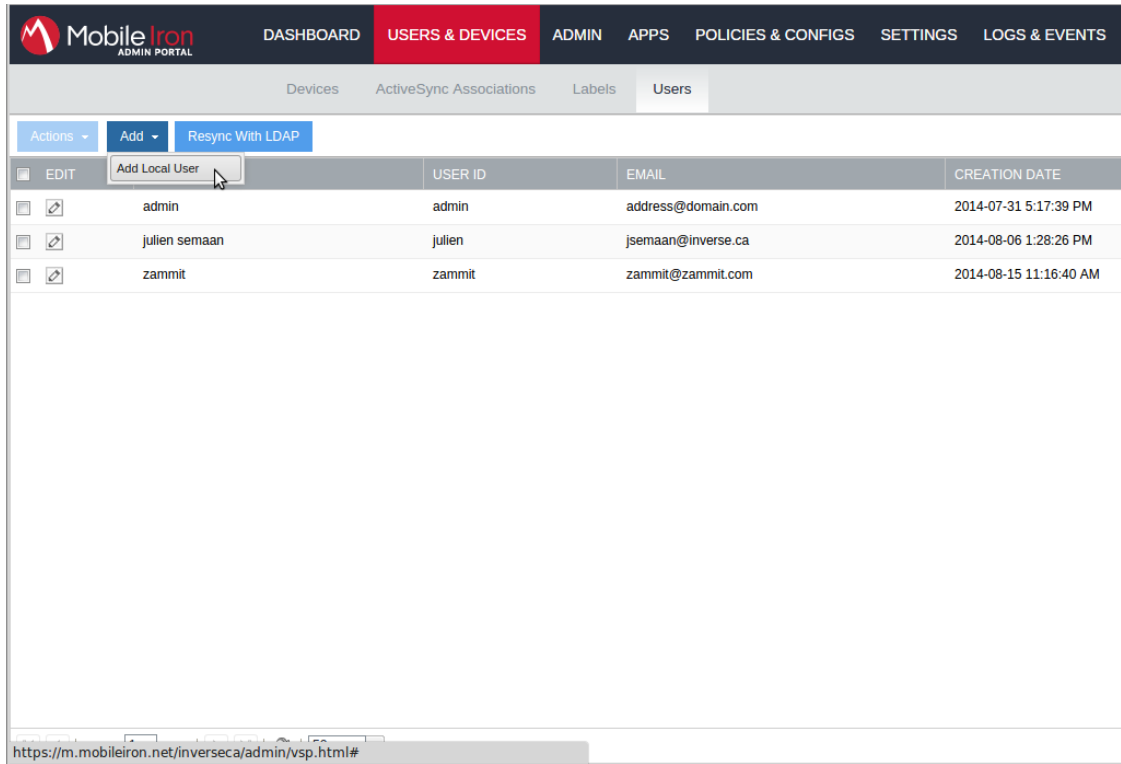
One important step is to enable the MDM profile like in this screenshot. Note that this will require you to create an MDM certificate with Apple. Refer to the MobileIron documentation for specifics about this step.



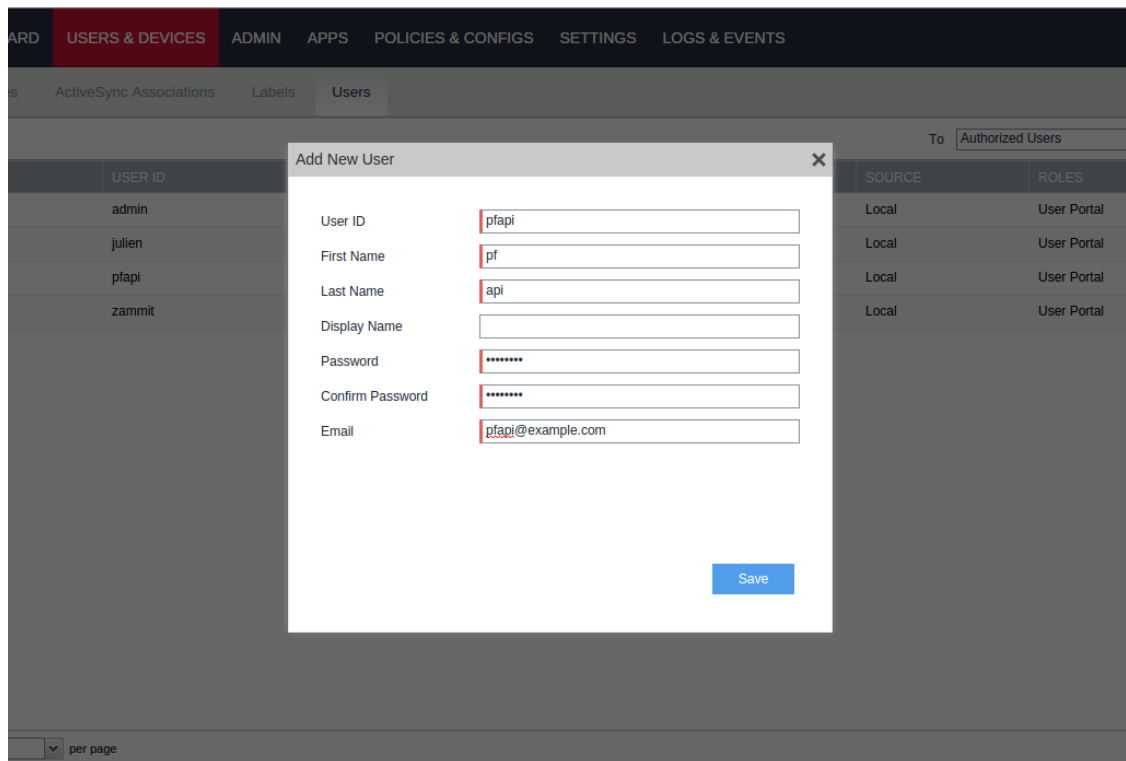
23.3.2. Create an API user

Next, we will need a user that has the rights to access the MobileIron API in order to verify the state of the devices directly from PacketFence.

First go in the 'USERS & DEVICES' tab and then in 'Users' and click 'Add local user'.

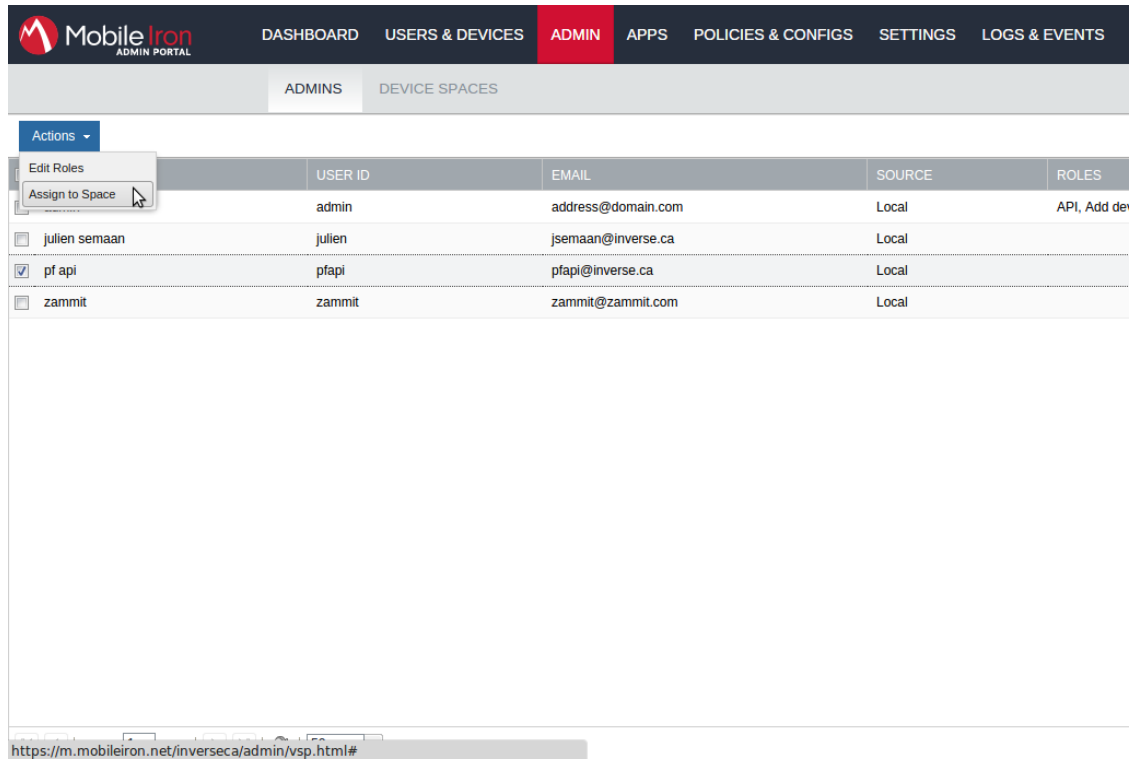


Now enter the information about your user and note the user ID and password for usage in the PacketFence configuration, then hit 'Save'.



Now go in the 'ADMIN' tab, check the box next to your newly created user and then in 'Actions'

select 'Assign to Space'.

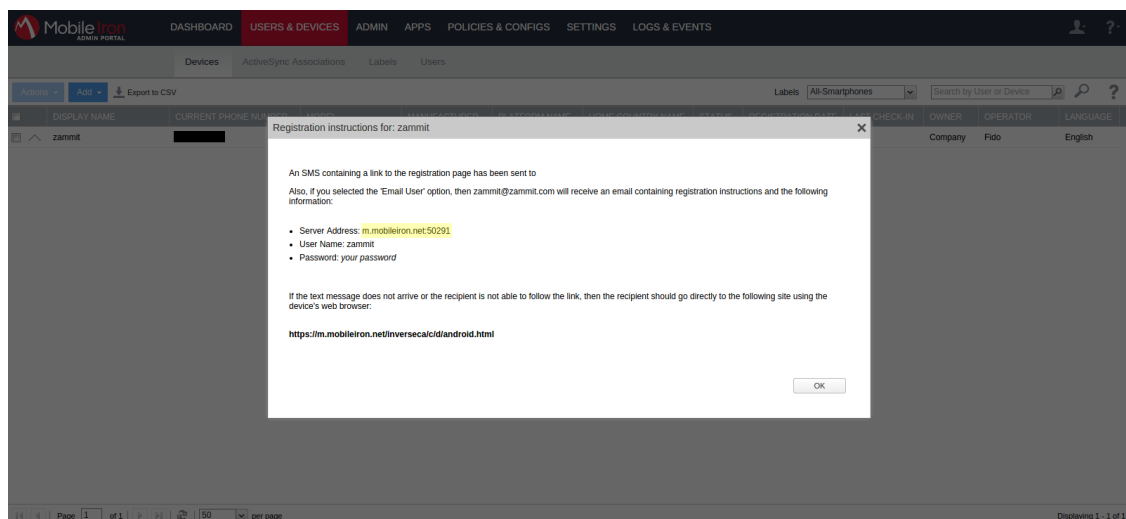


Select the Global space at the top and then check API at the bottom. You should now see API in the roles list of your newly created user when viewing the users list.

23.3.3. Gather the boarding host

To find the boarding host, add a fake device to MobileIron and at the end of the process you will see the registration instructions.

In it you will find the boarding host and port for the PacketFence configuration. In this case, the boarding host is m.mobileiron.net and the boarding port is **50291**.



23.3.4. Configure PacketFence

In PacketFence, MDM are referred to as provisioners. This will walk you through adding MobileIron as a provisioner.

Create the provisioner

Login in the PacketFence administration interface, then go in the 'Configuration' tab, then in 'Provisioners'. Click 'Add provisioner' then select 'mobileiron'.

The screenshot shows a web application interface for configuring a provisioning agent. The top navigation bar includes 'Status', 'Reports', 'Auditing', 'Nodes', 'Users', and 'Configuration'. The user is logged in as 'admin'. The left sidebar contains a 'Filter' search box and a menu with categories: 'Policies and Access Control', 'Compliance', 'Integration', 'Advanced Access Configuration', 'Captive Portal', 'Filter Engines', 'Billing Tiers', 'PKI Providers', 'Provisioners', 'Portal Modules', 'Access Duration', 'Self Service Portal', 'Network Configuration', and 'System Configuration'. The main panel is titled 'Provisioner mobileiron' and contains the following configuration fields:

- Provisioning ID: mobileiron
- Description: Mobile Iron
- Roles: (dropdown menu)
- OS: Type to search. (dropdown menu)
- Username: admin
- Client Secret: (masked with dots)
- Host: m.mobileiron.ca/inverseca
- Android download URI: https://m.mobileiron.net/accountName/c/d/android.html
- IOS download URI: https://m.mobileiron.net/accountName/c/d/ios.html
- Windows phone download URI: https://m.mobileiron.net/accountName/EnrollmentServer/Discovery.svc
- Boarding host: m.mobileiron.net
- Boarding port: 50291

At the bottom of the configuration panel are four buttons: 'Save', 'Reset', 'Clone', and 'Delete'.

Now configure this new provisioner with the information you got above.

- The Provisioning ID is the friendly name of the provisioner.
- The Username is the user you created with API access above.
- The password is the password of the API user.

- The host is the domain name of the instance + your account name if you have a cloud account (ex: m.mobileiron.net/accountName)
- Now add the download URI for the agent. See below for more details.
- The Boarding host is the host that you got in step 3.
- The Boarding port is the port that you got in step 3.

Here are the URIs that should work by default. Replace **accountName** by your real account/instance name at MobileIron.

- Android: <https://m.mobileiron.net/accountName/c/d/android.html>
- IOS devices: <https://m.mobileiron.net/accountName/c/d/ios.html>
- Windows: <https://m.mobileiron.net/accountName/EnrollmentServer/Discovery.svc>

Add the provisioner to the connection profile

In order for the provisioner to be used by your captive portal you need to add it in its configuration. Go in 'Connection Profiles', then select the portal you want to modify and add 'mobileiron' as a provisioner.

[Status](#)
[Reports](#)
[Auditing](#)
[Nodes](#)
[Users](#)
[Configuration](#)

[API dashboard](#)
admin

Filter

- Policies and Access Control**
- Roles
- Domains
 - Active Directory Domains
 - Realms
- Authentication Sources
- Network Devices
 - Switches
 - Switch Groups
- Connection Profiles
- Compliance**
- Integration**
- Advanced Access Configuration**
- Network Configuration**
- System Configuration**

Connection Profile default Preview

Settings [Captive Portal](#) [Files](#)

Profile Name 🔒
A profile id can only contain alphanumeric characters, dashes, period and or underscores.

Profile Description

Root Portal Module ▼
The Root Portal Module to use.

Activate preregistration

This activates preregistration on the connection profile. Meaning, instead of applying the access to the currently connected device, it displays a local account that is created while registering. Note that activating this disables the on-site registration on this connection profile. Also, make sure the sources on the connection profile have "Create local account" enabled.

Automatically register devices

This activates automatic registration of devices for the profile. Devices will not be shown a captive portal and RADIUS authentication credentials will be used to register the device. This option only makes sense in the context of an 802.1x authentication.

Reuse dot1x credentials

This option emulates SSO when someone needs to face the captive portal after a successful 802.1x connection. 802.1x credentials are reused on the portal to match an authentication and get the appropriate actions. As a security precaution, this option will only reuse 802.1x credentials if there is an authentication source matching the provided realm. This means, if users use 802.1x credentials with a domain part (username@domain, domain/username), the domain part needs to be configured as a realm under the RADIUS section and an authentication source needs to be configured for that realm. If users do not use 802.1x credentials with a domain part, only the NULL realm will match if an authentication source is configured for it.

Dot1x recompute role from portal

When enabled, PacketFence will not use the role initially computed on the portal but will use the dot1x username to recompute the role.

MAC Auth recompute role from portal

When enabled, PacketFence will not use the role initially computed on the portal but will use an authorized source if defined to recompute the role.

Dot1x unset on unmatched

When enabled, PacketFence will unset the role of the device if no authentication sources returned one.

Enable DPSK

This enables the Dynamic PSK feature on this connection profile. It means that the RADIUS server will answer requests with specific attributes like the PSK key to use to connect on the SSID.

Default PSK key

This is the default PSK key when you enable DPSK on this connection profile. The minimum length is eight characters.

Automatically deregister devices on accounting stop

This activates automatic deregistration of devices for the profile if PacketFence receives a RADIUS accounting stop.

VLAN pool technique ▼
The algorithm used to calculate the VLAN in a VLAN pool.

Sources

1	<input type="text" value="null"/>	-	+
---	-----------------------------------	---	---

Billing Tiers With no billing tiers specified, all billing tiers will be used.

Provisioners

1	<input type="text" value="mobileiron"/>	-	+
---	---	---	---

Scanners With no scan specified, the scan engine will not be triggered.

Self service policy

Save
Reset
Clone

23.3.5. Add the necessary passthroughs

Next, still in the PacketFence administration console, go in 'Fencing' in the left menu, then scroll then to 'Passthroughs'.

Check the 'Passthrough' box above the field and add the following domains to the passthrough list.

- m.mobileiron.net
- *.itunes.apple.com
- itunes.apple.com
- play.google.com
- *.play.google.com

[Status](#) [Reports](#) [Auditing](#) [Nodes](#) [Users](#) [Configuration](#)

[API](#) [dashboard](#) [admin](#)

Filter

- [Policies and Access Control](#)
- [Compliance](#)
- [Integration](#)
- [Advanced Access Configuration](#)
- [Network Configuration](#)
- Networks
 - [Network Settings](#)
 - [Interfaces](#)
 - [Inline](#)
 - [Inline Traffic Shaping](#)
 - [Fencing](#)
 - [Device Parking](#)
- [SNMP](#)
- [Floating Devices](#)
- [System Configuration](#)

Networks

[Network Settings](#) [Interfaces](#) [Inline](#) [Inline Traffic Shaping](#) [Fencing](#) [Device Parking](#)

Fencing

Wait for redirect

How many seconds the webservice should wait before deassociating or reassigning VLAN. If we don't wait, the device may switch VLAN before it has a chance to load the redirection page.

Whitelist

Comma-separated list of MAC addresses that are immune to isolation. In inline Level 2 enforcement, the firewall is opened for them as if they were registered. This feature will probably be reworked in the future.

Addresses ranges

Address ranges/CIDR blocks that PacketFence will monitor/detect/trap on. Gateway, network, and broadcast addresses are ignored. Comma-separated entries should be of the form a.b.c.0/24 a.b.c.0-255 a.b.c.0-a.b.c.255 a.b.c.d

Passthrough

When enabled, PacketFence uses pfdns if you defined Passthroughs or Apache mod-proxy if you defined Proxy passthroughs to allow trapped devices to reach web sites. Modifying this parameter requires to restart pfdns and iptables to be fully effective.

Passthroughs Domains

Comma-separated list of domains to allow access from the registration VLAN. If no port is specified for the domain (ex: example.com), it opens TCP 80 and 443. You can specify a specific port to open (ex: example.com:tcp:25) which opens port 25 in TCP. When no protocol is specified (ex: example.com:25), this opens the port for both the UDP and TCP protocol. You can specify the same domain with a different port multiple times and they will be combined. The configuration parameter passthrough must be enabled for passthroughs to be effective. These passthroughs are only effective in registration networks, for passthroughs in isolation, use fencing_isolation_passthroughs.

Proxy Passthroughs

Built-in Proxy Passthroughs:
[crl.geotrust.com](#) [ocsp.geotrust.com](#)
[crl.thawte.com](#) [ocsp.thawte.com](#) [crl.comodoca.com](#) [ocsp.comodoca.com](#)
[crl.incommon.org](#) [ocsp.incommon.org](#) [crl.usertrust.com](#) [ocsp.usertrust.com](#)
[mscrl.microsoft.com](#) [crl.microsoft.com](#) [ocsp.apple.com](#) [ocsp.digicert.com](#)
[ocsp.entrust.com](#) [svint-crl.verisign.com](#) [ocsp.verisign.com](#)
[crl.windowsupdate.com](#) [crl.globalsign.net](#) [pki.google.com](#) [www.microsoft.com](#)
[crl.godaddy.com](#) [ocsp.godaddy.com](#) [certificates.godaddy.com](#)
[crl.globalsign.com](#) [secure.globalsign.com](#) [cacerts.digicert.com](#)
[crl.comodoca.com](#) [crl.incommon-rsa.org](#) [crl.quovadisglobal.com](#)
[crl.incommon.org](#) [crl.usertrust.com](#) [crl.verisign.com](#) [crl.starfieldtech.com](#)
[developer.apple.com](#) [ts-crl.ws.symantec.com](#) [certificates.intel.com](#)

Comma-separated list of domains to be used with apache passthroughs. The configuration parameter passthrough must be enabled for passthroughs to be effective.

Isolation Passthrough

When enabled, PacketFence uses pfdns if you defined Isolation Passthroughs to allow trapped devices in isolation state to reach web sites. Modifying this parameter requires to restart pfdns and iptables to be fully effective.

Isolation Passthroughs Domains

Comma-separated list of domains to allow access from the isolation VLAN. If no port is specified for the domain (ex: example.com), it opens TCP 80 and 443. You can specify a specific port to open (ex: example.com:tcp:25) which opens port 25 in TCP. When no protocol is specified (ex: example.com:25), this opens the port for both the UDP and TCP protocol. You can specify the same domain with a different port multiple times and they will be combined. The configuration parameter isolation_passthrough must be enabled for passthroughs to be effective.

Proxy Interception

If enabled, we will intercept proxy request on the specified ports to forward to the captive portal.

Proxy Interception Port

Comma-separated list of port used by proxy interception.

[Save](#)
[Reset](#)
 pfdns
 iptables

Restart PacketFence

In order to enable the boarding passthrough for the device enrollment, you will need to restart the iptables service of PacketFence.

You can do this using the command line by doing `/usr/local/pf/bin/pfcmd service iptables restart` or in the administration interface under 'Status / Services'.

23.3.6. Testing

You can now test that MobileIron is mandatory after the device registration. Connect a device to your test network and register like you normally would. At the end of the registration process you will be presented a page asking you to install the MobileIron on your device. After you install the agent click 'Continue'. If your access is enabled than this means the connectivity between PacketFence and MobileIron is good.

23.4. OPSWAT

23.4.1. Configure OPSWAT Metadefender Endpoint

You will first need to create an OPSWAT Metadefender Endpoint account at <https://www.opswat.com/products/metadefender/endpoint/management/> and configure your account according to OPSWAT's documentation.

23.4.2. Developer account

Now that you have basic functionality for your OPSWAT Metadefender Endpoint account, you will need to create a Metadefender Endpoint developer account so PacketFence can access the OPSWAT Metadefender Endpoint API. You can do this here <https://gears.opswat.com/developers>.

Creating the application

Once this is done, click 'Register a new application'. The only thing important here is to set the callback URL to <http://127.0.0.1/opswat>.

Once you created the application, note the client key and client secret for usage below.

23.4.3. Gathering the install URL




From your OPSWAT Metadefender Endpoint console, click '+Devices' at the top. Then click on Enable 'Metadefender Endpoint client on another device'.

Then click 'Download or send link for guest Metadefender Endpoint clients'







×

Add devices

To monitor more devices, simply download the Gears client and run on those machines. Gears will send device information to your cloud account and enable you to begin managing the devices from the cloud.

 **Download managed Gears clients for distribution**
  Windows and Mac only

- or -


 **Download or send link for guest Gears clients**
     Windows, Mac, Linux, Android, iOS

Then note the URL at the bottom of the screen.


×

Add guest devices


After users download and run the client, you will be able to monitor and manage their devices through your Gears account.

 **Go to the guest device download page**

- or -

 **Email the download link**

- or -

 **Send the link via chat**
Paste the following message into your chat client window:

We are using OPSWAT Gears to manage the network. Please follow the instructions at this link to enable your device with OPSWAT Gears:
<https://gears.opswat.com/gears/a/download/4655c62dd5b9e12c873e2b7f0944446b>

Click to copy text

23.4.4. API access

In order to configure OPSWAT Metadefender Endpoint in PacketFence you will need to generate an OAuth2 access and refresh token so PacketFence can access the OPSWAT Metadefender Endpoint API.

Generate the authorization code

First you will access this page using your browser (replace `-clientid-` by your client ID that you got when creating the application):

```
https://gears.opswat.com/o/oauth/authorize?client_id=-clientid-  
&response_type=code&redirect_uri=http://127.0.0.1/opswat
```

Authorize the application and you will then be redirected to an unavailable page but the URL will contain the code in its parameters (ex: <http://127.0.0.1/opswat?code=wJ2RTE>).

Generate the access and refresh token

We will now use the code at the end to generate the access and refresh token using another HTTP request that will be done in your browser. Replace `-clientid-` and `-clientsecret-` by the client id and secret of your application. Then add the code you got above at the end of this URL.

```
https://gears.opswat.com/o/oauth/token?client_id=-clientid-&client_secret=-  
clientsecret-  
&grant_type=authorization_code&redirect_uri=http://127.0.0.1/opswat&code=
```

You should now be presented with a JSON response that contains the access and refresh token. Take note of both of these values for the PacketFence configuration. Example:

```
{"access_token": "ab3aec71-fa6a-4752-8804-00c37f934059", "token_type": "bearer",  
  "refresh_token": "f9e7c698-4d88-42cb-b9ae-c067557e8385", "expires_in": 43199,  
  "scope": "read", "client_id": "1234567890"}
```

23.4.5. Configure PacketFence

Create a new provisioner

Login in the PacketFence administration interface, then go in the 'Configuration' tab, then in 'Provisioners'. Click 'Add provisioner' then select opswat.

The screenshot shows a web-based configuration interface for a Provisioner named 'opswat'. The interface is divided into a sidebar on the left and a main content area on the right. The sidebar contains a search bar and a list of navigation items: Policies and Access Control, Compliance, Integration, Advanced Access Configuration, Captive Portal, Filter Engines, Billing Tiers, PKI Providers, Provisioners, Portal Modules, Access Duration, Self Service Portal, Network Configuration, and System Configuration. The main content area is titled 'Provisioner opswat' and has two tabs: 'Settings' and 'Compliance'. The 'Compliance' tab is active, showing a form with the following fields: Provisioning ID (opswat), Description (OPSWAT), Roles (dropdown), OS (dropdown), Client Key (1234567890), Client Secret (0987654321), Host (gears.opswat.com), Port (443), Protocol (https), Access token (b5275f8c-a22c-4260-8090-696c2b3), Refresh token (ec532cc4-0d78-426e-8c44-1411c5b), and Agent download URI (https://gears.opswat.com/gears/a/download/4655c62dd5b9e12c873e2b7f094). At the bottom of the form are four buttons: Save, Reset, Clone, and Delete.

Now configure this new provisioner with the information you got above.

- The Provisioning ID is the friendly name of the provisioner.
- The Client Id is the ID of the application you created in the developer account.
- The Client Secret is the secret of the application you created in the developer account.

- The default host should work if you have a cloud account, if not adapt it to your local instance.
- The port and protocol should be left to default.
- The access and refresh token are the tokens you got at the end of step 4.
- The 'Agent download uri' is the one you got in step 3.

Add the provisioner to the profile

Now that you have created the provisioner, go in the 'Connection Profiles' menu on the left and select the default portal. Click 'Add Provisioner' and select the new OPSWAT Metadefender Endpoint provisioner that was created earlier.

[Status](#) [Reports](#) [Auditing](#) [Nodes](#) [Users](#) **Configuration**

[API dashboard](#) [admin](#) [?](#) [✖](#) [🔔](#)

Filter

- Policies and Access Control**
- Roles
- Domains
- Active Directory Domains
- Realms
- Authentication Sources
- Network Devices
- Switches
- Switch Groups
- Connection Profiles
- Compliance**
- Integration**
- Advanced Access Configuration**
- Network Configuration**
- System Configuration**

Connection Profile default Preview

Settings **Captive Portal** Files

Profile Name: 🔒
A profile id can only contain alphanumeric characters, dashes, period and or underscores.

Profile Description:

Root Portal Module: ▼
The Root Portal Module to use.

Activate preregistration:

This activates preregistration on the connection profile. Meaning, instead of applying the access to the currently connected device, it displays a local account that is created while registering. Note that activating this disables the on-site registration on this connection profile. Also, make sure the sources on the connection profile have "Create local account" enabled.

Automatically register devices:

This activates automatic registration of devices for the profile. Devices will not be shown a captive portal and RADIUS authentication credentials will be used to register the device. This option only makes sense in the context of an 802.1x authentication.

Reuse dot1x credentials:

This option emulates SSO when someone needs to face the captive portal after a successful 802.1x connection. 802.1x credentials are reused on the portal to match an authentication and get the appropriate actions. As a security precaution, this option will only reuse 802.1x credentials if there is an authentication source matching the provided realm. This means, if users use 802.1x credentials with a domain part (username@domain, domain/username), the domain part needs to be configured as a realm under the RADIUS section and an authentication source needs to be configured for that realm. If users do not use 802.1x credentials with a domain part, only the NULL realm will be match if an authentication source is configured for it.

Dot1x recompute role from portal:
When enabled, PacketFence will not use the role initially computed on the portal but will use the dot1x username to recompute the role.

MAC Auth recompute role from portal:
When enabled, PacketFence will not use the role initially computed on the portal but will use an authorized source if defined to recompute the role.

Dot1x unset on unmatched:
When enabled, PacketFence will unset the role of the device if no authentication sources returned one.

Enable DPSK:

This enables the Dynamic PSK feature on this connection profile. It means that the RADIUS server will answer requests with specific attributes like the PSK key to use to connect on the SSID.

Default PSK key:

This is the default PSK key when you enable DPSK on this connection profile. The minimum length is eight characters.

Automatically deregister devices on accounting stop:
This activates automatic deregistration of devices for the profile if PacketFence receives a RADIUS accounting stop.

VLAN pool technique: ▼
The algorithm used to calculate the VLAN in a VLAN pool.

Sources: With no source specified, all internal and external sources will be used.

Billing Tiers: With no billing tiers specified, all billing tiers will be used.

Provisioners: 1 ▼ ⊖ ⊕

Scanners: With no scan specified, the scan engine will not be triggered.

Self service policy:

23.4.6. Add the necessary passthroughs

Next, still in the PacketFence administration console, go in 'Fencing' in the left menu, then scroll then to 'Passthroughs'. Check the 'Passthrough' box above the field and add the following domains to the passthrough list.

- gears.opswat.com
- software.opswat.com
- opswat-gears-cloud-clients.s3.amazonaws.com

Status Reports Auditing Nodes Users Configuration
API dashboard admin

Filter

- Policies and Access Control
- Compliance
- Integration
- Advanced Access Configuration
- Network Configuration
 - Networks
 - Network Settings
 - Interfaces
 - Inline
 - Inline Traffic Shaping
 - Fencing**
 - Device Parking
 - SNMP
 - Floating Devices
- System Configuration

Networks

[Network Settings](#)
[Interfaces](#)
[Inline](#)
[Inline Traffic Shaping](#)
[Fencing](#)
[Device Parking](#)

Fencing

Wait for redirect:

How many seconds the webservice should wait before deassociating or reassigning VLAN. If we don't wait, the device may switch VLAN before it has a chance to load the redirection page.

Whitelist

Comma-separated list of MAC addresses that are immune to isolation. In inline Level 2 enforcement, the firewall is opened for them as if they were registered. This feature will probably be reworked in the future.

Addresses ranges

Address ranges/CIDR blocks that PacketFence will monitor/detect/trap on. Gateway, network, and broadcast addresses are ignored. Comma-separated entries should be of the form
 a.b.c.0/24
 a.b.c.0-255
 a.b.c.0-a.b.c.255
 a.b.c.d

Passthrough

When enabled, PacketFence uses pfdrns if you defined Passthroughs or Apache mod-proxy if you defined Proxy passthroughs to allow trapped devices to reach web sites. Modifying this parameter requires to restart pfdrns and iptables to be fully effective.

Passthroughs Domains

[qears.opswat.com](#)
[software.opswat.com](#)

Comma-separated list of domains to allow access from the registration VLAN. If no port is specified for the domain (ex: example.com), it opens TCP 80 and 443. You can specify a specific port to open (ex: example.com:tcp:25) which opens port 25 in TCP. When no protocol is specified (ex: example.com:25), this opens the port for both the UDP and TCP protocol. You can specify the same domain with a different port multiple times and they will be combined. The configuration parameter passthrough must be enabled for passthroughs to be effective. These passthroughs are only effective in registration networks, for passthroughs in isolation, use fencing_isolation_passthroughs.

Proxy Passthroughs

Built-in Proxy Passthroughs: [crl.geotrust.com](#) [ocsp.geotrust.com](#)
[crl.thawte.com](#) [ocsp.thawte.com](#) [crl.comodoca.com](#) [ocsp.comodoca.com](#)
[crl.incommon.org](#) [ocsp.incommon.org](#) [crl.usertrust.com](#) [ocsp.usertrust.com](#)
[mscrl.microsoft.com](#) [crl.microsoft.com](#) [ocsp.apple.com](#) [ocsp.digicert.com](#)
[ocsp.entrust.com](#) [srvm1-crl.verisign.com](#) [ocsp.verisign.com](#)
[crtld.windowsupdate.com](#) [crl.globalign.net](#) [pki.google.com](#) [www.microsoft.com](#)
[crl.godaddy.com](#) [ocsp.godaddy.com](#) [certificates.godaddy.com](#)
[crl.globalign.com](#) [secure.globalign.com](#) [cacerts.digicert.com](#)
[crl.comodoca.com](#) [crl.incommon-fsa.org](#) [crl.quovadisglobal.com](#)
[cert.incommon.org](#) [crl.usertrust.com](#) [crl.verisign.com](#) [crl.starfieldtech.com](#)
[developer.apple.com](#) [ts-crl.ws.symantec.com](#) [certificates.intel.com](#)

Comma-separated list of domains to be used with apache passthroughs. The configuration parameter passthrough must be enabled for passthroughs to be effective.

Isolation Passthrough

When enabled, PacketFence uses pfdrns if you defined Isolation Passthroughs to allow trapped devices in isolation state to reach web sites. Modifying this parameter requires to restart pfdrns and iptables to be fully effective.

Isolation Passthroughs Domains

Comma-separated list of domains to allow access from the isolation VLAN. If no port is specified for the domain (ex: example.com), it opens TCP 80 and 443. You can specify a specific port to open (ex: example.com:tcp:25) which opens port 25 in TCP. When no protocol is specified (ex: example.com:25), this opens the port for both the UDP and TCP protocol. You can specify the same domain with a different port multiple times and they will be combined. The configuration parameter isolation_passthrough must be enabled for passthroughs to be effective.

Proxy Interception

If enabled, we will intercept proxy request on the specified ports to forward to the captive portal.

Proxy Interception Port

8080
3128

Comma-separated list of port used by proxy interception.

pfdrns
 iptables

23.4.7. Testing

You can now test that the installation of the OPSWAT Metadefender Endpoint client is mandatory after the device registration. Connect a device to your test network and register like you normally would. At the end of the registration process you will be presented a page asking you to install the OPSWAT Metadefender Endpoint client on your device. After you install the client click continue. If your access is enabled than this means the connectivity between PacketFence and OPSWAT Metadefender Endpoint is good.

23.4.8. Compliance enforcement

PacketFence polls the OPSWAT Metadefender Endpoint API in order to trigger security events on noncompliant devices.

PacketFence uses the number of critical issues the device has to determine whether or not it needs to isolate it.

Configure OPSWAT Metadefender Endpoint

First you need to configure what you consider as a critical issue in your OPSWAT Metadefender Endpoint console.

You will do that through the 'Configure' menu. Then you'll see a column that allows you to flag what is considered as a critical issue.

Consider an issue	Critical	All	Desktops	Laptops	VMs	Servers
Antiphishing		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> Require at least one antiphishing product to be enabled	<input type="checkbox"/>					
Antivirus		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> Report if no antivirus application is installed	<input checked="" type="checkbox"/>					
<input checked="" type="checkbox"/> Require real time protection from at least one antivirus product	<input checked="" type="checkbox"/>					
<input type="checkbox"/> Attempt to enable real time protection in all antivirus products	<input type="checkbox"/>					
<input checked="" type="checkbox"/> Require at least one antivirus product to have definitions less than 3 days old	<input checked="" type="checkbox"/>					
<input type="checkbox"/> Attempt to update all antivirus definitions	<input type="checkbox"/>					
<input checked="" type="checkbox"/> Require full system scan from at least one antivirus in the last 7 days	<input type="checkbox"/>					
<input checked="" type="checkbox"/> Report if at least one antivirus has detected any threats in the last 7 days	<input type="checkbox"/>					
Backup		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> Report if no backup application is installed	<input type="checkbox"/>					
<input checked="" type="checkbox"/> Report if no backup activity in the last 7 days	<input type="checkbox"/>					
Encryption		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Configure PacketFence

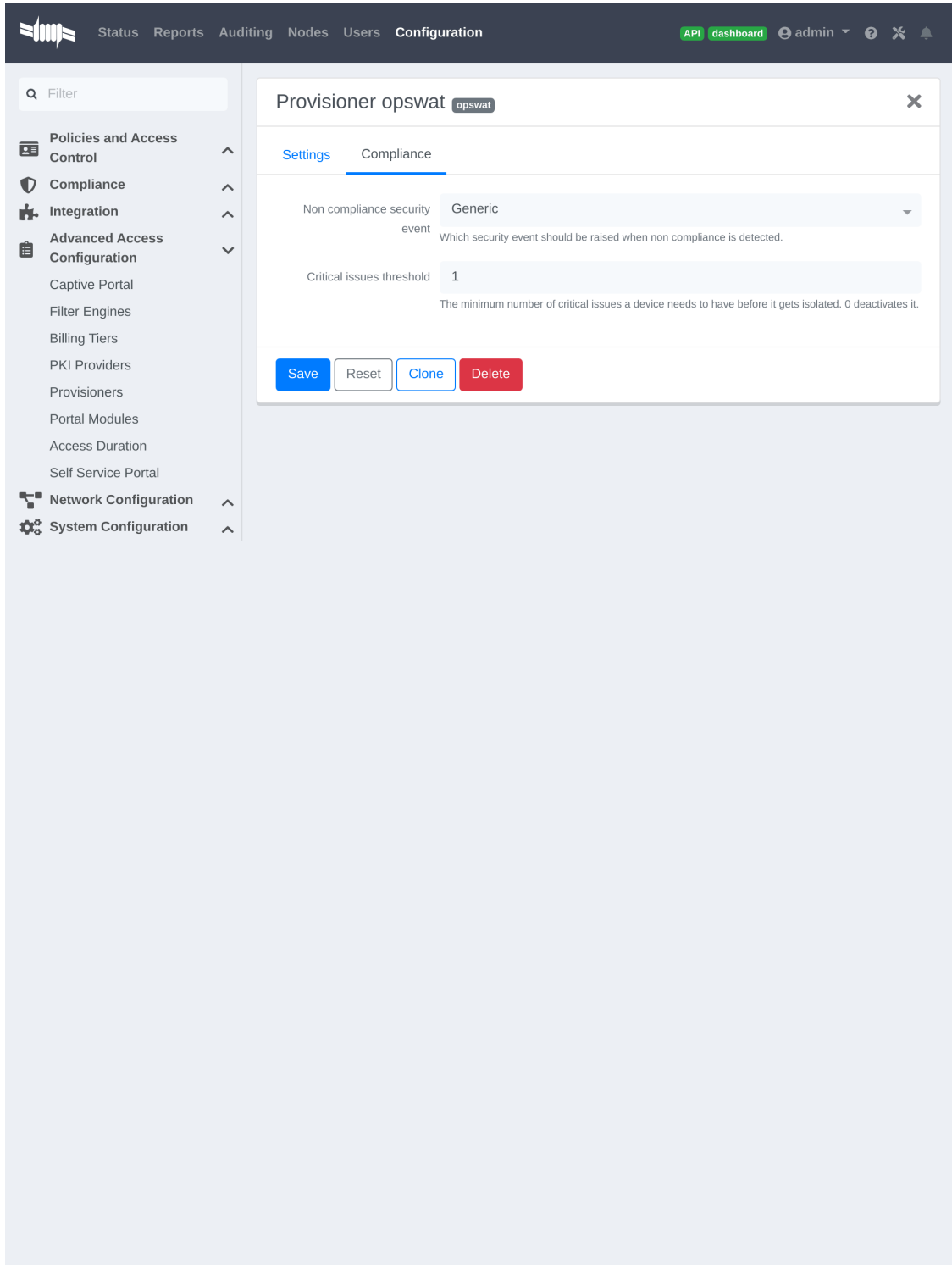
Now in order to enforce the compliance of the devices using the flagged critical issues above, you will need to configure the provisioner you created above to activate the enforcement.

Back in the provisioner configuration, go in the 'Compliance' tab.

You now have to configure the security event that is raised when a device is noncompliant. Using the 'Generic' security event should fit your needs for now, and you can modify the template after.

Then configure the 'Critical issues threshold' and put it at the minimum critical issues a device needs to have before it gets isolated.

Putting 1 into that field will isolate the device whenever there is at least one critical issue with the device.



You can then hit 'Save' and now the device will get isolated whenever it's found as noncompliant.

Customize the template

You can now customize the security event template from the 'Connection Profile' section. Simply select your connection profile and then go in the 'Files' tab.

You can then modify the template `security_events/generic.html` so it displays additional information.

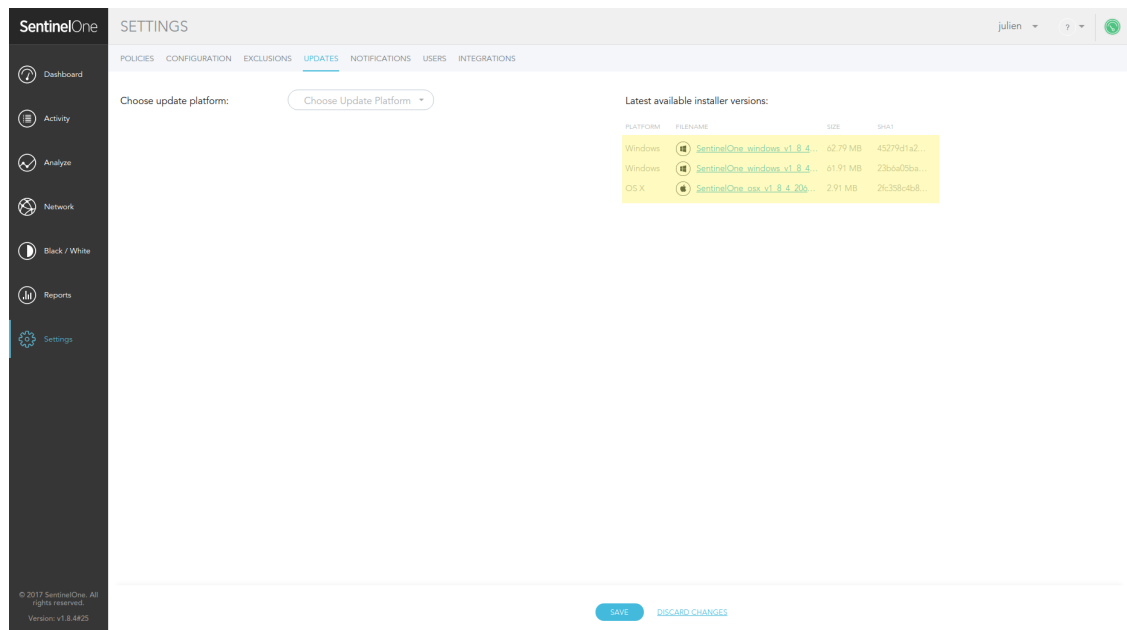
You can also customize this security event from the 'Security Events' section of the administration interface. Refer to the PacketFence Administration Guide for additional information about this.

23.5. SentinelOne

23.5.1. Download the agents

You will first need to download the SentinelOne agents in order to host them on the PacketFence server.

In order to do so, in your SentinelOne management console, go in 'Settings→Updates', then download the Windows and Mac OSX agents on your computer. Once they have been download transfer them on your PacketFence server using SCP. This example will use `/usr/local/pf/html/common/SentinelOne.exe` as the Windows agent path and `/usr/local/pf/html/common/SentinelOne.pkg` as the Mac OSX agent path.



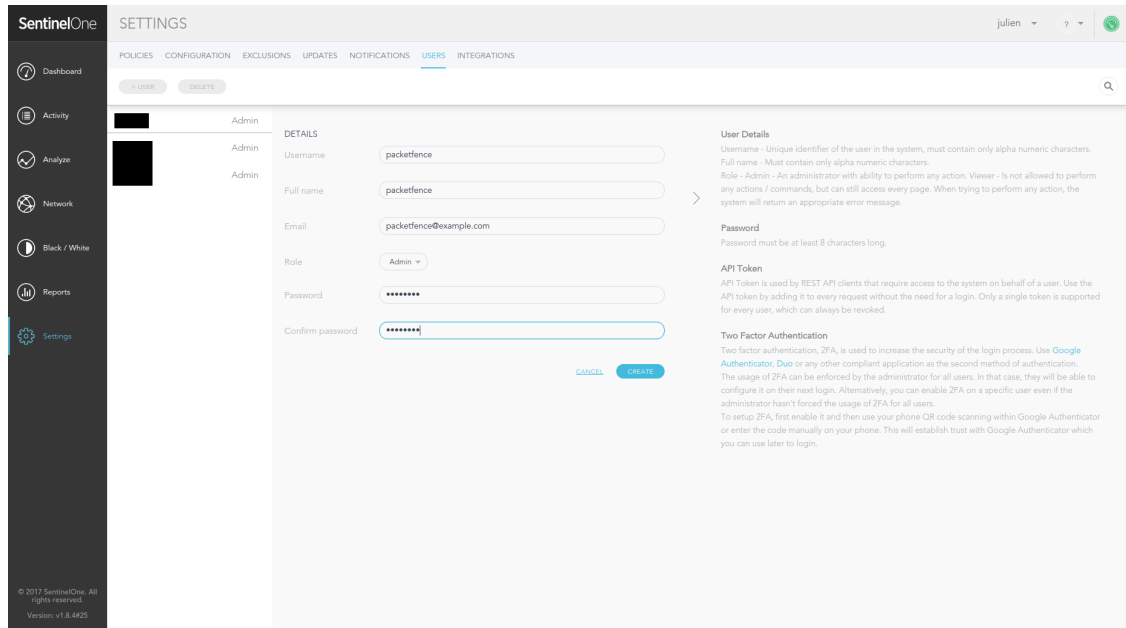
NOTE

All files in `/usr/local/pf/html/common/` are accessible to users that are on the captive portal. Make sure you put the agents file there or in another user-accessible location.

23.5.2. Create an API user

PacketFence will need a user on your SentinelOne instance in order to access the SentinelOne API. To create it, go in 'Settings→Users' and create a new user. Make sure, you note the password

you put here for configuration in PacketFence.



23.5.3. Configure PacketFence

Create a new provisioner

Login in the PacketFence administration interface, then go in the 'Configuration' tab, then in 'Provisioners'. Click 'Add provisioner' then select **SentinelOne**.

The screenshot shows a 'New Provisioner' configuration window. The provisioning ID is 'sentinelone' and the description is 'SentinelOne'. The host is 'packetfence.sentinelone.net' and the port is '443'. The protocol is 'https'. The API username is 'packetfence' and the password is masked with dots. The Windows agent download URI is '/common/SentinelOne.exe' and the Mac OSX agent download URI is '/common/SentinelOne.pkg'. There are 'Create' and 'Reset' buttons at the bottom.

Where:

- 'Provisioning ID' is the user-defined identifier of the provisioner.
- 'Description' is a user friendly description of the provisioner.
- 'Host' is the hostname of your SentinelOne instance.

- 'Port' should be left to default unless your SentinelOne management console is on another port.
- 'API username' is the username of the user you created above in SentinelOne.
- 'API password' is the password of the API user.
- 'Windows agent download URI' is the URI on which the users should download the Windows agent. If you followed the path in this guide, it should be `/common/SentinelOne.exe`.
- 'Mac OSX agent download URI' is the URI on which the users should download the Mac OS agent. If you followed the path in this guide, it should be `/common/SentinelOne.pkg`.

Add the provisioner to the profile

Now that you have created the provisioner, go in the 'Connection Profiles' menu on the left and select the default connection profile. Click 'Add Provisioner' and select the new SentinelOne that was created earlier.

NOTE Make sure you have passthroughs enabled before proceeding further. Instructions on how to enable passthroughs can be found in the 'Passthroughs' section of the Administration Guide.

Once you have completed the configuration, you need to restart pf dns in order for the SentinelOne specific passthroughs to be taken into consideration.

```
# /usr/local/pf/bin/pfcmd service pf dns restart
```

23.5.4. Testing

You can now test that the installation of the SentinelOne client is mandatory after the device registration. Connect a device to your test network and register like you normally would. At the end of the registration process you will be presented a page asking you to install the SentinelOne client on your device. After you install the client click continue. If your access is enabled then this means the connectivity between PacketFence and SentinelOne is good.

PacketFence polls SentinelOne at a regular interval (30 seconds by default) to find devices that have uninstalled their agent. When it detects them as uninstalled, it automatically brings the device back to the portal so the agent is installed.

Everytime your device connects to PacketFence using RADIUS, it schedules a provisioning check to occur 2 minutes after the connection (controlled via security event 1300002). If the agent is inactive on the device or was uninstalled, PacketFence will bring the device back to the portal so the agent is installed again or brought back to an active state.

23.6. Symantec SEPM

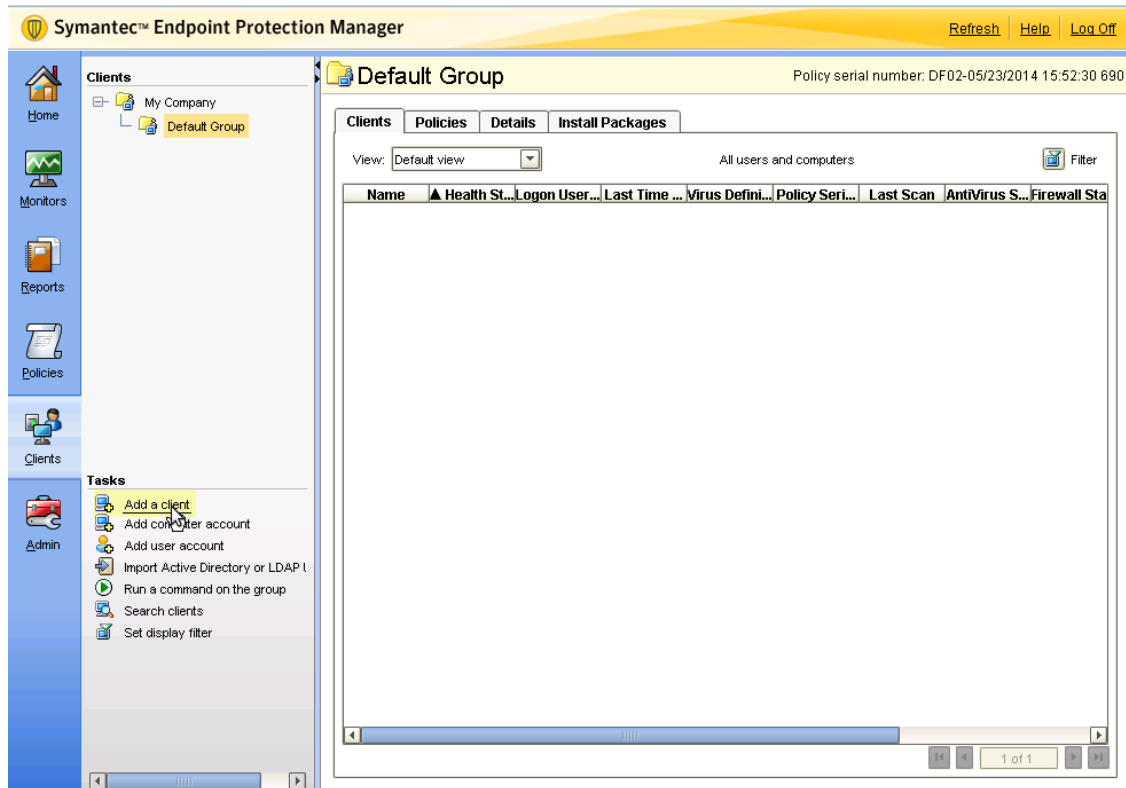
23.6.1. Configure the SEPM

Configure the necessary policies in your SEPM before the creation of the install package. This document does not cover the policy and group configuration. Please refer to Symantec's documentation for more information. This document will use the default policies and the default

group for the package creation.

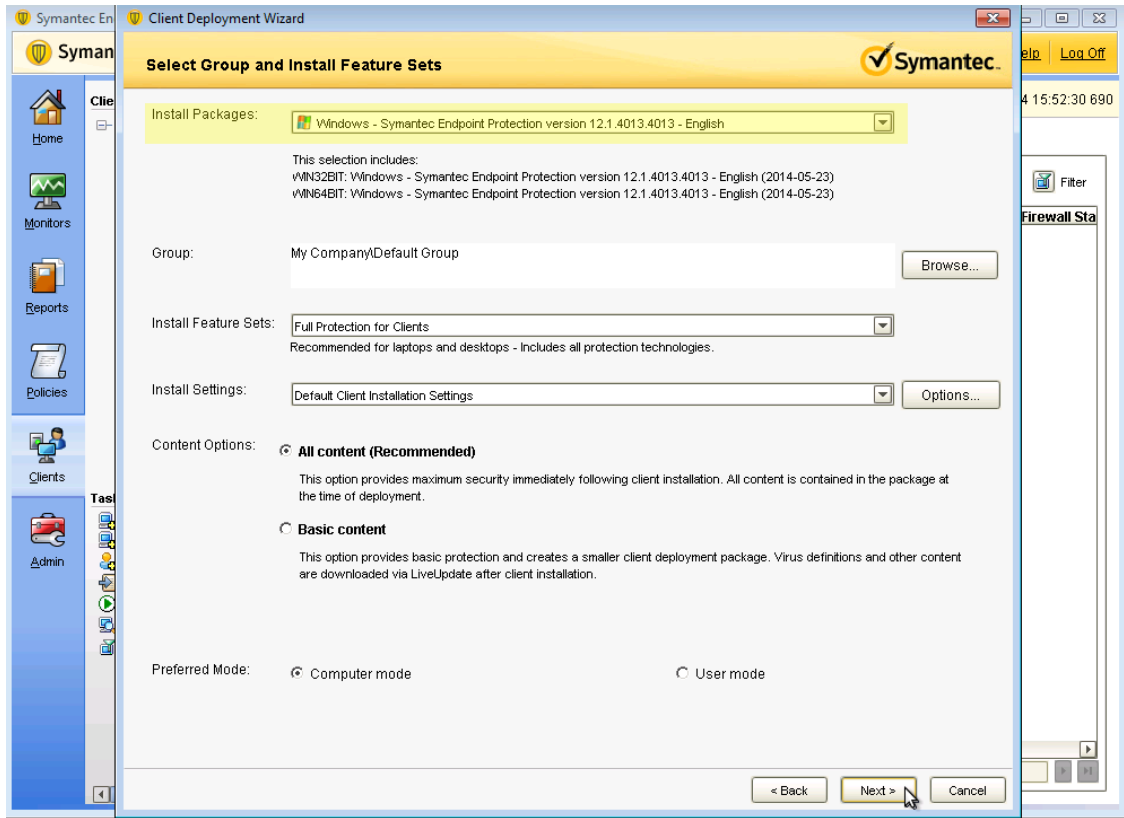
23.6.2. Create the install package

Login in your SEPM console and then go in the 'Clients' tab on the left. Select the group your clients should belong and then click 'Add a client'.



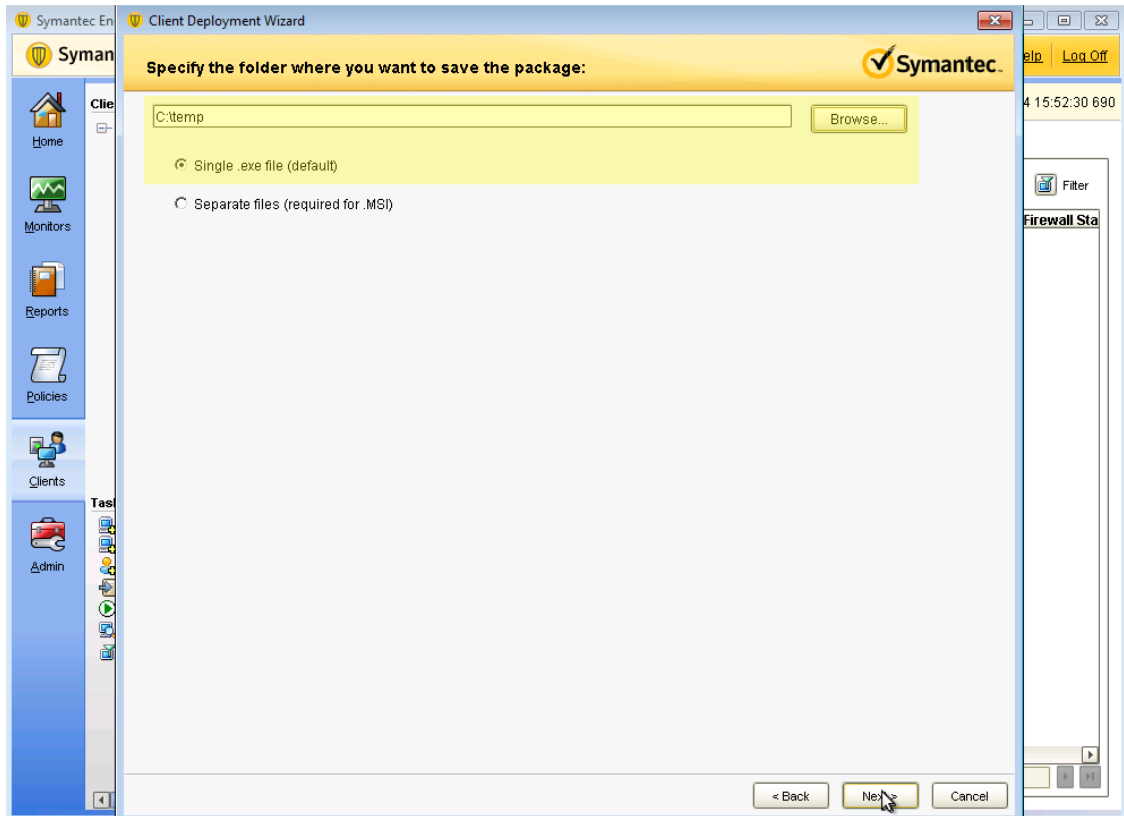
The wizard for the package creation will open. On the first page, make sure 'New Package Deployment' is selected and click 'Next'.

Now on this page, make sure you are creating the package for Windows. Then select the content options you prefer and click 'Next'.



Now on this page, select 'Save Package' and click 'Next'.

Now you will need to select the export location of your new packages. Select any location you prefer. This guide will use `C:\temp\`. Once you are done, click 'Next'.



On the next page, confirm the settings and click 'Next'.

Once the package is created go in the directory where you created the package and navigate your way to the 32 bit package. Then using an SCP or any other method, upload this file to `/usr/local/pf/html/captive-portal/content/sep.exe` on your PacketFence server. Do the same thing for the 64 bit package by uploading it to `/usr/local/pf/html/captive-portal/content/sep64.exe`.

23.6.3. API access

In order to configure the SEPM in PacketFence you will need to generate an OAuth2 access and refresh token so PacketFence can access the SEPM API.

NOTE

The next steps use `192.168.1.100` as the SEPM address. Adapt the URLs to your own SEPM address.

Create an application

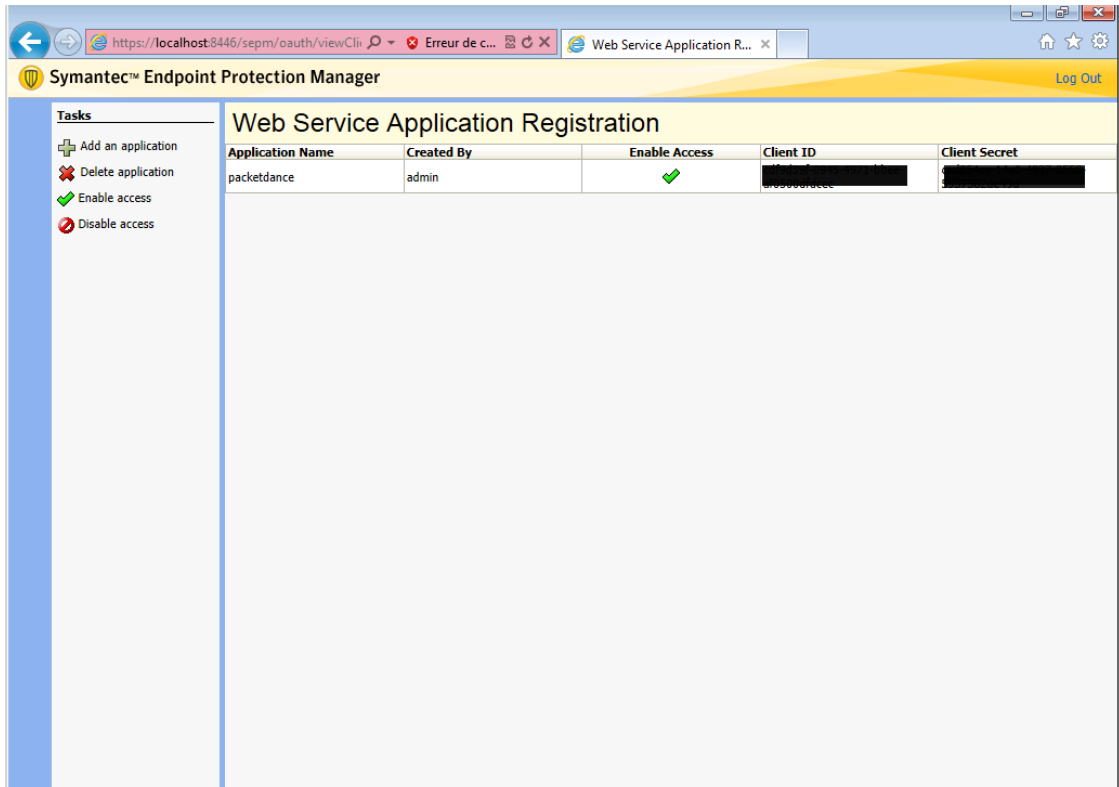
On your computer open a browser and access <https://192.168.1.100:8446/sep.m>.

Accept any certificate error and login with your SEPM credentials.

On the left of the screen, click 'Add an application' and give it a name.

You should now see your application in the list on the right.

Take note of the 'Client ID' and 'Client Secret' of your application



Generate the authorization code

First you will access this page using your browser (replace **-clientid-** by your client ID that you got when creating the application)

```
https://192.168.1.100:8446/sep/.../oauth/authorize?response_type=code&client_id=-clientid-&redirect_uri=http://localhost/
```

Authorize the application and you will then be redirected to an unavailable page but the URL will contain the code in its parameters (ex: <http://127.0.0.1/?code=wJ2RTE>).

Generate the access and refresh token

We will now use the code at the end to generate the access and refresh token using another HTTP request that will be done in your browser. Replace **-clientid-** and **-clientsecret-** by the client id and secret of your application. Then add the code you got above at the end of this URL.

```
https://192.168.1.100:8446/sep/.../oauth/token?grant_type=authorization_code&client_id=-clientid-&client_secret=-clientsecret-&redirect_uri=http://localhost/&code=
```

You should now be presented with a JSON response that contains the access and refresh token. Take note of both of these values for the PacketFence configuration. Example:

```
{ "access_token": "4e3ab3ab-7b1e-4d24-9f5e-c347599a8a72", "token_type": "bearer",  
  "refresh_token": "e03fd915-e9dd-45a6-a05a-e5a1c53c1ccd", "expires_in": 43199 }
```

23.6.4. Configure PacketFence

Create a new provisioner

Login in the PacketFence administration interface, then go in the 'Configuration' tab, then in 'Provisioners'. Click 'Add provisioner' then select sepm.

The screenshot shows a web interface for configuring a new provisioner. The left sidebar contains navigation menus for Policies and Access Control, Compliance, Integration, Advanced Access Configuration, Network Configuration, and System Configuration. The main content area is titled 'New Provisioner' and contains the following fields:

- Provisioning ID: sepm
- Description: Symantec Endpoint Protection
- Roles: (Dropdown menu)
- OS: Type to search. (Dropdown menu)
- Client Key: 1234567890-0987654321-1234567890
- Client Secret: 0987654321-1234567890-0987654321
- Host: 192.168.1.100
- Port: 8446
- Protocol: https (Dropdown menu)
- Access token: 4e3ab3ab-7b1e-4d24-9f5e-c347699
- Refresh token: e03fd915-e9dd-45a6-a05a-e5a1c53
- Agent download URI: http://192.168.1.5/content/sep.exe
- Alt agent download URI: http://192.168.1.5/content/sep64.exe

At the bottom of the form, there are two buttons: 'Create' (in blue) and 'Reset' (in white).

Now configure this new provisioner with the information you got above.

- The Provisioning ID is the friendly name of the provisioner.
- The Client Id is the ID of the application you created in above.
- The Client Secret is the secret of the application you created above.

- The host is the IP address of your SEPM.
- The port and protocol should be left to default.
- The access and refresh token are the tokens you got at the end of step 3.
- The 'Agent download uri' is the HTTP path where we placed the 32 bit package on step 2. In this example it should be <http://packet.fence/content/sep.exe> where packet.fence is the domain name of the registration website of your PacketFence server.
- The 'Alt agent download URI' is the HTTP path where we placed the 64 bit package on step 2. In this example it should be <http://packet.fence/content/sep64.exe> where packet.fence is the domain name of the registration website of your PacketFence server.

Add the provisioner to the profile

Now that you have created the provisioner, go in the 'Connection Profiles' menu on the left and select the default portal. Click 'Add Provisioner' and select the new SEPM provisioner that was created earlier.

[Status](#)
[Reports](#)
[Auditing](#)
[Nodes](#)
[Users](#)
[Configuration](#)

[API dashboard](#)
admin

Filter

- Policies and Access Control**
- Roles
- Domains
- Active Directory Domains
- Realms
- Authentication Sources
- Network Devices
- Switches
- Switch Groups
- Connection Profiles
- Compliance**
- Integration**
- Advanced Access Configuration**
- Network Configuration**
- System Configuration**

Connection Profile default Preview

Settings **Captive Portal** Files

Profile Name: 🔒
A profile id can only contain alphanumeric characters, dashes, period and or underscores.

Profile Description:

Root Portal Module: ▼
The Root Portal Module to use.

Activate preregistration:

This activates preregistration on the connection profile. Meaning, instead of applying the access to the currently connected device, it displays a local account that is created while registering. Note that activating this disables the on-site registration on this connection profile. Also, make sure the sources on the connection profile have "Create local account" enabled.

Automatically register devices:

This activates automatic registration of devices for the profile. Devices will not be shown a captive portal and RADIUS authentication credentials will be used to register the device. This option only makes sense in the context of an 802.1x authentication.

Reuse dot1x credentials:

This option emulates SSO when someone needs to face the captive portal after a successful 802.1x connection. 802.1x credentials are reused on the portal to match an authentication and get the appropriate actions. As a security precaution, this option will only reuse 802.1x credentials if there is an authentication source matching the provided realm. This means, if users use 802.1x credentials with a domain part (username@domain, domain/username), the domain part needs to be configured as a realm under the RADIUS section and an authentication source needs to be configured for that realm. If users do not use 802.1x credentials with a domain part, only the NULL realm will be match if an authentication source is configured for it.

Dot1x recompute role from portal:
When enabled, PacketFence will not use the role initially computed on the portal but will use the dot1x username to recompute the role.

MAC Auth recompute role from portal:
When enabled, PacketFence will not use the role initially computed on the portal but will use an authorized source if defined to recompute the role.

Dot1x unset on unmatched:
When enabled, PacketFence will unset the role of the device if no authentication sources returned one.

Enable DPSK:

This enables the Dynamic PSK feature on this connection profile. It means that the RADIUS server will answer requests with specific attributes like the PSK key to use to connect on the SSID.

Default PSK key:

This is the default PSK key when you enable DPSK on this connection profile. The minimum length is eight characters.

Automatically deregister devices on accounting stop:

This activates automatic deregistration of devices for the profile if PacketFence receives a RADIUS accounting stop.

VLAN pool technique: ▼
The algorithm used to calculate the VLAN in a VLAN pool.

Sources: With no source specified, all internal and external sources will be used.

Billing Tiers: With no billing tiers specified, all billing tiers will be used.

Provisioners: 1 ▼ ⊕

Scanners: With no scan specified, the scan engine will not be triggered.

Self service policy:

Restart PacketFence

In order to enable the boarding passthrough for the device enrollment, you will need to restart the iptables service of PacketFence.

You can do this using the command line by doing `/usr/local/pf/bin/pfcmd service iptables restart` or in the administration interface under 'Status / Services'.

23.6.5. Testing

You can now test that the installation of the Symantec Endpoint Protection client is mandatory after the device registration.

Connect a device to your test network and register like you normally would.

At the end of the registration process you will be presented a page asking you to install the Symantec Endpoint Protection client on your device.

After you install the client click 'Continue'. If your access is enabled than this means the connectivity between PacketFence and the Symantec Endpoint Protection Manager is working.

23.7. Microsoft Intune

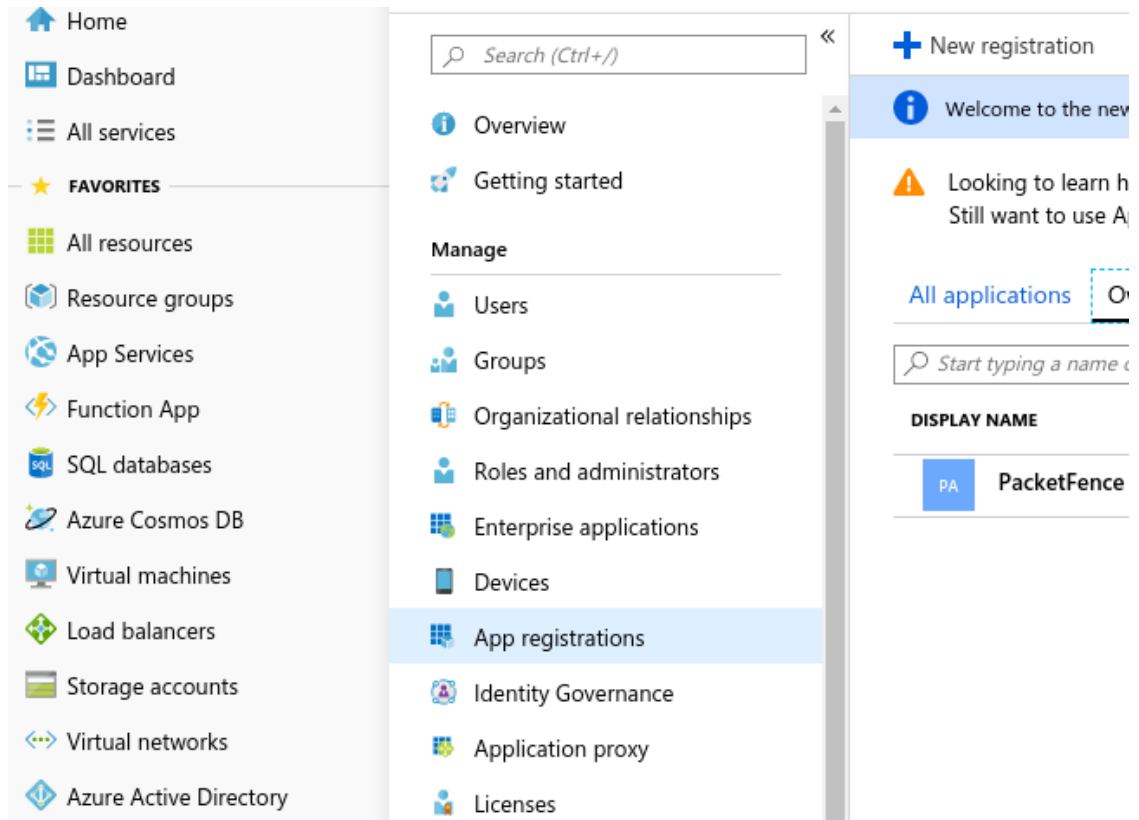
23.7.1. Configure from the Azure portal

You will first need to connect to the Azure portal and be sure that you have the Intune licenses.

Creating the application

Once you are logged in the portal you need to create an application to allow the access to the Graph API.

Click on 'Azure Active Directory' and on 'App registrations' and on 'New registration'



Set a name for the application (in this case PacketFence) and choose as 'Supported account types' : 'Accounts in this organizational directory only' and click 'Register'

Register an application

*** Name**
The user-facing display name for this application (this can be changed later).

PacketFence ✓

Supported account types

Who can use this application or access this API?

Accounts in this organizational directory only (Inverse inc)

Accounts in any organizational directory

Accounts in any organizational directory and personal Microsoft accounts (e.g. Skype, Xbox, Outlook.com)


[Help me choose...](#)

Redirect URI (optional)

We'll return the authentication response to this URI after successfully authenticating the user. Providing this now is optional and it can be changed later, but a value is required for most authentication scenarios.

Web ▼

On the next page you will be able to configure the application, first copy the 'Application (client ID)' and the 'Directory (tenant ID)', you will need them to define your provisioner.


PacketFence

«
🗑 Delete
🌐 Endpoints

- 🏠 Overview
- 👉 Quickstart
- Manage**
- 📄 Branding
- 🔑 Authentication
- 🔑 Certificates & secrets
- 🔑 API permissions

i
Welcome to the new and improved App registrations. Looking

Display name : [PacketFence](#)

Application (client) ID : 724cad4f-4d1c-4970-b405-e4bd6f9475ab

Directory (tenant) ID : 5c21efa5-a2ab-4ce4-96fd-1fad347ebcab

Object ID : 838f146f-f4a9-466e-af0d-71538ab63621

Next click on 'Certificates & secrets' and 'New client secret', this will provide you the password to use for the application (Save it right now because you won't be able to have it after).

Home > Inverse inc - App registrations > PacketFence - Certificates & secrets

PacketFence - Certificates & secrets

Search (Ctrl+/)

- Overview
- Quickstart
- Manage
 - Branding
 - Authentication
 - Certificates & secrets**
 - API permissions
 - Expose an API
 - Owners
 - Roles and administrators (Previ...
 - Manifest
- Support + Troubleshooting
 - Troubleshooting
 - New support request

Copy the new client secret value. You won't be able to retrieve it after you leave this blade.

Credentials enable applications to identify themselves to the authentication service when receiving tokens at a web addressable location (using an HTTPS scheme). For a higher level of assurance, we recommend using a certificate (instead of a client secret) as a credential.

Certificates

Certificates can be used as secrets to prove the application's identity when requesting a token. Also can be referred to as public keys.

[Upload certificate](#)

No certificates have been added for this application.

THUMBPRINT	START DATE	EXPIRES

Client secrets

A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

[New client secret](#)

DESCRIPTION	EXPIRES	VALUE
Password uploaded on Thu Aug 01 2019	12/31/2299	POW0JL7cWxdTpko1dEuC/hMJ853:5+j

The last thing, you need to add permissions on the API, to do that click on 'API permissions' and 'Microsoft Graph' then on the right pane select 'Application permissions' and add:

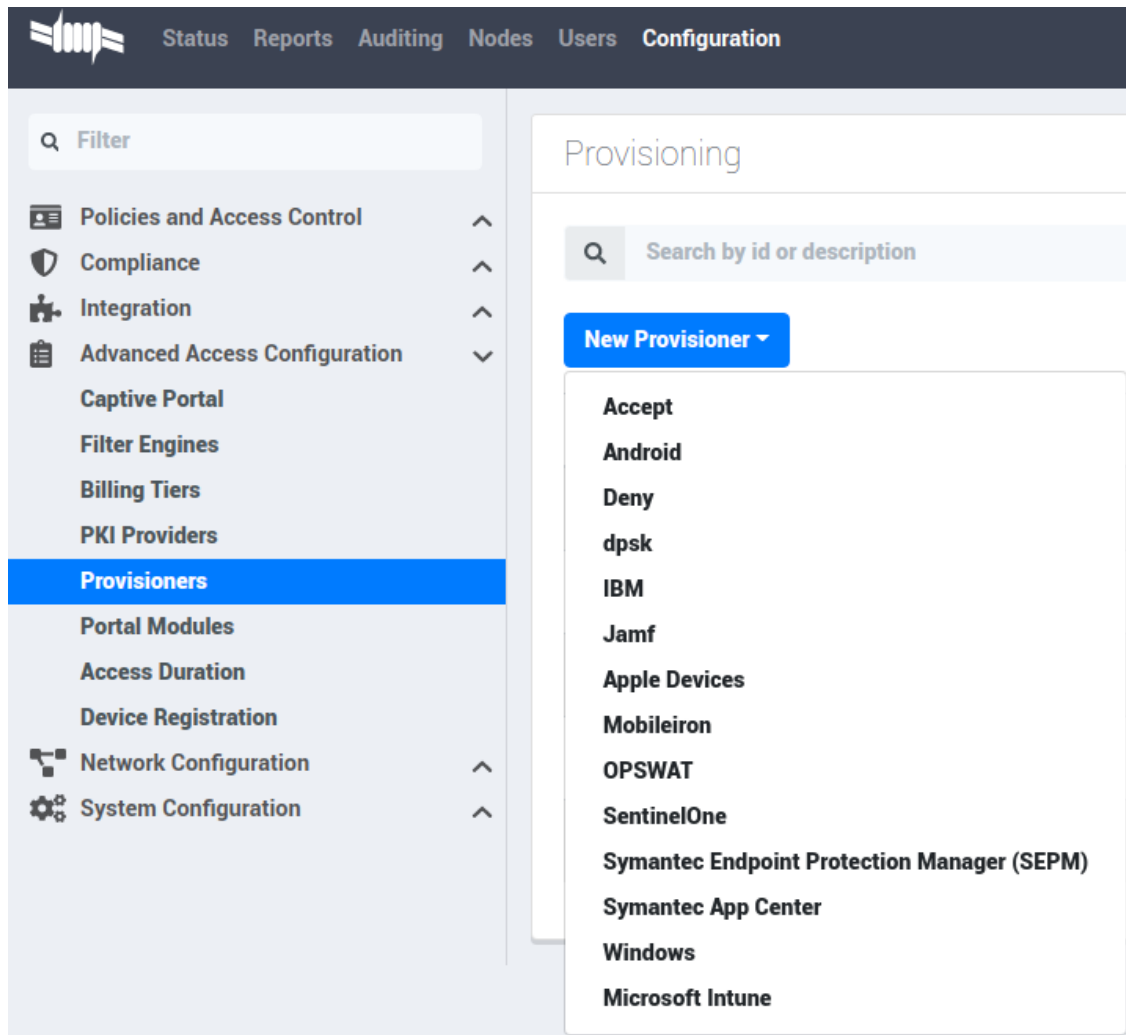
```
Device.ReadWrite.All
DeviceManagementManagedDevices.Read.All
```

And click on 'Grant admin consent for (Name of your app)'

23.7.2. Configure PacketFence

Create a new provisioner

Login in the PacketFence administration interface, then go in the 'Configuration' tab, then in 'Provisioners'. Click 'Add provisioner' then select Microsoft Intune.



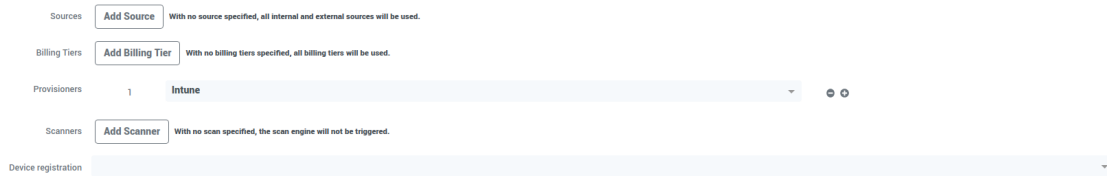
Now configure this new provisioner with the information you got above.

- The Provisioning ID is the friendly name of the provisioner.
- The Application ID is the 'Application (client ID)'.
- The Application Secret is the 'Client secret'.
- The Tenant ID is the 'Directory (tenant ID)'.
- The Client Secret is the secret of the application you created in the developer account.
- The default host should work.
- The default Login URL should work.
- The port and protocol should be left to default.
- The 'Agent download URI' should be ok.
- Authorized domains need to be adapted to allow the device to reach the download URI (per example google play needs multiple domains to be able to install the agent).

Add the provisioner to the profile

Now that you have created the provisioner, go in the 'Connection Profiles' menu on the left and

select the default portal. Click 'Add Provisioner' and select the new Microsoft Intune provisioner that was created earlier.



23.7.3. Testing

You can now test that the installation of the Microsoft Intune client is mandatory after the device registration. Connect a device to your test network and register like you normally would. At the end of the registration process, you will be presented a page asking you to install the Intune client on your device. After you install the client click continue. If your access is enabled then this means the connectivity between PacketFence and Azure is good.

23.8. Google Chromebook Provisioner

23.9. Creating a service account JSON config

In order to communicate with the Google API you must configure a service account, and download JSON security keys, and create a user to impersonate with the proper permissions.

These instructions have been adapted from <https://developers.google.com/identity/protocols/oauth2/service-account>.

- Open the Service accounts page <https://console.developers.google.com/iam-admin/serviceaccounts>.
- If prompted, select a project, or create a new one.
- Click add Create service account.
- Under Service account details, type a name, ID, and description for the service account, then click Create.
- Click on the newly created service account.
- Click on SHOW DOMAIN-WIDE DELEGATION
- Select Enable Google Workspace Domain-wide Delegation
- Save
- Copy the Client ID provided
- Click on Keys > Add Key > Create New Key > Key Type JSON > Create.
- Note where the JSON file is stored.

23.9.1. Delegating domain-wide authority to the service account

- Go to <https://admin.google.com/> click on the Main Menu > Security > API Controls.
- Scroll down to the Domain wide delegation pane, select Manage Domain Wide Delegation.
- Click Add new.

- In the Client ID field, enter the newly created service account's Client ID.
- In the OAuth scopes (comma-delimited) field, enter the the scope <https://www.googleapis.com/auth/admin.directory.device.chromeos.readonly>
- Click Authorize.

23.9.2. Create Role

- Go to <https://admin.google.com/> click on the Main Menu > Account > Admin roles.
- Click 'Create new role'
- Enter Name and Description click 'CONTINUE'
- Search for the Admin console privilege 'Manage Chrome OS Devices (read only)'
- Select 'Manage Chrome OS Devices (read only)' then click 'CONTINUE'
- Click 'CREATE ROLE'

23.9.3. Create a user if needed.

- Go to <https://admin.google.com/> click on the Main Menu > Directory > Users.
- Click 'Create new user'
- Enter First Name, Last Name and Primary email. then click 'ADD NEW USER'.

23.9.4. Assign Role to a user.

- Go to <https://admin.google.com/> click on the Main Menu > Directory > Users.
- Select user for service account
- Click 'Admin roles and privileges'
- Assign the Role previously created.

23.10. Configure PacketFence

23.10.1. Create a new provisioner

Login in the PacketFence administration interface, then go to 'Configuration' > 'Advanced Access Configuration' > 'Provisioners' > 'New provisioner' > 'Google Workspace Chromebook'.

New Provisioner Google Workspace Chromebook ✕

Provisioning ID ⓘ
ID required.

Description

Enforce Disabled
Whether or not the provisioner should be enforced. This will trigger checks to validate the device is compliant with the provisioner during RADIUS authentication and on the captive portal.

Auto register Disabled
Whether or not devices should be automatically registered on the network if they are authorized in the provisioner.

Apply role Disabled
When enabled, this will apply the configured role to the endpoint if it is authorized in the provisioner.

Role to apply ▼
When 'Apply role' is enabled, this defines the role to apply when the device is authorized with the provisioner.

Roles ▼
Nodes with the selected roles will be affected.

OS ▼
Nodes with the selected OS will be affected.

Non compliance security event ▼
Which security event should be raised when non compliance is detected.

User
User the service account will impersonate.

Service Account JSON data 📎
Service Account JSON data.

Now configure this new provisioner with the information you got above.

- The Provisioning ID is the friendly name of the provisioner.
- User for service account to impersonate.
- The JSON security keys for Service Account created.

24. PKI Integration

24.1. Microsoft PKI

This section has been created to give a quick start to configure the Microsoft PKI with PacketFence. This guide does not include advanced troubleshooting of EAP-TLS connections. Refer to the relevant documentation of EAP-TLS, RADIUS and OpenSSL for advanced features.

24.1.1. Assumptions

- You have at least one server with PacketFence 5.4 or later.
- The server already has a properly configured switch or access point with 802.1X support.
- The PacketFence RADIUS server is working in your environment.
- You have a Microsoft Windows 2008 R2 Enterprise server installed.
- The PacketFence management IP will be 192.168.1.5.
- The RADIUS shared secret is "useStrongerSecret".
- In this guide you will see a lot of use of <ServerDNSName>, most of the MSPKI services requires in their configuration to use the FQDN of the server and not his IP.

24.1.2. Installation

Install Active Directory Certificate Service (ADCS)

NOTE

This section will cover the configuration for Active Directory Certificate Services (ADCS) on Microsoft Windows 2008 R2 Enterprise. The installation of ADCS is not covered by this guide, refer to the Microsoft documentation about it for more information (<http://social.technet.microsoft.com/wiki/contents/articles/9063.network-device-enrollment-service-ndes-in-active-directory-certificate-services-ad-cs.aspx>).

For the integration with PacketFence, the following subroles need to be installed in ADCS:

- Certification Authority Web Enrollment
- Network Device Enrollment Service
- Online Responder

Before you start the configuration, a hotfix is necessary due to a Microsoft issue. After restarting the ADCS service, the server cannot enroll new certificates and display the following error message: "The RPC Server is unavailable". The hotfix is available here: <https://support.microsoft.com/en-us/kb/2633200>

Communication between the MSPKI and PacketFence will be using port 80.

Configuring Network Device Enrollment Service (NDES)

For the deployment of ADCS you will need to configure Network Device Enrollment Service (NDES). This subrole will allow us to exchange certificates with the MSPKI server via Simple Certificate Exchange Protocol (SCEP).

Every configuration change has to be done by an account with administrative privileges.

Challenge Password

Microsoft SCEP (MSCEP) includes by default a challenge password, which is unique and dynamically generated for each device which wants to enroll. In a BYOD deployment, this can be a barrier as a user cannot register a device by himself without the intervention of an administrator. Since we use NDES with PacketFence, our security to obtain a certificate would be the credentials necessary to access the enrollment system.

To disable the challenge password you need to modify the following key in the Windows registry.

Click **Start** and enter **regedit**.

Navigate to
Computer HKEY_LOCAL_MACHINE SOFTWARE Microsoft Cryptography MSCEP EnforcePassword.

Change the value of **EnforcePassword** to **0** (default is **1**).

Extend URL length for the request

Best practices recommends to extend the URL length to avoid issue with longer request.

To do so, enter the following command in the CLI on the NDES server:

```
%systemroot%\system32\inetsrv\appcmd.exe set config /section:system.webServer/  
security/requestFiltering /requestLimits.maxQueryString:"16384" /commit:apphost
```

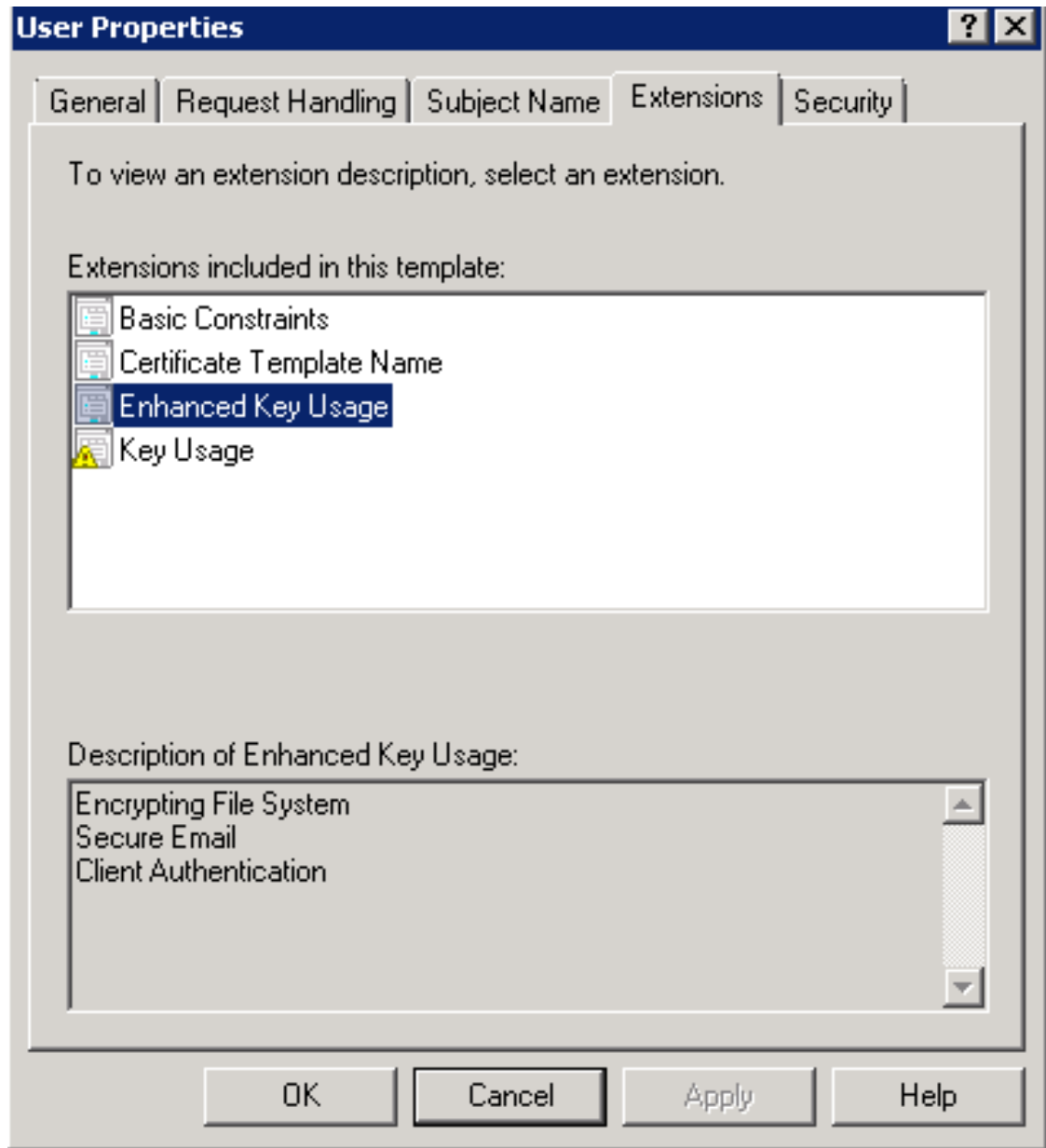
Certificate Template

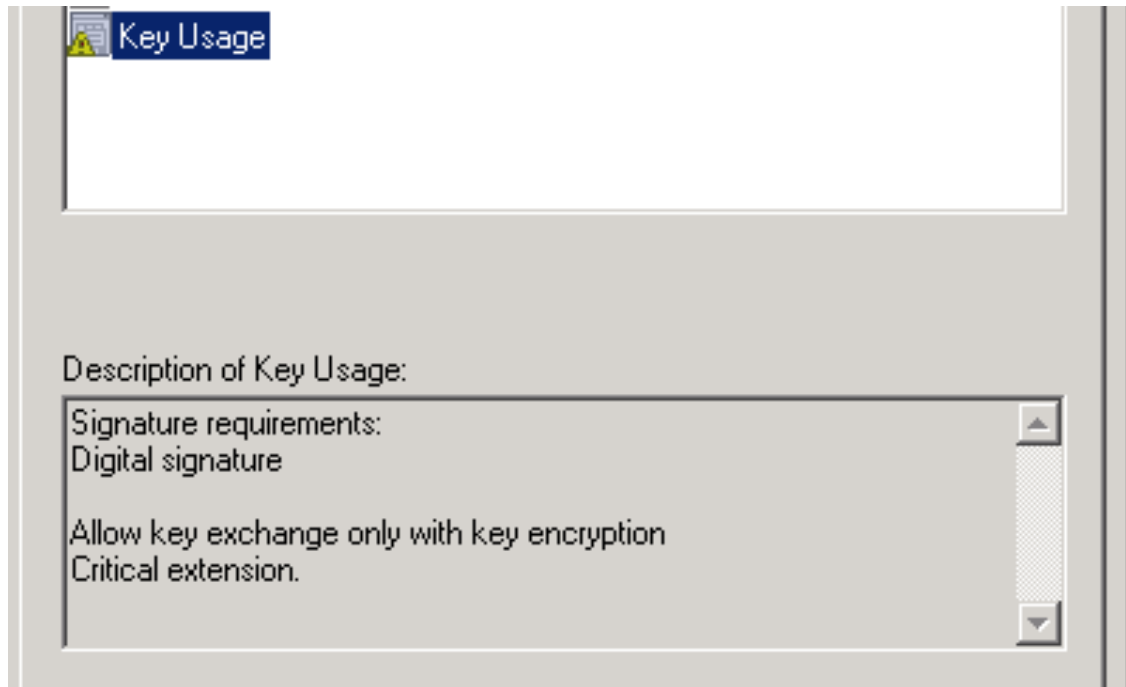
CAUTION

Remember that the validity of your CA can impact your whole certificate architecture.

The goal is to deliver certificates for **user Authentication**, this means you will need to setup a specific template.

First, the certificate template needs at least the following **Enhanced Key Usage** and **Key Usage**:





The next step is to duplicate a template where those **Key Usage**, and **Enhanced Key Usage** are already configured. We advise to duplicate the template **User** and change the necessary settings.

To duplicate the template, you need to navigate through **Server Manager Roles Active Directory Certificates Services Certificate templates**. Now right click the template **User** and select **Duplicate this template**.

Once duplicated, right click your new template, go to **Properties**. Navigate to the tab **Subject Name**. Make sure to select **Supplied in the request** over **Built from information in Active Directory**, otherwise the requested CN will be overwritten by NDES.

To allow NDES to use this template you need to navigate to **Server Manager Roles Active Directory Certificates Services**, expand **<ServerDNSName>**, right click **Certificate template** and choose **New template to issue**, in the list select your newly created template.

Now that you choose the template to deliver you need to configure it in the registry.

To access the registry editor, press **Start** and type **regedit**.

While in the registry navigate to **Computer HKEY_LOCAL_MACHINE SOFTWARE Microsoft Cryptography MSCEP**.

You should have a list of three keys entries:

- EncryptionTemplate,
- GeneralPurposeTemplate,
- SignatureTemplate.

The default value should be **IPSECIntermediateOffline**. Replace each value with your newly created template name.

At this point, you need to reboot the NDES server to apply changes to the registry.

IIS configuration

The use of SCEP with PacketFence also require a change in the IIS configuration.

Navigate to **Server Manager Web(IIS)**, expand **Default web site** then select **CertSrv mscep**. Select **Authentication**, and double click **Anonymous Authentication**. Make sure that **Application pool identity** is selected.

Online Certificate Status Protocol (OCSP)

For the configuration of OCSP, the following changes are necessary.

First we need to allow the use of the template **OCSPResponseSigning** by the server, to do so navigate to **Server Manager Roles Active Directory Certificates Services**, expand **<ServerDNSName>**, right click **Certificate template** and choose **New template to issue**, in the list select **OCSPResponseSigning**.

After the installation of OCSP we need to create a Revocation Configuration.

To create the Revocation Configuration navigate to **Server Manager Roles Active Directory Certificate Services** and expand **OnlineResponder: <ServerDNSName>**. Right click **Revocation Configuration**, select **Add Revocation Configuration**, click **Next**, choose a name for your configuration and click **Next**.

Choose **Select a certificate for an existing enterprise CA**, click **Next**. Click **Browse** and find your enterprise CA in the list, select it, click **OK** and then **Next**. Choose **Automatically select a signing certificate**, make sure **Auto-Enroll for an OCSP signing certificate** is selected, then choose the default template of OCSP which is **OCSPResponseSigning** in the dropdown list next to **Certificate Template:**. You need to add providers only if you wish to use a CRL in addition to OCSP.

Once created, right click the revocation configuration and select **Edit properties**, go to the **Signing** tab, then select **Enable NONCE extension support** then click **OK**.

Make sure that your OCSP server appears in the CA settings. Right click your CA, choose **Properties**. Navigate to the tab **Extension**, in the dropdown list **Select extension** choose **Authority Information Access (AIA)**. Make sure that you have the following in the list of locations: <http://<ServerDNSName>/OCSP>.

If you do not have it, add it via the button **Add...** In this menu type the <http://> then insert **<ServerDNSName>** and type **/OCSP**, validate by clicking **OK**. Also verify that **Include in the online certificate status protocol(OCSP) extension** is selected.

By default OCSP has a two days delay to refresh it's CRL information. Which means if you revoke a certificate on MSPKI, it will take two days before PacketFence detects the certificate is revoked. If this delay is too long for your needs, you can change it on the NDES server. To do so, navigate to **Server Manager Roles Active Directory Certificate Service** and right click **Enterprise PKI**, in the menu select **Options...** The delay can be changed by modifying the value of **Set CRL status to Expiring when expiring in:** to your convenience.

RADIUS Certificate Generation

Using the Microsoft PKI involves that all your certificates will be delivered by the root CA of the MSPKI.

As for RADIUS authentication you will need to generate a certificate for PacketFence.

To generate the RADIUS certificate, the template **WebServer** will be used.

The next step is to create the request (CSR), a private key from the PacketFence server and submit the CSR to the NDES server. Connect to PacketFence via SSH and type the following in the CLI to generate the CSR and sign it with the private key:

```
openssl req -new -newkey rsa:2048 -nodes -keyout server.key -out server.csr
```

You will be prompted for some information, here is an example of a valid configuration.

- CN=packetfence.local
- C=CA
- ST=QC
- Locality=Montreal
- Organization=Inverse
- Organization Unit=IT

No fields are mandatory other than the CN.

Once you have your CSR you will submit it to the NDES server.

To submit the request you need to copy the content of the request (CSR) on the MSPKI enrollment website. The URL to input the request will be: <http://<ServerDNSName>/CertSrv/>.

When reaching the website, click **Request a certificate**, select **advanced certificate request**. Paste the content of your CSR file and select the template **Web Server**. Click **Submit**. On this page select **Base 64 encoded** and click **Download certificate**.

This will give you the certificate (public key) for PacketFence.

Now download the CA file by reaching the following URL in your browser: <http://<ServerDNSName>/CertSrv/>.

Click **Download a CA certificate, certificate chain or CRL**, select your CA certificate in the list, select **Base 64** as the encoding method and finally click **Download CA certificate**.

Copy those files to PacketFence.

24.1.3. Configuring PacketFence

Certificate Storage on PacketFence

It is recommended to create a separate directory to separate EAP-TLS certificates from server certificates:

```
# mkdir /usr/local/pf/conf/ssl/tls_certs/
```

RADIUS EAP-TLS authentication requires three files, the CA certificate, the server certificate and the server private key.

Copy those files in your newly created folder:

- Private Key of the RADIUS server (obtained while generating the CSR)
- Certificate for RADIUS (obtained from the submitted CSR)
- CA Certificate (downloaded from the NDES website)

Ensure that the files are readable by the user `pf`:

```
# chown pf:pf /usr/local/pf/conf/ssl/tls_certs/*
```

RADIUS EAP-TLS and MSPKI

In order to use the certificates generated by the MSPKI, edit the radius EAP configuration file.

Edit `/usr/local/pf/conf/radiusd/eap.conf` and replace the following lines with references to your new certificates in the `tls` configuration block:

```
private_key_file = [% install_dir %]/conf/ssl/server.key
certificate_file = [% install_dir %]/conf/ssl/server.pem
```

E.g.

```
private_key_file = [% install_dir %]/conf/ssl/tls_certs/server.key
certificate_file = [% install_dir %]/conf/ssl/tls_certs/server.pem
ca_file = [% install_dir %]/conf/ssl/tls_certs/MyCA.pem
```

Certificate revocation checks have to be configured in the `OCSP` sub-block of `tls`.

For example:

```
ocsp {
    enable = yes
    override_cert_url = yes
    url = "http://<MSPKI ServerDNSName or IP>/ocsp"
}
```

Restart `radiusd` to regenerate the new configuration files and enable EAP-TLS using your CA signed certificates:

```
# /usr/local/pf/bin/pfcmd service radiusd restart
```

PacketFence PKI Provider Configuration

Using the PKI requires configuring the PKI providers section in the PacketFence GUI under *Configuration*→*Advanced Access Configuration*→*PKI Providers*. The provider configuration defines

how PacketFence connects to the MSPKI and what information will be sent.

Add a new PKI provider and select SCEP.

Fill out the form for a PKI provider according to your Certificate of Authority configuration.

For the URL it will be <http://<ServerDNSName>/CertSrv/mscep/>.

WARNING | Don't use **https:** scheme.

You do not need any Username/Password combination for this configuration.

The screenshot shows a web interface for configuring a new PKI provider. The top navigation bar includes 'Status', 'Reports', 'Auditing', 'Nodes', 'Users', and 'Configuration'. The user is logged in as 'admin'. The left sidebar lists various configuration categories, with 'Advanced Access Configuration' expanded. The main form is titled 'New PKI Provider' and contains the following fields:

- PKI Provider Name:** MSPKI
- URL:** http://MyPKIServer.example.com/ (The url used to connect to the SCEP PKI service.)
- Username:** (Username to connect to the SCEP PKI Service.)
- Password:** (Password for the username filled in above.)
- Country:** Canada (Country for the certificate.)
- State:** QC (State for the certificate.)
- Locality:** (Locality for the certificate.)
- Organization:** Inverse (Organization for the certificate.)
- Organizational unit:** IT (Organizational unit for the certificate.)
- Common Name Attribute:** Username (Defines what attribute of the node to use as the common name during the certificate generation.)
- Common Name Format:** %s (Defines how the common name will be formatted. %s will expand to the defined Common Name Attribute value.)
- CA cert path:** /usr/local/pf/conf/ssl/tls_certs/MyCa (Path of the CA certificate used to generate client certificate/key combination.)
- Server cert path:** /usr/local/pf/conf/ssl/tls_certs/MyCert (Path of the RADIUS server authentication certificate.)

At the bottom of the form, there are two buttons: 'Create' and 'Reset'.

The "Server cert path" and "CA cert path" both need to be absolute (e.g. `/usr/local/pf/conf/ssl/tls_certs/MyCA.pem` is an absolute path).

The "Common name attribute" field defines how the certificate will be generated and what type of "ownership" will associate the certificate to the connection. If you select 'MAC address', a certificate will be generated using the MAC address as the identifier. If you select 'Username', a

certificate will be generated using his login name on the authentication backend.

Provisioners Configuration

Provisioners allow devices to automatically configure themselves to connect to the proper SSID (if applicable), use the proper authentication method (e.g. EAP-TLS) and trust the CA certificate and any certificate signed by it.

Provisioners are configured in the PacketFence administration GUI under *Configuration*→*Advanced Access Configuration*→*Provisioners*.

Add a new provisioner for each of the classes of devices to be supported amongst Android, Apple Devices and Windows. Fill out the form, choosing a different Provisioning Id per provisioner.

- Roles: The "Roles" field defines which devices will be affected by the provisioning item. If empty, all devices for this class will be affected.
- SSID: The "SSID" field defines which SSID will be configured on the device using the authentication profile.
- EAP-Type: The EAP type defines the authentication method supported and should be set to EAP-TLS to integrate with the PacketFence PKI.
- Security type: The security type should be set to WPA2 to integrate with the PacketFence PKI.
- PKI Provider: This should match the provider you configured earlier in the PKI provider section.

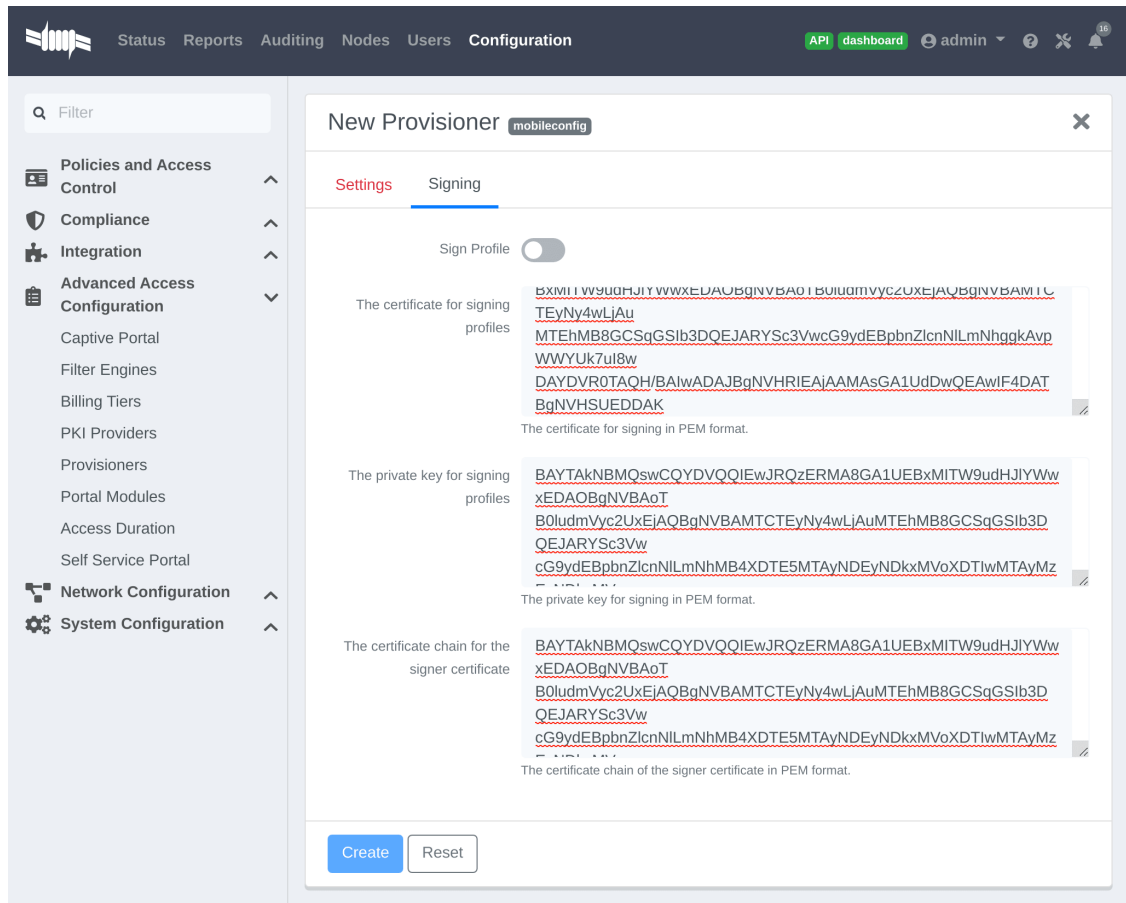
The following is an example on how to configure an EAP-TLS connection for Windows/Android/Mac OS X/iOS

The screenshot shows a web-based configuration interface for a network management system. The top navigation bar includes 'Status', 'Reports', 'Auditing', 'Nodes', 'Users', and 'Configuration'. The user is logged in as 'admin'. The left sidebar contains a search filter and a menu with categories: 'Policies and Access Control', 'Compliance', 'Integration', 'Advanced Access Configuration', 'Captive Portal', 'Filter Engines', 'Billing Tiers', 'PKI Providers', 'Provisioners', 'Portal Modules', 'Access Duration', 'Self Service Portal', 'Network Configuration', and 'System Configuration'. The main content area is titled 'New Provisioner' and contains the following fields:

- Provisioning ID: EAPTLS
- Description: Windows EAP-TLS
- Roles: default (with a close icon)
- SSID: PF-Secure
- Broadcast network: (with a note: 'Uncheck this box if you are using a hidden SSID.')
- Security type: WPA2 (with a note: 'Select the type of security applied for your SSID.')
- EAP type: EAP-TLS (with a note: 'Select the EAP type of your SSID. Leave empty for no EAP.')
- PKI Provider: MS-SCEP

At the bottom of the form are 'Create' and 'Reset' buttons.

Mac OS X/iOS require the provisioning profile to be signed if you want to remove the **untrusted** warning when installing the profile. You need to sign it with a Certification Authority already trusted by the device such as e.g. VeriSign. Configuring this has to be done in the 'Signing' tab in the "Apple devices".



Fill out the fields with the contents of the Base64 encoded certificates. To extract this information from a pem formatted certificate, copy the file content.

Certificate file example:

```
----- BEGIN CERTIFICATE -----
1234567890asdfghjkl
zxcvbnmqwertyuiop78
----- END CERTIFICATE -----
```

Copy everything from the BEGIN to END lines. Repeat this operation for the certificate key and intermediate certificate.

```
----- BEGIN PRIVATE KEY -----
1234567890asdfghjkl
zxcvbnmqwertyuiop78
----- END PRIVATE KEY -----
```

Connection Profiles Configuration

Provisioners have to be enabled on the Connection Profiles configuration in the PacketFence

GUI.

Under *Configuration*→*Policies and Access control*→*Connection Profiles*, select each of the provisioners created above which should be active for the profile. If no connection profile is defined, configure the "default" profile to use the provisioners created.

Passthroughs Required for Android

Android devices require passthroughs to be created to allow them to fetch the configuration application from the Google Play Store.

IMPORTANT

Passthroughs will vary depending on the location where your Google account was created. You will need to add some extra passthroughs for the store of your country. In the section debug there is a how-to determine which address you need to add.

Add the following to the "Fencing" section of the Configuration tab in the PacketFence GUI.

```
passthrough=enabled
passthroughs=*.ggpht.com,*.googleusercontent.com,android.clients.google.com,
*.googleapis.com,*.android.clients.google.com,*.gvt1.com
```

Debugging MSPKI Integration with PacketFence

This is a way to do the procedure of enrollment manually, mainly for debugging purposes.

First you need to generate a request and its private key via the openssl command. Type following commands in PacketFence CLI:

```
mkdir temp; cd temp
openssl req -newkey rsa:2048 -nodes -keyout local.key -out local.csr -subj
'/C=CA/ST=QC/L=Montreal/O=Inverse/OU=IT/CN=www.test.example.com'
```

This will create 2 files in your current directory, `local.csr` and `local.key`.

Now you need to obtain the CA and some specific certificates from the MSPKI.

```
sscep getca -u http://<ServerDNSName>/CertSrv/mscep/ -c MyCA.crt
```

Now you need to use the "CEP encryption" certificate and the "Enrollment agent". Both were obtained when doing the `sscep getca`. You should have at least three certificates with the same name and a different number at the end. e.g. `MyCA.crt-0` (Enrollment agent certificate), `MyCA.crt-1` (CEP encryption certificate) and `MyCA.crt-2` (CA certificate).

To display the content of each certificate use following commands:

```
openssl x509 -in MyCA.crt-0 -text
openssl x509 -in MyCA.crt-1 -text
openssl x509 -in MyCA.crt-2 -text
```

In the output search for **X509v3 extensions:**. When using the **sscep enroll** command you will need the "Enrollment agent" certificate as an argument for **-c** and the "CEP Encryption" certificate as an argument for **-e**. **-d** is use for the debug output. **-l** is the local file where your certificate will be save.

```
sscep enroll -c MyCA.crt-0 -e MyCA.crt-1 -k local.key -r local.csr \
-l MyCert.crt -S sha1 -u http://<ServerDNSName>/CertSrv/mscep/ -d
```

To verify your certificate against the OCSP you can use the following **openssl** command:

```
openssl OCSP -issuer path/CA-Certificate -cert path/Certificate-to-verify \
-text -url http://<ServerDNSName>/OCSP
```

Debugging Android Passthroughs

If you need to add domains to passthroughs, we advise you to capture the traffic coming from the device which cannot access the Google Play Store. To do this you can use tcpdump for instance, collect the IP address of the device then run the following in PacketFence CLI:

```
tcpdump -i $REGISTRATION_INTERFACE -n dst port 53 and src host @IP_Device
```

This will output any DNS requests from the device to PacketFence. You will need to find **google** related domain and add them to your passthroughs list.

24.2. PacketFence PKI

This section has been created to give a quick start to configure the PacketFence PKI in PacketFence. This guide does not include advanced troubleshooting of EAP-TLS connections. Refer to the relevant documentation of EAP-TLS, RADIUS and OpenSSL for advanced features. The PKI comes installed by default since PacketFence version 10. All certificates would be saved in the database. If you want to migrate your certificate from the old PacketFence PKI please see the upgrade section.

24.2.1. Certificate Authority creation

You will need to create a new certificate authority. Go to the PacketFence web administration under the section Configuration → Integration → PKI → Certificate Authorities and click on **New Certificate Authority**

The screenshot shows a web interface for PKI configuration. The top navigation bar includes 'Status', 'Reports', 'Auditing', 'Nodes', 'Users', and 'Configuration'. The user is logged in as 'admin'. The left sidebar contains a navigation menu with categories like 'Policies and Access Control', 'Compliance', 'Integration', and 'PKI'. The main content area is titled 'PKI' and has tabs for 'Certificate Authorities', 'Templates', 'Certificates', and 'Revoked Certificates'. The 'Certificate Authorities' tab is active, showing a search bar with the text 'Search by identifier, common name, email or organisation'. Below the search bar, there is a 'New Certificate Authority' button, a dropdown menu set to 'pfpci', and a pagination control showing '25' items per page. The table below has columns for 'Identifier', 'Common Name', 'Email', and 'Organisation'. The table is currently empty, displaying a message: 'No certificate authorities found. Please refine your search.'

Here's a CA example:

The screenshot shows the 'New Certificate Authority' configuration page in the Inverse firewall management interface. The left sidebar contains navigation options such as Policies and Access Control, Compliance, Integration, Firewall SSO, Cisco Mobility Services Engine, Web Services, Switch Templates, Syslog Parsers, Syslog Forwarding, WRIX, PKI (Certificate Authorities, Templates, Certificates, Revoked Certificates), Advanced Access Configuration, Network Configuration, and System Configuration. The main form contains the following fields:

- Common Name: Inverse_Root_CA
- Email: administrator@inverse.ca
- Organisation: Inverse Inc.
- Country: Canada
- State or Province: Quebec
- Locality: Montreal
- Street Address: Park Avenue
- Postal Code: H3N 1X1
- Key type: KEY_RSA
- Key size: 4096
- Digest: SHA256WithRSA
- Key usage: (dropdown menu)
- Extended key usage: (dropdown menu)
- Days: 750

At the bottom of the form, there are 'Create' and 'Reset' buttons. The 'Create' button is highlighted in blue.

Once you have created the CA, you should see the Root CA certificate displayed at the bottom of the page:

Status Reports Auditing Nodes Users Configuration
API dashboard
admin

Filter

- Policies and Access Control**
- Compliance**
- Integration**
 - Firewall SSO
 - Cisco Mobility Services Engine
 - Web Services
 - Switch Templates
 - Syslog Parsers
 - Syslog Forwarding
 - WRIX
- PKI
 - Certificate Authorities
 - Templates
 - Certificates
 - Revoked Certificates
- Advanced Access Configuration**
- Network Configuration**
- System Configuration**

Certificate Authority

Identifier	2	🔒
Common Name	Inverse_Root_CA	🔒
Email	administrator@inverse.ca	🔒
Organisation	Inverse Inc.	🔒
Country	Canada	🔒
State or Province	Quebec	🔒
Locality	Montreal	🔒
Street Address	Park Avenue	🔒
Postal Code	H3N 1X1	🔒
Key type	KEY_RSA	🔒
Key size	4096	🔒
Digest	SHA256WithRSA	🔒
Key usage		🔒
	<small>Optional. One or many of: digitalSignature, nonRepudiation, keyEncipherment, dataEncipherment, keyAgreement, keyCertSign, cRLSign, encipherOnly, decipherOnly.</small>	
Extended key usage		🔒
	<small>Optional. One or many of: serverAuth, clientAuth, codeSigning, emailProtection, timeStamping, msCodeInd, msCodeCom, msCTLSign, msSGC, msEFS, nsSGC.</small>	
Days	750	🔒
	<small>Number of days the CA will be valid.</small>	
Certificate	<pre> -----BEGIN CERTIFICATE----- MIIGHDCCBASgAwIBAgIBATANBgkqhkiG9w0BAQsFADCBijELMAKGA1UEB hMCQ0Ex DzANBgNVBAGTBIF1ZWIyZERMAM0G9w0BAQsFADCBijELMAKGA1UEB hMCQ0Ex cm9uZG9uZm9udGVudGVudGVudGVudGVudGVudGVudGVudGVudGVudGV nZlcnNlElu Yy4xNDUwMDUwMDUwMDUwMDUwMDUwMDUwMDUwMDUwMDUwMDUwMDUwMD E5NDZaFw0y MjAzMjg0NDUwMDUwMDUwMDUwMDUwMDUwMDUwMDUwMDUwMDUwMDUwMD GUXVlYmVjMREw DwYDQGEwhN250cmVhbDEUMBIGA1UECRMLUGFyayBBdmVudWUx EDA0BgNVBBET B0gzTlAxWDExFTATBgNVBAoTDEludmVyc2UgSW5lLjEYMBYGA1UEAwP SW52Z2ZjZj ZV9Sb290X0NBMIICjANBgkqhkiG9w0BAQFAOCAG8AMIICGkCAGEAu 0IEIG5 j2For+UtPoackMkkKhQRjMbrzjf/hUeIje8/h16en7SNyzTrzHXlB5pttNomRo b P8KWHNy7hqlcbOc9YOKd2ilgEcrOl/hdSuf992cT8djMXU+hDZ6ygidg WJPs3 M5qwuVML/RJBXC4jlx2rXk13GFRIVW7UAFevquRtlH9i9DQ9oxhDGno4 FJ6Uw Mf9PdN36In9YdmXHyOkjJISJz7DwVFT3zCV7Nr4DIZohLLbRdPC0z3H Bvd1Oo xslh2uV2gN/htLzEDF/wADGaF4xsSROkQ+QH3FV7j8rV6g9BhUfDLYKPL P1JQJ v06MGwa1SCRw2P28TweZqL0qLqAqXu5cROHqVYnby2wUbgdx386ijuked c9NVqXJ yevC+Gj3Sc/nmoFqrZgk6o05Plx4p+O8phwL9lhbQm+DuC+xYFFWIMsMK FoH2O6 rwvyTXMFYHsPGTDBEBtirtPiebLXGRWOCPhUU4/65Nax9srNhatOCOE wrQQtuu 2CgK7lIXRP15D201nY96kMgFyTZF8bQgzjON9lvNkoOr0dVrLHpt3Bj+F e2yz Faal8gKtt3OzarMeXoJE9FKzfzS6QkYkLkLz0s4YcYBPXDR/nS5NQkpW </pre>	

[Clone](#)

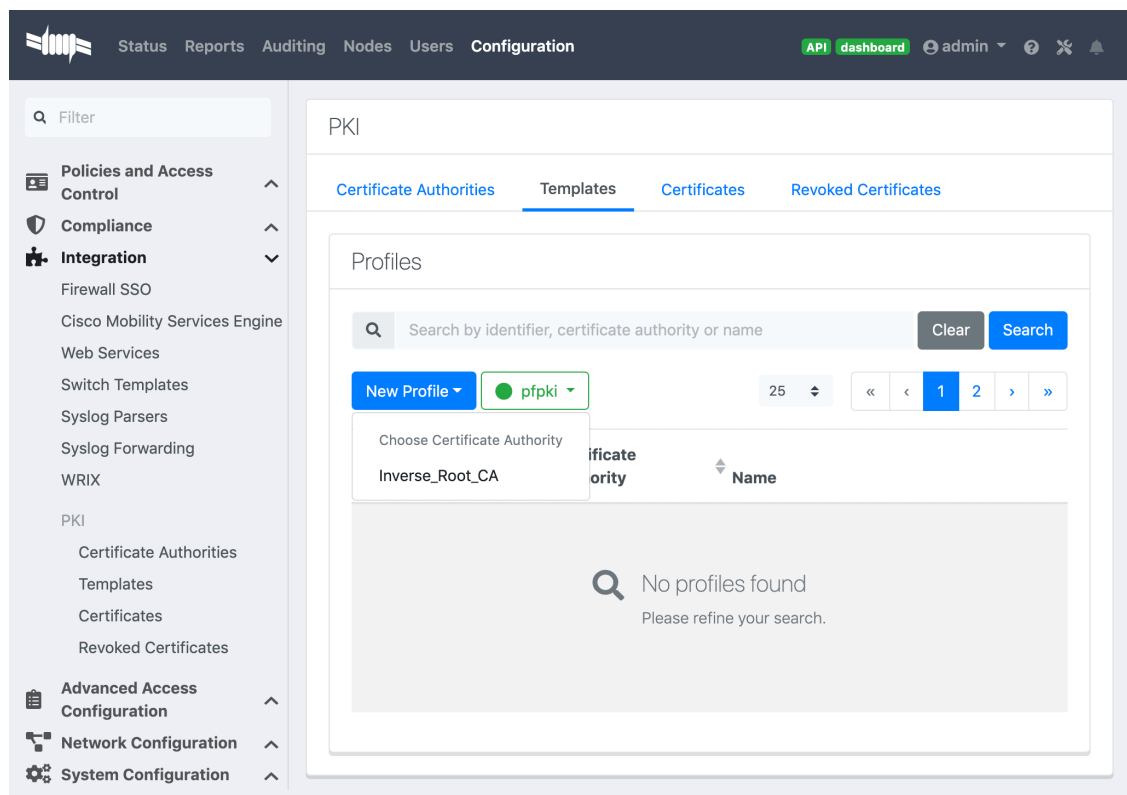
Once done copy the certificate in the clipboard from the Certificate Authorities list (Configuration → Integration → PKI → Certificate Authorities and click on **Copy Certificate**) then edit the RADIUS certificate section in Configuration → System Configuration → SSL Certificates → RADIUS → Edit and paste the public key in "Certificate Authority" and Save. (Don't forget to restart radiusd-auth)

This will authorize the EAP TLS authentications using the PKI issued certificates.

24.2.2. Template creation

Now you will need to create a certificate template that will gather all the settings for your certificate like the validity period or the certificate usage.

Select the Certificate Authority previously created:



Here's a template example:

The screenshot shows a web interface for configuring a new PKCS 12 profile. The main content area is titled "New Profile" and has two tabs: "General" (selected) and "PKCS 12". The configuration fields are as follows:

- Certificate Authority:** Inverse_Root_CA
- Name:** User_Certificate (Profile Name)
- Validity:** 365 (Number of days the certificate will be valid)
- Key type:** KEY_RSA
- Key size:** 2048
- Digest:** SHA256WithRSA
- Key usage:** DigitalSignature
- Extended key usage:** ServerAuth, ClientAuth

At the bottom of the form are "Save" and "Reset" buttons. The sidebar on the left contains navigation options: Policies and Access Control, Compliance, Integration (with sub-items like Firewall SSO, Cisco Mobility Services Engine, Web Services, Switch Templates, Syslog Parsers, Syslog Forwarding, WRIX), PKI (with sub-items like Certificate Authorities, Profiles, Certificates, Revoked Certificates), Advanced Access Configuration, Network Configuration, and System Configuration.

Key usage clientAuth: To use your certificate for a client authentication.

Key usage serverAuth: If you want to install your certificate on a server.

P12 mail password emailed to the users:

The screenshot shows a web-based configuration interface. At the top, there is a navigation bar with 'Configuration' selected. Below it is a sidebar with a search filter and several menu items: Policies and Access Control, Compliance, Integration, Firewall SSO, Cisco Mobility Services Engine, Web Services, Switch Templates, Syslog Parsers, Syslog Forwarding, WRIX, PKI (with sub-items: Certificate Authorities, Templates, Certificates, Revoked Certificates), Advanced Access Configuration, Network Configuration, and System Configuration. The main content area is titled 'Template' and has a 'PKCS 12' tab. It contains the following fields:

- P12 mail password:** A toggle switch is turned on. Below it, the text reads 'Email the password of the pkcs12 file.'
- P12 mail subject:** An empty text input field. Below it, the text reads 'Email subject.'
- P12 mail from:** An empty text input field. Below it, the text reads 'Sender email address.'
- P12 mail header:** A large empty text area. Below it, the text reads 'Email header.'
- P12 mail footer:** A large empty text area. Below it, the text reads 'Email footer.'

At the bottom of the configuration area, there are three buttons: 'Save' (highlighted in blue), 'Reset', and 'Clone'.

SCEP

You can choose to enable SCEP on this template.

General **PKCS 12** **SCEP**

Enable SCEP On
Enable SCEP for this template.

SCEP challenge password
SCEP challenge password.

SCEP days before renewal
Number of days before SCEP authorize renewal

Enable Cloud Integration Off
Enable Cloud integration for this template.

Cloud Service
Cloud Service to integrate.

Save **Clone** **Reset** **Cancel**

SCEP Test

Let's do a scep request by hand. Directly from the PacketFence server do that:

Create a private key and a csr file:

```
openssl req -newkey rsa:2048 -keyout PRIVATEKEY.key -out MYCSR.csr
```

```

Generating a 2048 bit RSA private key
.....
.....
.....+++
.....+++
writing new private key to 'PRIVATEKEY.key'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----

You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----

Country Name (2 letter code) [XX]:CA
State or Province Name (full name) []:QC
Locality Name (eg, city) [Default City]:Montreal
Organization Name (eg, company) [Default Company Ltd]:Acme
Organizational Unit Name (eg, section) []:IT
Common Name (eg, your name or your server's hostname) []:acme.com
Email Address []:admin@acme.com

```

```

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:password
An optional company name []:

```

Get the CA certificate:

```

sscep getca -u http://ip_address/scep/template_name -c ./ca-prefix -i MyPKI -v
-d

```

```
sscep: starting sscep, version 0.6.1
sscep: new transaction
sscep: transaction id: SSCEP transactionId
sscep: hostname: ip_address
sscep: directory: scep/template_name
sscep: port: 80
sscep: SCEP_OPERATION_GETCA
sscep: requesting CA certificate
sscep: scep msg: GET /scep/template_name?operation=GetCACert&message=MyPKI
HTTP/1.0
```

```
sscep: server returned status code 200
sscep: MIME header: application/x-x509-ca-cert
sscep: valid response from server
sscep: MD5 fingerprint: 22:DE:09:17:8B:5F:94:1E:EB:0D:9C:12:EF:05:F0:C5
sscep: CA certificate written as ./ca-prefix
```

Remove the private key passphrase:

```
openssl rsa -in PRIVATEKEY.key -out private.key
Enter pass phrase for PRIVATEKEY.key:
writing RSA key
```

Send the CSR and retrieve the certificate:

```
scep enroll -c ./ca-prefix -k ./private.key -r ./MYCSR.csr -u
http://ip_address/scep/template_name -S sha1 -l ./cert.crt
```

24.2.3. Certificate creation

Navigation: Status Reports Auditing Nodes Users **Configuration** API dashboard admin

Filter

- Policies and Access Control
- Compliance
- Integration
 - Firewall SSO
 - Cisco Mobility Services Engine
 - Web Services
 - Switch Templates
 - Syslog Parsers
 - Syslog Forwarding
 - WRIX
 - PKI
 - Certificate Authorities
 - Templates
 - Certificates
 - Revoked Certificates
- Advanced Access Configuration
- Network Configuration
- System Configuration

PKI

Certificate Authorities Templates **Certificates** Revoked Certificates

Certificates

Search by identifier, certificate authority, profile, common name or email Clear Search

New Certificate pfpci 25 << < 1 2 > >>

Choose Certificate Authority - Profile

Inverse_Root_CA - User_Certificate

Profile	Common Name	Email	Valid Until
<p>🔍 No certificates found</p> <p>Please refine your search.</p>			

Navigation: Status Reports Auditing Nodes Users **Configuration** API dashboard admin

Filter

- Policies and Access Control
- Compliance
- Integration
 - Firewall SSO
 - Cisco Mobility Services Engine
 - Web Services
 - Switch Templates
 - Syslog Parsers
 - Syslog Forwarding
 - WRIX
 - PKI
 - Certificate Authorities
 - Templates
 - Certificates
 - Revoked Certificates
- Advanced Access Configuration
- Network Configuration
- System Configuration

PKI

New Certificate ×

Certificate Template Inverse_Root_CA - User_Certificate
Certificate profile used for this certificate.

Common Name Test_User_1
Username for this certificate.

Email test-user@inverse.ca
Email address of the user. The email with the certificate will be sent to this address.

Organisation Inverse

Country Canada

State or Province Quebec

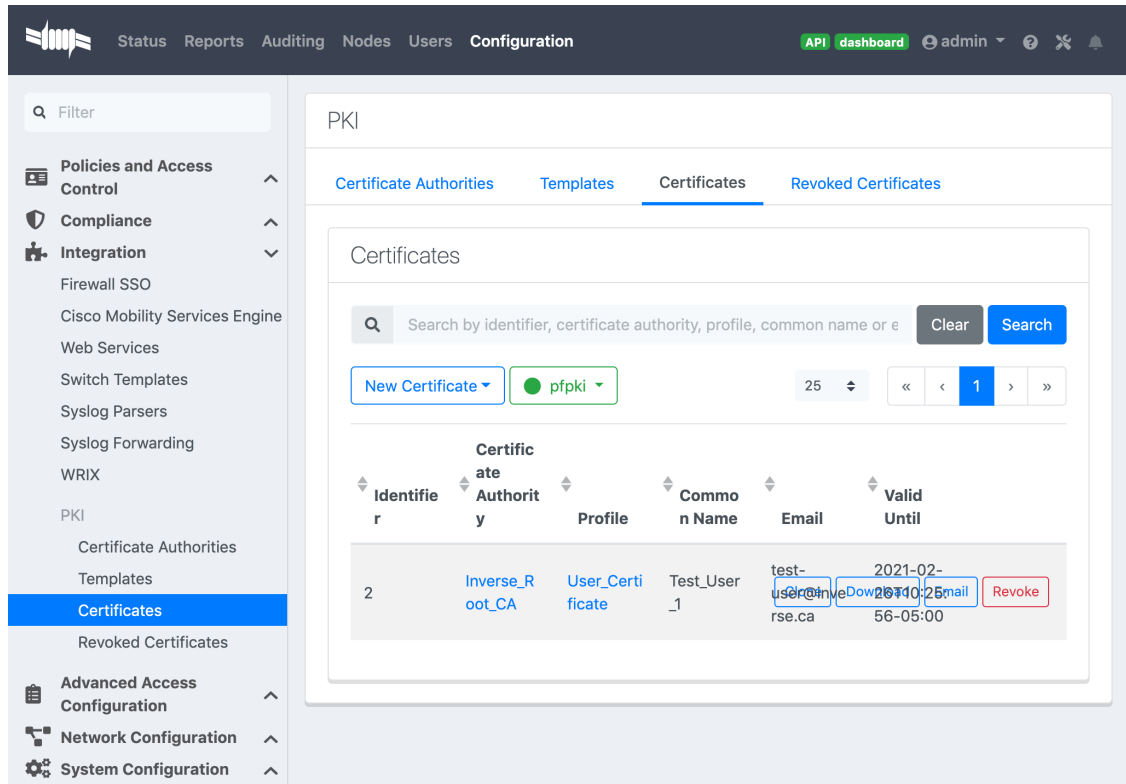
Locality Montreal

Street Address Park Avenue

Postal Code H3N 1X1

Create Reset

Once it's created, you can send it to the email user or download the p12 format:



24.2.4. PEM format

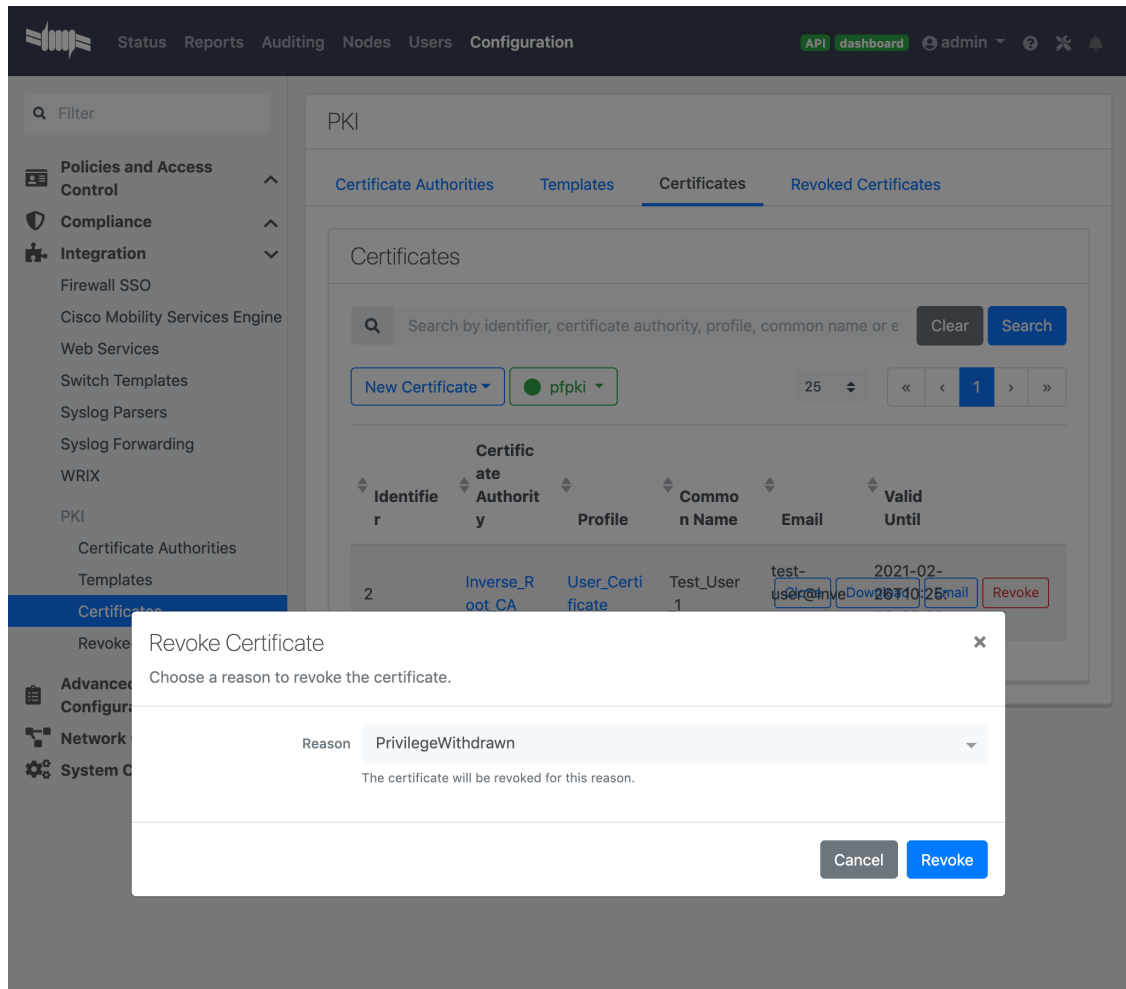
The PacketFence PKI hand out PKCS12 certificates, if you want to convert your certificate to PEM format, you can use the commands:

```
openssl pkcs12 -in YourCert.p12 -nocerts -out YourCert.key -nodes
openssl pkcs12 -in YourCert.p12 -out YourCert.pem -clcerts -nokeys
```

24.2.5. Revoke a certificate

If you revoke a certificate it can't be recovered and you would need to recreate a new one. You will need to specify a reason of the revocation.

Click on the Revoke button on the certificate:



24.2.6. PKI Provider

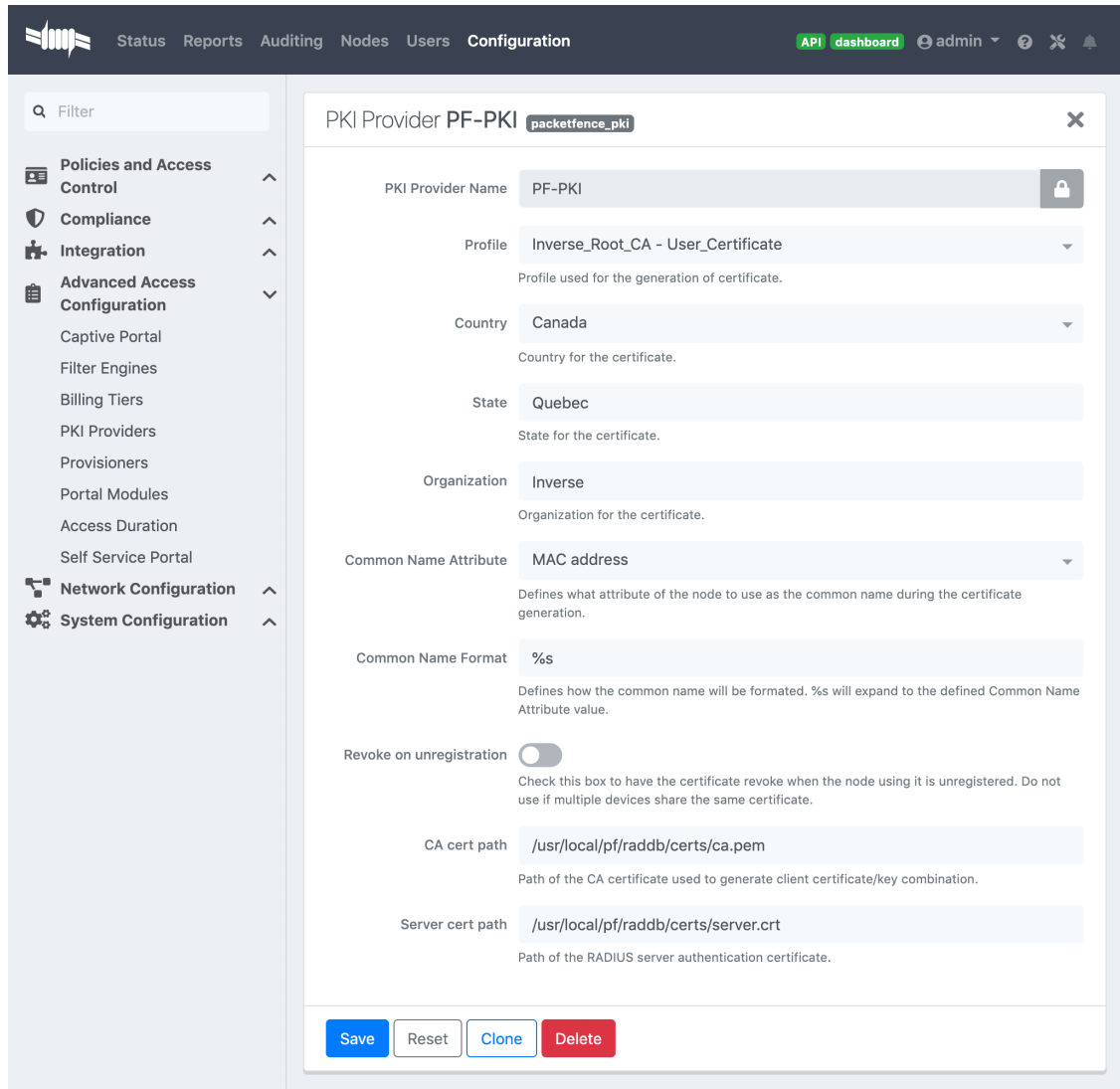
You can hand out certificate to non-BYOD device on a captive portal.

First, you would need to create the PKI provider that will query the PacketFence PKI for new certificate. Go to Configuration → Advanced Access Configuration → PKI provider

The screenshot shows the PacketFence configuration interface. The top navigation bar includes 'Status', 'Reports', 'Auditing', 'Nodes', 'Users', and 'Configuration'. The user is logged in as 'admin'. The left sidebar contains a navigation menu with categories like 'Policies and Access Control', 'Compliance', 'Integration', 'Advanced Access Configuration', 'Network Configuration', and 'System Configuration'. The 'PKI Providers' section is selected and highlighted in blue. The main content area displays a table of PKI Providers. A search bar at the top of the table allows searching by name or description. A 'New PKI Provider' button is visible. The table has columns for 'Name', 'Description', and 'Type'. One entry, 'Packetfence PKI', is circled in red. Below the table, there are 'Delete' and 'Clone' buttons for the selected entry.

Name	Description	Type
Packetfence Local		
Packetfence PKI		packetfence_pki

Create a certificate per user or per device mac address, this example will cover one certificate per device:



24.2.7. Intune Integration

Azure configuration

You can hand out certificates when you use intune enrolment.

First you need to create an application on Azure that allow PacketFence to connect to the Intune API.

To do that first you have to go in Azure portal and App registration then click **New registration**

Home > inverse inc

inverse inc | App registrations

Azure Active Directory

- Overview
- Getting started
- Preview features
- Diagnose and solve problems

Manage

- Users
- Groups
- External Identities
- Roles and administrators
- Administrative units
- Enterprise applications
- Devices
- App registrations**

+ New registration

Try out the new App reg

Starting June 30th, 2020

All applications **Owned**

Start typing a name or /

Display name

PA PacketFence

Next set a Name and in "Supported account types" select "Accounts in this organizational directory only" then click **Register**

Home > inverse inc >

Register an application

* Name

The user-facing display name for this application (this can be changed later).

PacketFence ✓

Supported account types

Who can use this application or access this API?

- Accounts in this organizational directory only (inverse inc only - Single tenant)
- Accounts in any organizational directory (Any Azure AD directory - Multitenant)
- Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)
- Personal Microsoft accounts only

[Help me choose...](#)

Redirect URI (optional)

We'll return the authentication response to this URI after successfully authenticating the user. Providing this now is optional and it can be changed later, but a value is required for most authentication scenarios.

Web e.g. https://example.com/auth

On the next page you have to copy the "Application (Client) ID" and the "Directory (tenant) ID", those will be needed to configure PacketFence.

Microsoft Azure

Home > inverse inc >

PacketFence

Search (Ctrl+I)

Delete Endpoints Preview features

Overview

Quickstart

Integration assistant

Manage

Branding

Authentication

Essentials

Display name : PacketFence

Application (client) ID : 36e3f54a-e38f-4e37-9470-91ba9d3c0158

Object ID : 20cd24e1-a989-47ad-ab45-3911464317b4

Directory (tenant) ID : 045285b8-2ed4-4c86-9763-e4a3564d5d55

Supported account types : My organization only

Client credentials : Add a certificate or secret

Redirect URIs : Add a Redirect URI

Application ID URI : Add an Application ID URI

Managed application in L... : PacketFence

Then you need to generate a "Client secrets", to do that click on "Add a certificate or secret"

Microsoft Azure | Search resources, services, and docs (G+)

Home > inverse inc > PacketFence

PacketFence | Certificates & secrets

Search (Ctrl+/) << Got feedback?

- Overview
- Quickstart
- Integration assistant
- Manage
 - Branding
 - Authentication
 - Certificates & secrets**
 - Token configuration
 - API permissions
 - Expose an API
 - App roles
 - Owners
 - Roles and administrators | Preview
 - Manifest
- Support + Troubleshooting
 - Troubleshooting
 - New support request

Credentials enable confidential applications to identify themselves to the authentication service when receiving tokens at a web addressable location (using an HTTPS scheme). For a higher level of assurance, we recommend using a certificate (instead of a client secret) as a credential.

Certificates

Certificates can be used as secrets to prove the application's identity when requesting a token. Also can be referred to as public keys.

Upload certificate

Thumbprint	Start date	Expires	Certificate ID
No certificates have been added for this application.			

Client secrets

A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

+ New client secret

Description	Expires	Value	Secret ID
No client secrets have been created for this application.			

Add a client secret

Description: PacketFence_SCEP

Expires: Recommended: 6 months

Copy the "Value" of the secret, this is the only time you should be able to see it.

Microsoft Azure | Search resources, services, and docs (G+)

Home > inverse inc > PacketFence

PacketFence | Certificates & secrets

Search (Ctrl+/) << Got feedback?

- Overview
- Quickstart
- Integration assistant
- Manage
 - Branding
 - Authentication
 - Certificates & secrets**
 - Token configuration
 - API permissions
 - Expose an API
 - App roles
 - Owners
 - Roles and administrators | Preview
 - Manifest
- Support + Troubleshooting
 - Troubleshooting
 - New support request

Credentials enable confidential applications to identify themselves to the authentication service when receiving tokens at a web addressable location (using an HTTPS scheme). For a higher level of assurance, we recommend using a certificate (instead of a client secret) as a credential.

Certificates

Certificates can be used as secrets to prove the application's identity when requesting a token. Also can be referred to as public keys.

Upload certificate

Thumbprint	Start date	Expires	Certificate ID
No certificates have been added for this application.			

Client secrets

A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

+ New client secret

Description	Expires	Value	Secret ID
PacketFence_SCEP	11/13/2021	_d87e2AcS1514233k.VNLxSua40--v-N96Y	48b92945-d0e8-4095-bd0e-da0bca1fe32

Next you have to add API permissions, click on "API permissions" → "Add a Permissions":

Intune -> "Application permissions" and select "scep_challenge_provider"
ActiveDirectory Graph -> "Application permissions" and select
"Application.Read.All", "Application.ReadWrite.All",
"Application.ReadWrite.OwnedBy"
ActiveDirectory Graph -> "Delegated permissions" and select "User.Read"

For more details about permissions <https://techcommunity.microsoft.com/t5/intune-customer-success/support-tip-intune-service-discovery-api-endpoint-will-require/ba-p/2428040>

+ Add a permission ✓ Grant admin consent for inverse inc

API / Permissions name	Type	Description	Admin consent requ...
▼ Azure Active Directory Graph (4)			
Application.Read.All	Application	Read all applications	Yes
Application.ReadWrite.All	Application	Read and write all applications	Yes
Application.ReadWrite.OwnedBy	Application	Manage apps that this app creates or owns	Yes
User.Read	Delegated	Enable sign-on and read users' profiles	No
▼ Intune (1)			
scep_challenge_provider	Application	SCEP challenge validation	Yes
▼ Microsoft Graph (1)			
User.Read	Delegated	Sign in and read user profile	No

Last step is to "Grant admin", just click on "Grant admin consent for ..." and click **Yes**

Microsoft Azure

Home > inverse inc > PacketFence

PacketFence | API permissions

Do you want to grant consent for the requested permissions for all accounts in inverse inc? This will update any existing admin consent records this application already has for these permissions.

Configured permissions

Applications are authorized to call APIs when they are granted permissions by users/admins as part of the consent process. The list of configured permissions should include all the permissions the application needs. [Learn more about permissions and consent](#)

+ Add a permission ✓ Grant admin consent for inverse inc

API / Permissions name	Type	Description	Admin consent req...	Status
▼ Intune (1)				
scep_challenge_provider	Application	SCEP challenge validation	Yes	⚠ Not granted for inverse inc
▼ Microsoft Graph (1)				
User.Read	Delegated	Sign in and read user profile	No	

To view and manage permissions and user consent, try [Enterprise applications](#).

PacketFence configuration

Intune definition

First of all you have to define the configuration parameters to reach the Intune API. To do that go in Configuration → Integration → Cloud Services → New Cloud → Microsoft Intune

Next fill the field with the values taken from the Azure portal ("Application (Client) ID" , "Directory (tenant) ID" and "Client secrets") and **Create**.

Cloud Service **Intune** Intune

Name	Intune
The tenant ID of the intune service	045285d8-2ed4-4c86-9763-e4a3564d5d55 <small>Define the tenant ID defined in the Azure admin portal.</small>
The client ID of the intune service	36e3f54a-e38f-4e37-9470-91ba9d3c0158 <small>Define the client ID defined in the Azure admin portal.</small>
The client secret of the intune service <small>Define the client secret defined in the Azure admin portal.</small>

Save **Clone** **Reset** **Cancel** **Delete**

SCEP configuration

Now let's configure the PKI template to enable SCEP on it. (go to the previous section on how to configure a template in the PKI)

Go in Configuration → Integration → PKI → Templates and edit the one you created previously.

You can see that there is a SCEP section. Enable SCEP and check Enable Cloud Integration and select the Cloud Service you created previously. (In the case the SCEP challenge password is not mandatory).

Starting from now the scep server will be available on each ip where the portal is running (you need to enable the portal on the management interface if you want to be able to do SCEP on this interface).

The URL of the SCEP server will be available on http://ip_adresse/scep/template_name (https too) where template_name is the name of your template in the PKI.

Intune configuration

For this section you can follow the instruction on the Microsoft web site:

<https://docs.microsoft.com/en-us/mem/intune/protect/certificates-profile-scep>

From the PacketFence server you will need to extract the PKI Ca certificate associated to the template and put it in Intune as a "trusted certificate"

Then set the SCEP URL to http://ip_adresse/scep/template_name or https://ip_adresse/scep/template_name

24.3. AirWatch

This section has been created to give a quick overview to configure AirWatch (WMware) with PacketFence. This guide does not include advanced troubleshooting of EAP-TLS connections. Refer to the relevant documentation of EAP-TLS, RADIUS and OpenSSL for advanced features. The PKI comes installed by default since PacketFence version 10. All certificates would be saved in the database. If you want to migrate your certificate from the old PacketFence PKI please see the upgrade section.

24.3.1. Assumptions

You have a functional PacketFence PKI and you already have created a Certificate Authority and its templates with SCEP enabled. The template used here is: 'airwatch2'

NOTE

Make sure that your PacketFence PKI Root CA validity is under 825 days and your PacketFence PKI Template is under 398 days. References: <https://support.apple.com/en-us/HT211025> and <https://support.apple.com/en-ca/HT210176>

Create the Certificate Authority (SCEP):

Certificate Authority - Add/Edit ✕

Name *

Description

Authority Type *

SCEP Provider *

SCEP URL * ⓘ

Challenge Type * STATIC NO CHALLENGE ⓘ

Static Challenge

Max Retries When Pending *

Enable Proxy ENABLED DISABLED ⓘ

Create a Certificate template:

Certificate Template - Add/Edit ×

Name *

Description

Certificate Authority *

Issuing Template

Subject Name

Private Key Length *

Private Key Type * Signing Encryption

SAN Type [Add](#)

Automatic Certificate Renewal ENABLED DISABLED

Auto Renewal Period (days) *

Publish Private Key ENABLED DISABLED

Create the SSID profile:

iOS PF-SCEP-Test

Find Payload

- General
- Passcode
- Restrictions
- Wi-Fi** ①
- VPN
- Email
- Exchange ActiveSync
- Notifications
- LDAP
- CalDAV
- Subscribed Calendars
- CardDAV
- Web Clips
- Credentials ②
- SCEP** ①
- Global HTTP Proxy
- Single App Mode
- Content Filter
- Managed Domains
- Network Usage Rules
- macOS Server Accounts
- Single Sign-On
- Skip Setup Assistant
- SSO Extension
- AirPlay Mirroring
- AirPrint
- Cellular
- Home Screen Layout

Wi-Fi

Service Set Identifier * +

Hidden Network

Auto-Join iOS 5

Security Type WPA/WPA2 Enterprise iOS 14

Prevent MAC address randomization iOS 14

Protocols*

- EAP-TLS
- TTLS
- LEAP
- PEAP
- EAP-FAST
- EAP-SIM
- EAP-AKA

Passpoint

Wi-Fi Hotspot 2.0 iOS 7

Authentication

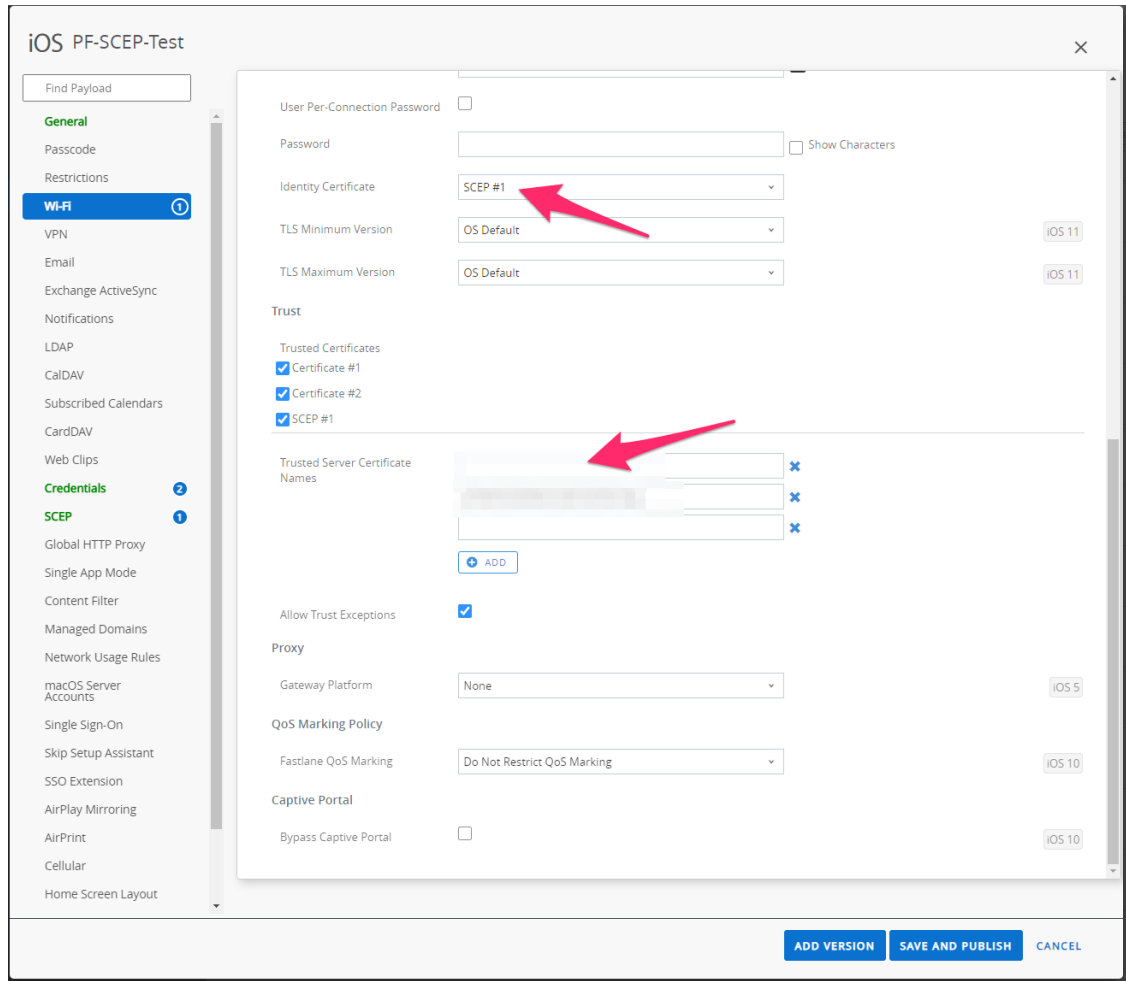
Username +

User Per-Connection Password

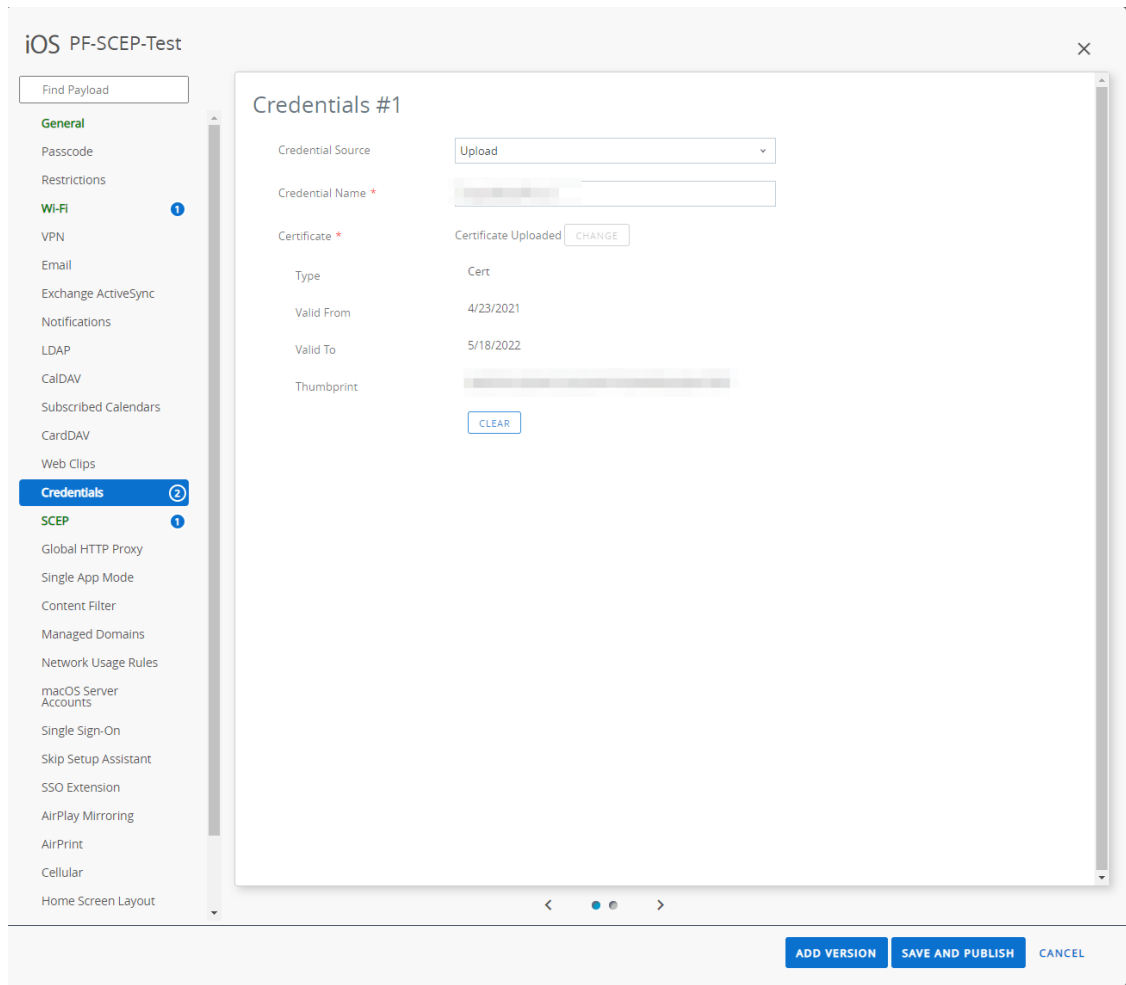
Password Show Characters

Identity Certificate SCEP #1

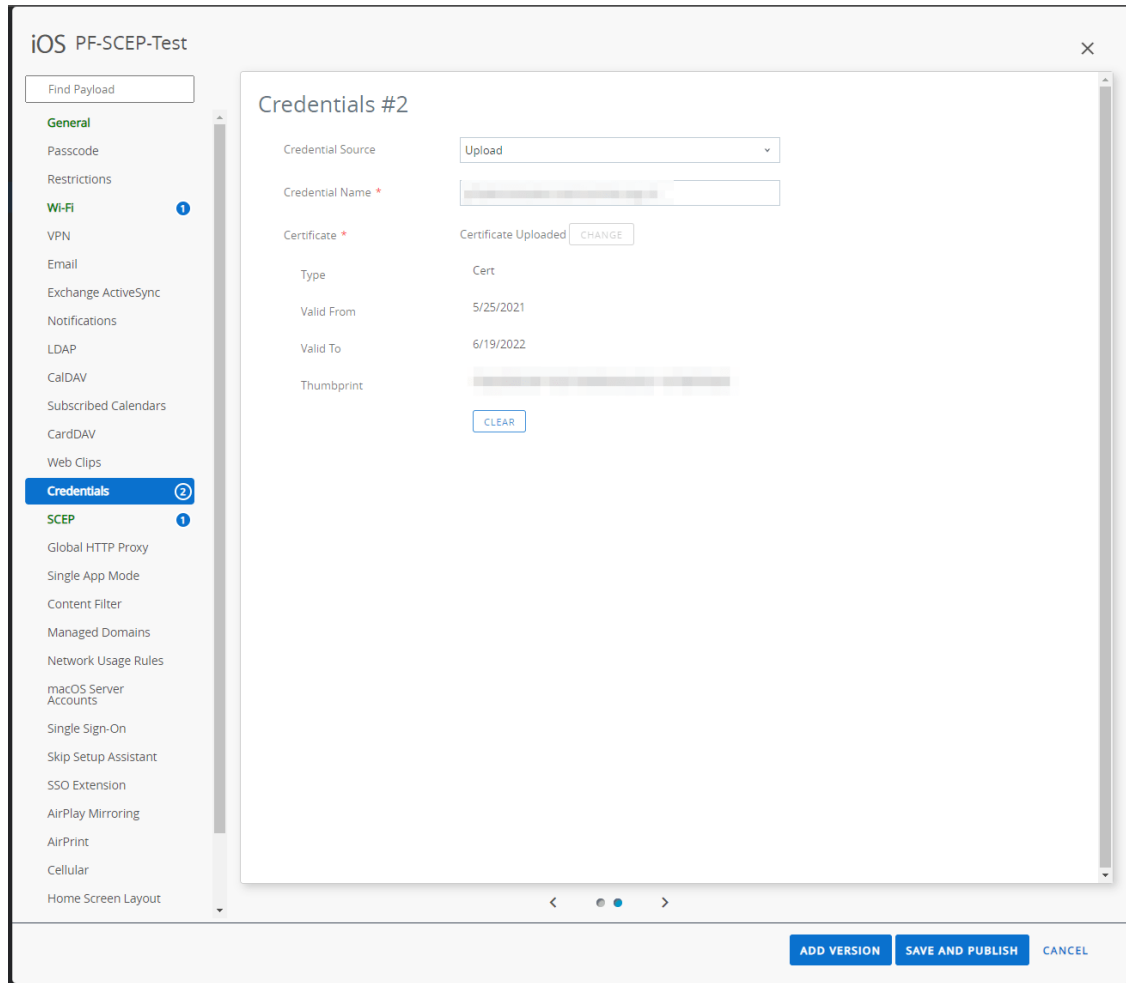
ADD VERSION SAVE AND PUBLISH CANCEL



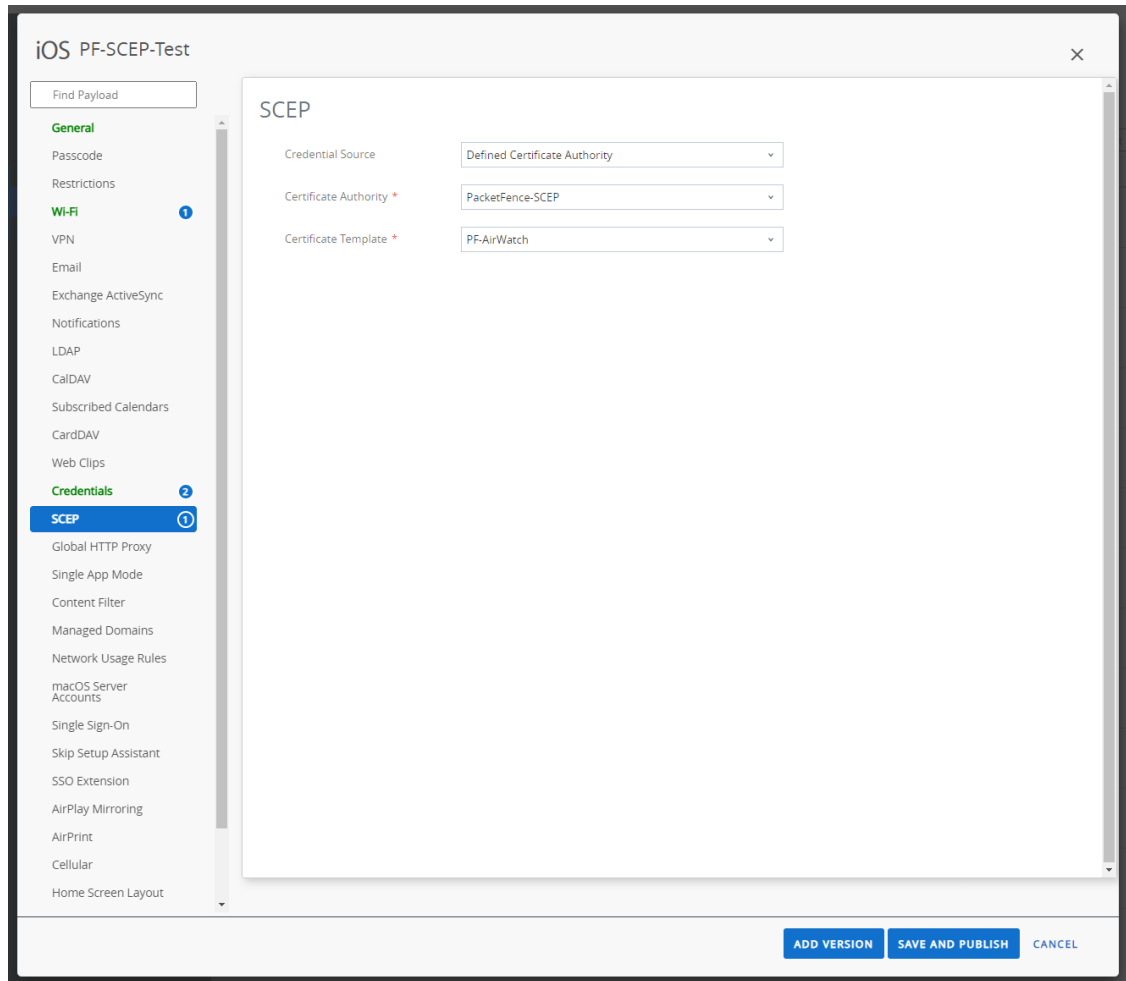
Add the Root CA certificate that issued the PacketFence RADIUS certificate:



Add the PacketFence RADIUS certificate:



Create the SCEP profile:



Assign and deploy the profile:

iOS PF-SCEP-Test

Find Payload

- General
- Passcode
- Restrictions
- Wi-Fi
- VPN
- Email
- Exchange ActiveSync
- Notifications
- LDAP
- CalDAV
- Subscribed Calendars
- CardDAV
- Web Clips
- Credentials
- SCEP
- Global HTTP Proxy
- Single App Mode
- Content Filter
- Managed Domains
- Network Usage Rules
- macOS Server Accounts
- Single Sign-On
- Skip Setup Assistant
- SSO Extension
- AirPlay Mirroring
- AirPrint
- Cellular
- Home Screen Layout

General

Name * PF-SCEP-Test

Version 32

Description PF SCEP Test

Deployment Managed

Assignment Type Auto

Allow Removal Always

Managed By Inspire Development Centers

Smart Groups

- SCEP Test
- Start typing to add a group

Exclusions

NO YES

VIEW DEVICE ASSIGNMENT

Additional Assignment Criteria

- Install only on devices inside selected areas
- Enable Scheduling and install only during selected time periods

Removal Date M/D/YYYY

Hub Required

ADD VERSION SAVE AND PUBLISH CANCEL

25. MFA Integration

This section has been created to give a quick overview on how to configure MFA integration with PacketFence.

25.1. Assumptions

You have a functional PacketFence server and you configured an Internal Source (like Active Directory Source) associated to a "Connection Profile" You also have a radius client that is doing PAP (like a VPN server or a switch with CLI access enabled to use RADIUS).

25.2. Create the MFA Configuration

25.2.1. Akamai MFA

This section has been created to give a quick overview on how to configure Akamai MFA in PacketFence.

25.2.2. Assumptions

You have all the MFA information provided by Akamai to configure in PacketFence.

Create the Multi-Factor configuration

In this section we will configure the Akamai MFA from the administration GUI.

Go in "Configuration→Integration→Multi-Factor Authentication" then click on new MFA and select Akamai.

In the form you have the following information to fill:

Name: Define a name

The App ID of the Akamai MFA: This is the App ID provided by Akamai

The signing key of the Akamai MFA: This is the signing key provided by Akamai

The verify key of the Akamai MFA: This is the verify key provided by Akamai

The host of the Akamai MFA: By default it is mfa.akamai.com

The callback URL to redirect back the user to PacketFence: This parameter is used when you trigger the MFA on the portal, once authenticate on Akamai Bind v2, it redirects to this specific URL to reach back the PacketFence's portal. This value should be the FQDN of the portal with /mfa at the end (<https://portal.acme.com/mfa>)

RADIUS OTP Method: It is where you define which method you want to use in RADIUS (Explanation are covered in the next section)

Character separator: The character used to split the password and OTP when "Strip OTP" RADIUS method is selected.

Cache duration: The amount of time PacketFence will store the MFA information of the user (used for "Strip OTP" and "Second Password Field" since PacketFence deal with multiple RADIUS requests)

Multi-Factor Authentication Akamai MFA Gateway Akamai

Name	Akamai MFA Gateway
The App ID of the Akamai MFA	app_4kB1YKE7wazaEunEMQZW6k <small>Define the App ID provided by Akamai MFA.</small>
The signing key of the Akamai MFA <small>Define the signing key provided by Akamai MFA.</small>
The verify key of the Akamai MFA <small>Define the verify key provided by Akamai MFA.</small>
The host of the Akamai MFA	mfa.akamai.com <small>Define the host of the Akamai MFA.</small>
The callback url to redirect back the user to PacketFence	http://172.20.20.152/mfa <small>Define the callback URL to redirect the user to PacketFence.</small>
RADIUS OTP Method	Strip OTP <small>Define the method to be used in RADIUS to trigger OTP.</small>
Character separator	, <small>Please specify the char to split password field to get the code.</small>
Cache duration	60 <small>seconds</small> <small>The duration time that is use to cache the MFA information. This approximately represent the time for the user to complete the authentication.</small>

25.2.3. TOTP MFA

This section has been created to give a quick overview on how to configure TOTP MFA in PacketFence.

25.2.4. Assumptions

You have a phone where you have an MFA application compatible with TOTP (Akamai MFA, Google Authenticator, Microsoft Authenticator, DUO).

Create the Multi-Factor configuration

In this section we will configure the OTP MFA from the administration GUI.

Go in "Configuration→Integration→Multi-Factor Authentication" then click on new MFA and select TOTP.

In the form you have the following information to fill:

Name: Define a name

RADIUS OTP Method: It is where you define which method you want to use in RADIUS (Explanation are covered in the next section)

Character separator: The character used to split the password and OTP when "Strip OTP" RADIUS method is selected.

Cache duration: The amount of time PacketFence will store the MFA information of the user (used for "Strip OTP" and "Second Password Field" since PacketFence deal with multiples RADIUS request)

New Multi Factor Authentication OTP

Name

RADIUS OTP Method
Define the method to be used in RADIUS to trigger OTP.

Character separator
Please specify the char to split password field to get the code.

Cache duration
The duration time that is use to cache the MFA information. This approximately represent the time for the user to complete the authentication.

25.2.5. Associate the source

The MFA is triggered by a authentication rule in the Internal Source. You have to create a rule with a condition like "memberof equals cn=otp_user,dc=acme,dc=com" and assign an Action:

"Trigger RADIUS MFA" if you want to be triggered in RADIUS "Trigger Portal MFA" if you want to be triggered in the Portal.

Authentication Rules

▼ MFA_Group (MFA Group)

Status Enabled

Name

Description

Matches

Conditions

1	memberof	equals	cn=mfa_group,dc=a	⊖ ⊕
---	----------	--------	-------------------	--------

Actions

1	Trigger RADIUS MFA	Akamai MFA Gateway	⊖ ⊕
2	Trigger Portal MFA	Akamai MFA Gateway	⊖ ⊕
3	Role	default	⊖ ⊕
4	Access duration	1 hour	⊖ ⊕

25.2.6. Portal Flow

Depending of the MFA provider you use, the portal flow will be different.

Akamai Bind v2

This section has been created to give a quick overview on how to configure Akamai Bind V2 in PacketFence.

Assumptions

You have all the Akamai MFA configuration made in PacketFence.

Connection Profile

First you need to have a connection profile that use the Internal Source where you defined a authentication rule that "Trigger Portal MFA" and also use the "Default portal policy" Root Portal Module (There is already the MFA policy defined in it).

Akamai Bind V2 portal

Once you are able to hit the portal and register with your credentials, the portal will forward you to the Akamai Bind V2 web interface. From this page you will be able to onboard your device and also trigger any type of MFA. Once done and authenticated, Akamai Bind V2 portal will forward you back to PacketFence's portal and will grant you the access.

Note: Before using Akamai MFA in the RADIUS flow, you need to onboard your device and it is a way to do it in PacketFence.

TOTP

This section has been created to give a quick overview on how to configure TOTP in PacketFence.

Assumptions

You have all the TOTP MFA configuration made in PacketFence.

Connection Profile

First you need to have a connection profile that use the Internal Source where you defined a authentication rule that "Trigger Portal MFA" and also use the "Default portal policy" Root Portal Module (There is already the MFA policy defined in it).

PacketFence Portal

Once you are able to hit the portal and register with your credentials, the portal will show you a QRcode you will need to scan with your device (Akamai / Goggle / Microsoft / DUO Authenticator per example). This will configure an account where you will be able to see "username.packetfence" and the OTP PIN code.

With that, you will be able to use this OTP on the portal to register your device.

Note: Before using OTP MFA in the RADIUS flow, you need to onboard your device on the portal.

25.2.7. RADIUS Flow

The RADIUS flow depends of the feature of the MFA provider and also depends of the RADIUS client.

Simple RADIUS client

In this use case only the username and password is sent in the RADIUS request, the only method available is the "push" notification. Once the user authenticated, a push notification will be sent on his phone and the user will have to validate in order to be granted.

Simple RADIUS client with password, <code>

In this user scenario the username and password is sent but the password can be splitted with a special character to obtain the code.

OTP code (123456):

The code is the OTP code you will read on your device (the one who change every 30s)

push code (push):

The code can be "push" to use the default phone or "pushx" (x represent the telephone index in the list if you have multiples one), push1 will trigger a push on the first phone, push2 on the second one. The user needs to validate on his phone in order to grant the access.

sms code (sms):

The code can be "sms" to use the default phone or "smsx" (x represent the telephone index in the list if you have multiples one), sms1 will trigger a push on the first phone, sms2 on the second one. The RADIUS request will be rejected and the RADIUS client will prompt again for the credentials.

Once the user receives the code by SMS he will need to reauthenticate with his username and password and append the SMS code. (like password,smscode)

phone code (phone):

The code can be "phone" to use the default phone or "phonex" (x represent the telephone index in the list if you have multiples one), phone1 will trigger a push on the first phone, phone2 on the second one. The RADIUS request will be rejected and the RADIUS client will prompt again for the credentials.

Once the user receives the code by phone call he will need to reauthenticate with his username and password and append the code. (like password,smscode)

Simple RADIUS client with 2nd password

In this user scenario the VPN client presents a login page with one username, password and a second password field. In this 2nd password field you can set multiples things like:

OTP code (123456):

The code is the OTP code you will read on your device (the one who change every 30s)

push code (push):

The code can be "push" to use the default phone or "pushx" (x represent the telephone index in the list if you have multiples one), push1 will trigger a push on the first phone, push2 on the second one. The user needs to validate on his phone in order to grant the access.

sms code (sms):

The code can be "sms" to use the default phone or "smsx" (x represent the telephone index in the list if you have multiples one), sms1 will trigger a push on the first phone, sms2 on the second one. The RADIUS request will be rejected and the RADIUS client will prompt again for the credentials.

Once the user receives the code by SMS he will need to reauthenticate with his username and password and set the code received by SMS in the 2nd password field.

phone code (phone):

The code can be "phone" to use the default phone or "phonex" (x represent the telephone index in the list if you have multiples one), phone1 will trigger a push on the first phone, phone2 on the second one. The RADIUS request will be rejected and the RADIUS client will prompt again for the credentials.

Once the user receives the code by phone call he will need to reauthenticate with his username and password and set the code received by phone in the 2nd password field.

26. Best Practices

26.1. IPTables

IPTables is now entirely managed by PacketFence. However, if you need to perform some custom rules, you can modify `/usr/local/pf/conf/iptables.conf` to your own needs. However, the default template should work for most users.

26.2. Log Rotations

PacketFence can generate a lot of log entries in huge production environments. This is why we recommend to use `logrotate` to periodically rotate your logs. A working `logrotate` script is provided with the PacketFence package. This script is located under the `/usr/local/pf/packetfence.logrotate` file, and it's configured to do a daily log rotation and keeping old logs with compression. It has been added during PacketFence initial installation.

26.3. Large Registration Network

When using the inline or VLAN enforcement mode in large environments, you may have ARP table overflows. This happens when a lot of devices are on the same layer 2 segment. The symptoms are `dhcpcd` not handing out IP addresses as it should or failing pings in the registration or quarantine VLANs. To identify if you have this problem look into your `dmesg` log and if you see `Neighbour table overflow` messages.

In order to mitigate the problem, you need to tweak kernel settings. In order to enlarge the ARP cache table on a live system, change the following in `sysctl.conf` :

```
net.ipv4.neigh.default.gc_thresh1 = 2048
net.ipv4.neigh.default.gc_thresh2 = 4096
net.ipv4.neigh.default.gc_thresh3 = 8192
```

Then run the following as root to enable the changes:

```
# sysctl -p
```

This means that the layer 2 garbage collection will kick in at 2048 MAC addresses exposed to the server with the most aggressive collection kicking in at 8192. This should be large enough for most but feel free to increase if necessary (at the cost of more kernel memory consumed). Another approach to solve this problem is to do more segmentation of your layer 2 networks.

26.4. Active Directory fail-over

The authentication and authorization layer of PacketFence relies on 2 different components to

connect to your Active Directory when doing 802.1x. For authentication, winbindd is used to perform NTLM authentication when doing EAP-PEAP MSCHAPv2. For authorization, LDAP connections are used to compute the role of the user. When using the captive portal or 802.1x authentication that doesn't rely on NTLM authentication (EAP-TLS, EAP-TTLS, etc), then only LDAP is used.

If you have multiple Active Directory servers, you will want to apply the following set of best practices to your installation so that PacketFence is able to efficiently detect a failure of one of your AD server and switch to the next one. This is even more important if your PacketFence deployment points to Active Directory servers located in 2 different availability zones (i.e. 2 different datacenters).

26.4.1. Authentication layer

In order to ensure the authentication layer will be able to fail-over efficiently, you will want to ensure that the 'Sticky DC' parameter of your domain configuration is set to *. Additionally, you will want to specify more than one DNS servers in that configuration. If you have more than one availability zone, then you will want to alternate the order of the servers. For example, if you have the following DNS servers in the first availability zone: 10.0.1.100,10.0.1.101 and the following in the second availability zone: 10.0.2.100,10.0.2.101, then the DNS servers list should be: 10.0.1.100,10.0.2.100,10.0.1.101,10.0.2.101 which will ensure the second DNS server to be queried is part of a different availability zone than the first one when winbindd queries DNS to find an available Active Directory domain controller.

Note that after changing the settings above, you need to regenerate the domain configuration and restart winbindd using:

```
/usr/local/pf/bin/pfcmd generatedomainconfig
/usr/local/pf/bin/pfcmd service winbindd restart
```

Additional safety using monit

Some versions of samba/winbindd may not failover correctly when one of the DC fails, even with the best practices above. For this reason, it is suggested to enable monit on your installation. This will automatically activate an additional check that will restart winbindd if authentication fails to the current DC. Upon restart, a new DC will be found and authentication will resume. To enable this mechanism, enable monit as described [in this section of the document](#) and it be added automatically.

26.4.2. Authorization layer

The authorization layer of PacketFence uses the DNS servers setup on the operating system to resolve names. With that in mind, you will need to ensure that the servers in `/etc/resolv.conf` allow for proper fail-over should one of them fail. Similarly to the authentication layer, you will want to alternate the order of the servers based on the different availability zones you have. You will also want to have aggressive settings for fail-over to the next DNS server. For example, if you have the following DNS servers in the first availability zone: 10.0.1.100,10.0.1.101 and the following in the second availability zone: 10.0.2.100,10.0.2.101, then the resulting `/etc/resolv.conf` should be:

```
search example.com
```

```
options timeout:1
```

```
options retries:1
```

```
nameserver 10.0.1.100
```

```
nameserver 10.0.2.100
```

```
nameserver 10.0.1.101
```

```
nameserver 10.0.2.101
```

Once the DNS servers of the OS are setup to fail-over efficiently, you will need to review the configuration of the different Active Directory sources you have in PacketFence ('Configuration→Policies and access control→Authentication Sources'). In these sources, you will need to ensure that you are either using a DNS name that resolves to multiple servers of your Active Directory domain or that multiple IP addresses are specified to connect. If you are not sure about the robustness of your DNS layer, use multiple IP addresses.

27. Performance Optimizations

27.1. NTLM Authentication Caching

NOTE

This section assumes that you already have an Active Directory domain configuration both in *Configuration → Policies and Access Control → Domains → Active Directory Domains* and *Configuration → Policies and Access Control → Authentication Sources*. If you don't, you need to first configure those. Refer to the appropriate sections of this guide for details on how to configure those two components.

CAUTION

The cache requires minimally Windows Server 2008. Older versions will not work.

When using NTLM authentication against an Active Directory for 802.1X EAP-PEAP connections, this can become a bottleneck when handling dozens of authentications per seconds.

To overcome this limitation, it is possible to use a Redis driven cache inside PacketFence to reduce the amount of authentications requiring an external NTLM authentication call. Should a user be in the cache, PacketFence will attempt to compare the 802.1X credentials with those. In the even that the validation fails, a call to `ntlm_auth` is made. In the event of a cache miss, an `ntlm_auth` call is made as well. This ensures that even if a user changes his password, his new password is immediately valid for 802.1X EAP-PEAP connections even if the cache contains the outdated entry.

NOTE

The NTLM cache doesn't cache clear text passwords, it caches the NT hash of the user password.

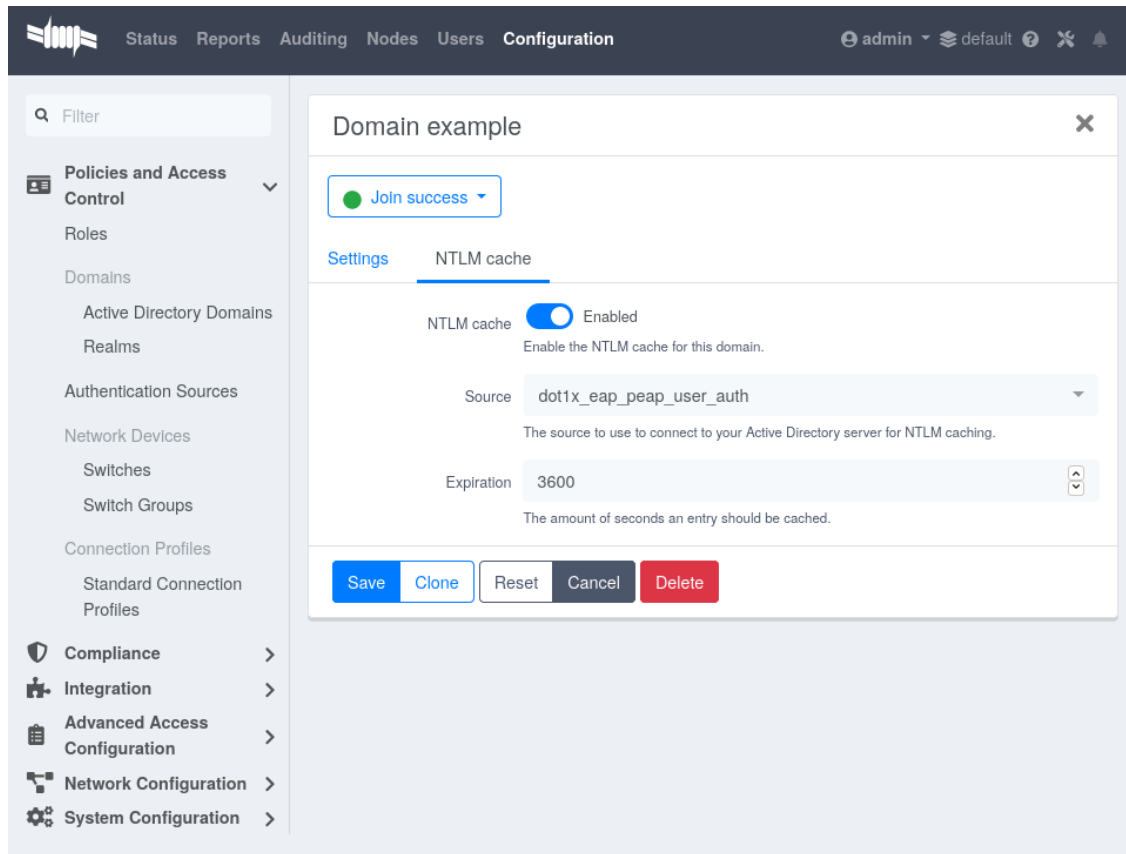
27.1.1. PacketFence Configuration

First of all, you will need to enable the NTLM caching globally by enabling 'NTLM Redis cache' in *Configuration → System Configuration → Radius → General*. You then need to restart `radiusd-auth` service.

Once that is done, you need to configure PacketFence to start caching the credentials. In order to do so, go in *Configuration → Policies and Access Control → Domains → Active Directory Domains* and select the domain you want to cache the credentials for.

Next, go in the **NTLM cache** tab and:

- Enable 'NTLM cache'
- Select the Active Directory authentication source that is tied to this domain.
- Adjust the 'Expiration'



Once done, click on **Save** to commit your changes.

After that, you will need to enable the `redis_ntlm_cache` service which is used by PacketFence to store the cached credentials. In order to do so, go in *Configuration* → *System Configuration* → *Main Configuration* → *Services* and enable 'redis_ntlm_cache' and save the changes.

Next, start the service via pfcmd:

```
/usr/local/pf/bin/pfcmd service redis_ntlm_cache start
```

27.1.2. Active Directory configuration

In order for PacketFence to be able to fetch the NTLM credentials from your Active Directory, it will need a user who has replication rights. The user to which you have to grant the rights, is the one that is configured in the authentication source that you associated in the 'NTLM cache' section of your domain.

Please refer to the following Microsoft KB entry to configure the replication rights (Replicating Directory Changes and Replicating Directory Changes All): <https://support.microsoft.com/en-us/kb/303972>

27.2. SNMP Traps Limit

PacketFence mainly rely on SNMP traps to communicate with equipment. Due to the fact that

traps coming in from approved (configured) devices are all processed by the daemon, it is possible for someone who want to generate a certain load on the PacketFence server to force the generation of non-legitimate SNMP traps or a switch can randomly generate a high quantity of traps sent to PacketFence for an unknown reason.

Because of that, it is possible to limit the number of SNMP traps coming in from a single switch port and take action if that limit is reached. For example, if over 100 traps are received by PacketFence from the same switch port in a minute, the switch port will be shut and a notification email will be sent.

Here's the default config for the SNMP traps limit feature. As you can see, by default, PacketFence will log the abnormal activity after 100 traps from the same switch port in a minute. These configurations are in the `conf/pf.conf` file:

```
[snmp_traps]
trap_limit = enabled
trap_limit_threshold = 100
trap_limit_action =
```

Alternatively, you can configure these parameters from the PacketFence Web administrative GUI, in the *Configuration* → *Network Configuration* → *SNMP* section.

27.3. MariaDB optimizations

27.3.1. Tuning MariaDB

If your PacketFence system is acting very slow, this could be due to your MariaDB configuration. You should do the following to tune performance:

Check the system load

```
# uptime
11:36:37 up 235 days, 1:21, 1 user, load average: 1.25, 1.05, 0.79
```

Check iostat and CPU


```

# iostat 5
avg-cpu:  %user   %nice    %sys %iowait  %idle
           0.60    0.00    3.20  20.20   76.00
Device:            tps   Blk_read/s   Blk_wrtn/s   Blk_read   Blk_wrtn
cciss/c0d0         32.40         0.00       1560.00         0       7800
avg-cpu:  %user   %nice    %sys %iowait  %idle
           0.60    0.00    2.20   9.20   88.00
Device:            tps   Blk_read/s   Blk_wrtn/s   Blk_read   Blk_wrtn
cciss/c0d0         7.80         0.00        73.60         0        368
avg-cpu:  %user   %nice    %sys %iowait  %idle
           0.60    0.00    1.80  23.80   73.80
Device:            tps   Blk_read/s   Blk_wrtn/s   Blk_read   Blk_wrtn
cciss/c0d0        31.40         0.00      1427.20         0       7136
avg-cpu:  %user   %nice    %sys %iowait  %idle
           0.60    0.00    2.40  18.16   78.84
Device:            tps   Blk_read/s   Blk_wrtn/s   Blk_read   Blk_wrtn
cciss/c0d0        27.94         0.00      1173.65         0       5880

```

As you can see, the load-average is 1.25 and iowait is peaking at 20% - this is not good. If your iowait is low but your MariaDB is taking over %50 CPU this is also not good. Check your MariaDB install for the following variables:

```

MariaDB> show variables;
| innodb_additional_mem_pool_size | 1048576 |
| innodb_autoextend_increment    | 8      |
| innodb_buffer_pool_ave_mem_mb  | 0      |
| innodb_buffer_pool_size        | 8388608 |

```

PacketFence relies heavily on InnoDB, so you should increase the `buffer_pool` size from the default values.

Go in the administration GUI , in *Configuration* → *System Configuration* → *Database* → *Advanced* and raise the value of **InnoDB buffer pool size**.

Then restart packetfence-mariadb

```
# systemctl restart packetfence-mariadb
```

Wait 10 minutes re-check iostat and CPU

```

# uptime
12:01:58 up 235 days, 1:46, 1 user, load average: 0.15, 0.39, 0.52
# iostat 5
Device:            tps    Blk_read/s    Blk_wrtn/s    Blk_read    Blk_wrtn
cciss/c0d0         8.00          0.00          75.20         0           376

avg-cpu:  %user   %nice    %sys %iowait  %idle
           0.60    0.00    2.99  13.37  83.03

Device:            tps    Blk_read/s    Blk_wrtn/s    Blk_read    Blk_wrtn
cciss/c0d0        14.97          0.00         432.73         0          2168
avg-cpu:  %user   %nice    %sys %iowait  %idle
           0.20    0.00    2.60   6.60  90.60

Device:            tps    Blk_read/s    Blk_wrtn/s    Blk_read    Blk_wrtn
cciss/c0d0         4.80          0.00         48.00         0          240

```

27.3.2. Avoid "Too many connections" problems

In a wireless context, there tends to be a lot of connections made to the database by our `freeradius` module. The default MariaDB value tends to be low (100) so we encourage you to increase that value to at least 300. See <http://dev.mysql.com/doc/refman/5.0/en/too-many-connections.html> for details.

27.3.3. Avoid "Host <hostname> is blocked" problems

In a wireless context, there tend to be a lot of connections made to the database by our `freeradius` module. When the server is loaded, these connection attempts can timeout. If a connection times out during connection, MariaDB will consider this a connection error and after 10 of these (by default) he will lock the host out with a:

```
Host 'host_name' is blocked because of many connection errors. Unblock with
'mysqldadmin flush-hosts'
```

This will grind PacketFence to a halt so you want to avoid that at all cost. One way to do so is to increase the number of maximum connections (see above), to periodically flush hosts or to allow more connection errors. See <http://dev.mysql.com/doc/refman/5.0/en/locked-host.html> for details.

27.3.4. Using MariaDB-backup

When dealing with a large database, the database backup and maintenance script (`/usr/local/pf/addons/backup-and-maintenance.sh`) which uses `mysqldump` may create a long lock on your database which may cause service to hang.

This is fixed easily by using `MariaDB-backup` which can complete a full database backup without locking your tables.

RHEL-based systems

```
yum install MariaDB-backup --enablerepo=packetfence
```

Debian-based systems (for PacketFence versions 11.0.0 and later)

```
apt install mariadb-backup
```

Debian-based systems (for PacketFence versions prior to 11.0.0)

```
apt install mariadb-backup-10.2
```

Once this is done, grant the proper rights to the **pf** user (or the one you configured in pf.conf):

```
# mysql -u root -p
MariaDB> GRANT PROCESS, RELOAD, LOCK TABLES, REPLICATION CLIENT ON *.* TO
'pf'@'localhost';
MariaDB> FLUSH PRIVILEGES;
```

Next, run the maintenance script `/usr/local/pf/addons/backup-and-maintenance.sh` and ensure that the following line is part of the output:

```
innobackupex: completed OK!
```

If the backup fails, check `/usr/local/pf/logs/innobackup.log` for details and refer to the MariaDB-backup documentation for troubleshooting.

NOTE | In the event that you want to stop using MariaDB-backup for your MariaDB backups, simply uninstall it and the database script will fallback to mysqldump.

27.4. Captive Portal Optimizations

27.5. Troubleshooting

This section will address specific problems and known solutions.

27.5.1. "Internet Explorer cannot display the webpage"

Problem: Internet Explorer 8-10 may raise an "Internet Explorer cannot display the webpage" error while attempting to access PacketFence administration interface because TLSv1.2 is not activated but required since PacketFence 7.

Solution:

- PacketFence administration interface is not started:

```
# cd /usr/local/pf
# bin/pfcmd service httpd.admin start
```

- It is strongly advised that you update your browser to Internet Explorer 11 or download an alternative.
- TLSv1.2 needs to be activated manually in Internet Explorer 8-10.

```
Within Internet Explorer: click `Tools -> Internet Options -> Advanced` and
make sure that TLS v1.2 is enabled under the security section. Retry.
```

28. Advanced Network Topics

28.1. Floating Network Devices

PacketFence supports floating network devices. A Floating network device is a device for which PacketFence has a different behavior compared to a regular device. This functionality was originally added to support mobile Access Points.

CAUTION

Right now PacketFence only supports floating network devices on Cisco and Nortel switches configured with port-security.

For a regular device, PacketFence put it in the VLAN corresponding to its status (Registration, Quarantine or Regular VLAN) and authorizes it on the port (port-security).

A floating network device is a device that PacketFence does not manage as a regular device.

When a floating network device is plugged, PacketFence will let/allow all the MAC addresses that will be connected to this device (or appear on the port) and if necessary, configure the port as multi-vlan (trunk) and set PVID and tagged VLANs on the port.

When an floating network device is unplugged, PacketFence will reconfigure the port like before it was plugged.

28.1.1. How it works

Configuration:

- floating network devices have to be identified using their MAC address.
- linkup/linkdown traps are not enabled on the switches, only port-security traps are.

When PacketFence receives a port-security trap for a floating network device, it changes the port configuration so that:

- it disables port-security
- it sets the PVID
- it eventually sets the port as multi-vlan (trunk) and sets the tagged Vlan
- it enables linkdown traps

When PF receives a linkdown trap on a port in which a floating network device was plugged, it changes the port configuration so that:

- it enables port-security
- it disables linkdown traps

28.1.2. Identification

As we mentioned earlier, each floating network device has to be identified. There are two ways to do it:

- by editing `conf/floating_network_device.conf`
- through the Web GUI, in *Configuration* → *Network Configuration* → *Floating Device*

Here are the settings that are available:

MAC Address

MAC address of the floating device

IP Address

IP address of the floating device (not required, for information only)

trunkPort

Yes/no. Should the port be configured as a multi-vlan port?

pvid

VLAN in which PacketFence should put the port

taggedVlan

Comma separated list of VLANs. If the port is a multi-vlan, these are the Vlan's that have to be tagged on the port.

28.2. Production DHCP access

In order to perform all of its access control duties, PacketFence needs to be able to map MAC addresses into IP addresses.

For all the networks/VLANs where you want PacketFence to have the ability to isolate a node or to have IP information about nodes, you will need to perform **one** of the techniques below.

Also note that this doesn't need to be done for the registration, isolation VLANs and inline interfaces since PacketFence acts as the DHCP server in these networks.

28.2.1. IP Helpers

If you are already using IP Helpers for your production DHCP in your production VLANs this approach is the simplest one and the one that works the best.

Add PacketFence's management IP address as the last `ip helper-address` statement in your network equipment. At this point PacketFence will receive a copy of all DHCP requests for that VLAN and will record what IP were distributed to what node using a `pfdhcp listener` daemon.

By default no DHCP Server should be running on that interface where you are sending the requests. This is by design otherwise PacketFence would reply to the DHCP requests which would be a bad thing.

28.2.2. Obtain a copy of the DHCP traffic

Get a copy of all the DHCP Traffic to a dedicated physical interface in the PacketFence server and run `pfdhcplistener` on that interface. It will involve configuring your switch properly to perform port mirroring (aka network span) and adding in PacketFence the proper interface statement at the operating system level and in `pf.conf`.

`/etc/sysconfig/network-scripts/ifcfg-eth2:`

```
DEVICE=eth2
ONBOOT=yes
BOOTPROTO=none
```

Add to `pf.conf`: (IPs are not important they are there only so that PacketFence will start)

```
[interface eth2]
mask=255.255.255.0
type=dhcp-listener
gateway=192.168.1.5
ip=192.168.1.1
```

Restart PacketFence and you should be good to go.

28.2.3. Interface in every VLAN

Because DHCP traffic is broadcast traffic, an alternative for small networks with few local VLANs is to put a VLAN interface for every VLAN on the PacketFence server and have a `pfdhcplistener` listen on that VLAN interface.

On the network side you need to make sure that the VLAN truly reaches all the way from your client to your DHCP infrastructure up to the PacketFence server.

On the PacketFence side, first you need an operating system VLAN interface like the one below. Stored in `/etc/sysconfig/network-scripts/ifcfg-eth0.1010`:

```
# Engineering VLAN
DEVICE=eth0.1010
ONBOOT=yes
BOOTPROTO=static
IPADDR=10.0.101.4
NETMASK=255.255.255.0
VLAN=yes
```

Then you need to specify in `pf.conf` that you are interested in that VLAN's DHCP by setting type to `dhcp-listener`.

```
[interface eth0.1010]
mask=255.255.255.0
type=dhcp-listener
gateway=10.0.101.1
ip=10.0.101.4
```

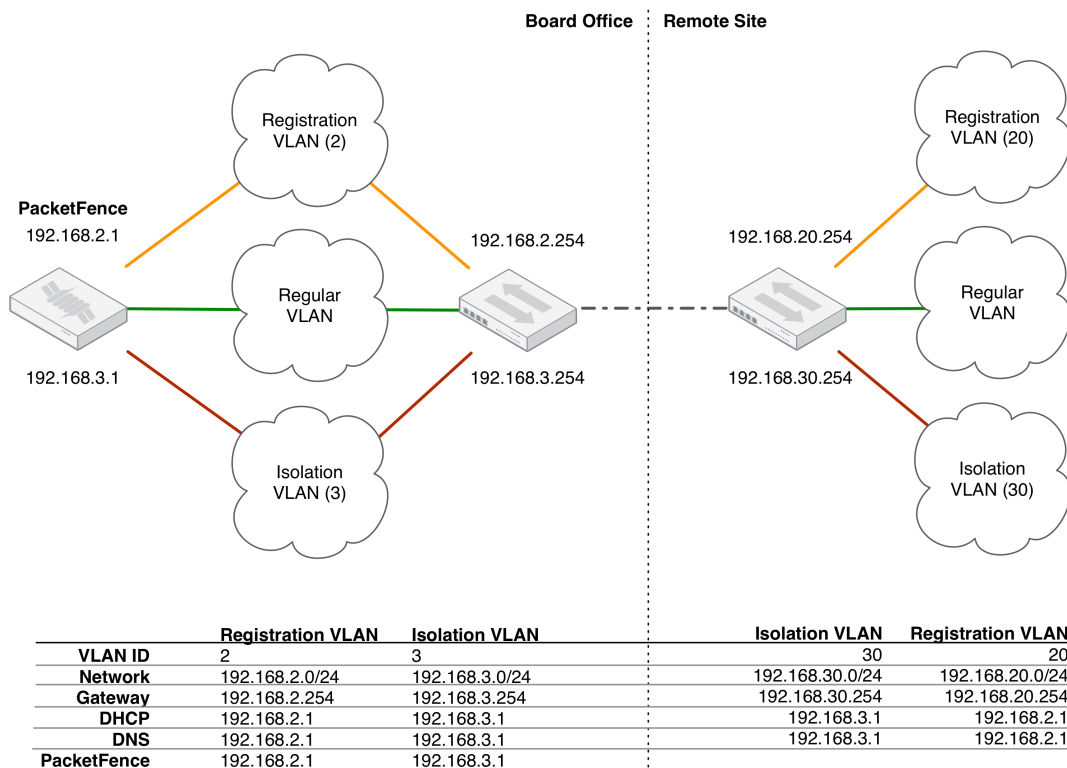
Repeat the above for all your production VLANs then restart PacketFence.

28.2.4. Host production DHCP on PacketFence

It's an option. Just modify `conf/dhcpd.conf` so that it will host your production DHCP properly and make sure that a `pfdhcpplisener` runs on the same interface where production DHCP runs. However, please note that this is **NOT** recommended. See [this ticket](#) to see why.

28.3. Routed Networks

If your isolation and registration networks are not locally-reachable (at layer 2) on the network, but routed to the PacketFence server, you'll have to let the PacketFence server know this. PacketFence can even provide DHCP and DNS in these routed networks and provides an easy to use configuration interface.



For dhcpd, make sure that the clients DHCP requests are correctly forwarded (IP Helpers in the remote routers) to the PacketFence server.

If we consider the network architecture illustrated in the above schema, `conf/pf.conf` will include the local registration and isolation interfaces only.

```
[interface eth0.2]
enforcement=vlan
ip=192.168.2.1
type=internal
mask=255.255.255.0
```

```
[interface eth0.3]
enforcement=vlan
ip=192.168.3.1
type=internal
mask=255.255.255.0
```

NOTE

PacketFence will not start unless you have at least one 'internal' interface, so you need to create local registration and isolation VLANs even if you don't intend to use them. Also, the 'internal' interfaces are the only ones on which dhcpd listens, so the remote registration and isolation subnets need to point their DHCP helper-address to those particular IPs.

Then you need to provide the routed networks information to PacketFence. You can do it through the GUI in **Administration** → **Networks** (or in `conf/networks.conf`).

`conf/networks.conf` will look like this:

```
[192.168.2.0]
netmask=255.255.255.0
gateway=192.168.2.1
next_hop=
domain-name=registration.example.com
dns=192.168.2.1
dhcp_start=192.168.2.10
dhcp_end=192.168.2.200
dhcp_default_lease_time=300
dhcp_max_lease_time=600
type=vlan-registration
named=enabled
dhcpd=enabled
```

```
[192.168.3.0]
netmask=255.255.255.0
gateway=192.168.3.1
next_hop=
domain-name=isolation.example.com
dns=192.168.3.1
dhcp_start=192.168.3.10
dhcp_end=192.168.3.200
dhcp_default_lease_time=300
dhcp_max_lease_time=600
type=vlan-isolation
named=enabled
dhcpd=enabled
```

```
[192.168.20.0]
netmask=255.255.255.0
gateway=192.168.20.254
next_hop=192.168.2.254
domain-name=registration.example.com
dns=192.168.2.1
dhcp_start=192.168.20.10
dhcp_end=192.168.20.200
dhcp_default_lease_time=300
dhcp_max_lease_time=600
type=vlan-registration
named=enabled
dhcpd=enabled
```

```
[192.168.30.0]
netmask=255.255.255.0
gateway=192.168.30.254
next_hop=192.168.3.254
domain-name=isolation.example.com
dns=192.168.3.1
dhcp_start=192.168.30.10
dhcp_end=192.168.30.200
dhcp_default_lease_time=300
dhcp_max_lease_time=600
type=vlan-isolation
named=enabled
dhcpd=enabled
```

Then you need to enable and start `packetfence-routes` service:

```
/usr/local/pf/bin/pfcmd service pf updatesystemd
/usr/local/pf/bin/pfcmd service routes start
```

`packetfence-routes` service will use this file to add static routes corresponding to routed networks (192.168.20.0/24 and 192.168.30.0/24) in PacketFence's server routing table. The `next_hop` statement in each routed network section defines next hop for such routes.

DHCP clients on the registration and isolation networks receive the PF server IP as their DNS server (dns=x.x.x.x), and PF spoofs DNS responses to force clients via the portal. However, clients could manually configure their DNS settings to escape the portal. To prevent this you will need to apply an ACL on the access router nearest the clients, permitting access only to the PF server and local DHCP broadcast traffic.

For example, for the VLAN 20 remote registration network:

```
ip access-list extended PF_REGISTRATION
 permit ip any host 192.168.2.1
 permit udp any any eq 67
 deny ip any any log
interface vlan 20
 ip address 192.168.20.254 255.255.255.0
 ip helper-address 192.168.2.1
 ip access-group PF_REGISTRATION in
```

If your edge switches support 'vlan-isolation' you can also apply the ACL there. This has the advantage of preventing machines in isolation from attempting to attack each other.

28.4. Network Devices Definition (switches.conf)

This section applies only for VLAN enforcement. Users planning to do inline enforcement only can skip this section.

PacketFence needs to know which switches, access points or controllers it manages, their type and configuration. All this information is stored in `/usr/local/pf/conf/switches.conf`. You can modify the configuration directly in the `switches.conf` file or you can do it from the Web Administration panel under *Configuration* → *Policies and Access Control* → *Switches* - which is now the preferred way.

The `/usr/local/pf/conf/switches.conf` configuration file contains a default section including:

- Default SNMP read/write communities for the switches
- Default working mode (see the note below about possible working modes)

and a switch section for each switch (managed by PacketFence) including:

- Switch IP/Mac/Range
- Switch vendor/type
- Switch uplink ports (trunks and non-managed IfIndex)

- per-switch re-definition of the VLANs (if required)

NOTE | `switches.conf` is loaded at startup. A reload is required when changes are manually made to this file `/usr/local/pf/bin/pfcmd configreload`.

NOTE | All the ports declared as uplinks will be ignored and not managed by PacketFence. This parameter is defined in the `[default]` section of `switches.conf`. You can define a different uplink list for each switch.

28.4.1. Switch import from CSV

Using this, you will be able to import a list of switches and update its description and switch group.

NOTE | You must create the switch group prior to importing the switches.

The CSV must have the following format: "description, IP or MAC, switch group". The first line will be skipped. If an entry with one provided IP/MAC already exists it will be updated. In order to only define the switch group through the import, leave the description field empty.

28.4.2. Working modes

There are three different working modes for a switch in PacketFence:

Testing

`pfsetvlan` writes in the log files what it would normally do, but it doesn't do anything.

Registration

`pfsetvlan` automatically registers all MAC addresses seen on the switch ports. As in testing mode, no VLAN changes are done.

Production

`pfsetvlan` sends the SNMP writes to change the VLAN on the switch ports.

28.4.3. RADIUS

To set the RADIUS secret, set it from the Web administrative interface when adding a switch. Alternatively, edit the switch config file (`/usr/local/pf/conf/switches.conf`) and set the following parameters:

```
radiusSecret = secretPassPhrase
```

Moreover, the RADIUS secret is required to support the RADIUS Dynamic Authentication (Change of authorization or Disconnect) as defined in RFC3576.

28.4.4. SNMP v1, v2c and v3

PacketFence uses SNMP to communicate with most switches. PacketFence also supports SNMP v3. You can use SNMP v3 for communication in both directions: from the switch to PacketFence and from PacketFence to the switch. SNMP usage is discouraged, you should now use RADIUS. However, even if RADIUS is being used, some switches might also require SNMP to be

configured to work properly with PacketFence.

From PacketFence to a switch

Edit the switch config file (`/usr/local/pf/conf/switches.conf`) and set the following parameters:

```
SNMPVersion = 3
SNMPEngineID = AA5ED139B81D4A328D18ACD1
SNMPUserNameRead = readUser
SNMPAuthProtocolRead = MD5
SNMPAuthPasswordRead = authpwdread
SNMPPrivProtocolRead = AES
SNMPPrivPasswordRead = privpwdread
SNMPUserNameWrite = writeUser
SNMPAuthProtocolWrite = MD5
SNMPAuthPasswordWrite = authpwdwrite
SNMPPrivProtocolWrite = AES
SNMPPrivPasswordWrite = privpwdwrite
```

From a switch to PacketFence

Edit the switch config file (`/usr/local/pf/conf/switches.conf`) and set the following parameters:

```
SNMPVersionTrap = 3
SNMPUserNameTrap = readUser
SNMPAuthProtocolTrap = MD5
SNMPAuthPasswordTrap = authpwdread
SNMPPrivProtocolTrap = AES
SNMPPrivPasswordTrap = privpwdread
```

Switch Configuration

Here is a switch configuration example in order to enable SNMP v3 in both directions on a Cisco Switch.

```

snmp-server engineID local AA5ED139B81D4A328D18ACD1
snmp-server group readGroup v3 priv
snmp-server group writeGroup v3 priv read v1default write v1default
snmp-server user readUser readGroup v3 auth md5 authpwdread priv aes 128
privpwdread
snmp-server user writeUser writeGroup v3 auth md5 authpwdwrite priv aes 128
privpwdwrite
snmp-server enable traps port-security
snmp-server enable traps port-security trap-rate 1
snmp-server host 192.168.0.50 version 3 priv readUser port-security

```

By default a switch has a SNMPv3 engine identifier (SNMPEngineID), you can get it with `show snmp engineid`.

Test from a PacketFence server

With the `net-snmp` package properly installed, you can test SNMPv3 communication with your switch:

```

snmpget -v3 -l authPriv -u readUser -a MD5 -A "authpwdread" \
-x AES -X "privpwdread" IP_OF_YOUR_SWITCH sysName.0

```

NOTE | Passwords should be at least 8 characters length.

28.4.5. Command-Line Interface: Telnet and SSH

WARNING

Privilege detection is disabled in the current PacketFence version due to some issues (see [#1370](#)). So make sure that the `cliUser` and `cliPwd` you provide always get you into a privileged mode (except for Trapeze hardware).

PacketFence needs sometimes to establish an interactive command-line session with a switch. This can be done using Telnet. You can also use SSH. In order to do so, edit the switch configuration file (`/usr/local/pf/conf/switches.conf`) and set the following parameters:

```

cliTransport = SSH (or Telnet)
cliUser = admin
cliPwd = admin_pwd
cliEnablePwd =

```

It can also be done through the Web Administration Interface under *Configuration* → *Policies and Access Control* → *Switches*.

28.4.6. Web Services Interface

PacketFence sometimes needs to establish a dialog with the Web Services capabilities of a switch. In order to do so, edit the switch config file (`/usr/local/pf/conf/switches.conf`) and

set the following parameters:

```
wsTransport = http (or https)
wsUser = admin
wsPwd = admin_pwd
```

It can also be done through the Web Administration Interface under *Configuration* → *Policies and Access Control* → *Switches*.

28.4.7. Role-based enforcement support

Some network devices support the assignment of a specific set of rules (firewall or ACLs) to a user. The idea is that these rules can be a lot more accurate to control what a user can or cannot do compared to VLAN which have a larger network management overhead.

PacketFence supports assigning roles on devices for switches and WiFi controllers that support it. The current role assignment strategy is to assign it along with the VLAN (that may change in the future). A special internal role to external role assignment must be configured in the switch configuration file (`/usr/local/pf/conf/switches.conf`).

The current format is the following:

```
Format: <rolename>Role=<controller_role>
```

And you assign it to the global `roles` parameter or the per-switch one. For example:

```
adminRole=full-access
engineeringRole=full-access
salesRole=little-access
```

would return the `full-access` role to the nodes categorized as admin or engineering and the role `little-access` to nodes categorized as sales. It can also be done through the Web Administration Interface under *Configuration* → *Policies and Access Control* → *Switches*.

CAUTION

Make sure that the roles are properly defined on the network devices prior to assigning roles!

28.5. More on VoIP Integration

VoIP has been growing in popularity on enterprise networks. At first sight, the IT administrators think that deploying VoIP with a NAC poses a huge complicated challenge to resolve. In fact, depending of the hardware you have, not really. In this section, we will see why.

28.5.1. CDP and LLDP are your friend

For those of you who are unaware of the existence of CDP or LLDP (or LLDP-MED), I suggest you start reading on this topic. Cisco Discovery Protocol (CDP) is device-discovery protocol that runs on all Cisco-manufactured equipment including routers, access servers, bridges, and

switches. Using CDP, a device can advertise its existence to other devices and receive information about other devices on the same LAN or on the remote side of a WAN. In the world of VoIP, CDP is able to determine if the connecting device is an IP Phone or not, and tell the IP Phone to tag its ethernet frame using the configured voice VLAN on the switchport.

On many other vendors, you are likely to find LLDP or LLDP-MED support. Link Layer Discovery Protocol (LLDP) is a vendor-neutral Link Layer protocol in the Internet Protocol Suite used by network devices for advertising their identity, capabilities, and neighbors. Same as CDP, LLDP can tell an IP Phone which VLAN id is the voice VLAN.

28.5.2. VoIP and VLAN assignment techniques

As you already know, PacketFence supports many VLAN assignment techniques such as port-security, MAC authentication or 802.1X. Let's see how VoIP is doing with each of those.

Port-security

Using port-security, the VoIP device rely on CDP/LLDP to tag its ethernet frame using the configured voice VLAN on the switch port. After that, we ensure that a security trap is sent from the voice VLAN so that PacketFence can authorize the mac address on the port. When the PC connects, another security trap will be sent, but from the data VLAN. That way, we will have 1 mac address authorized on the voice VLAN, and 1 on the access VLAN.

NOTE

Not all vendors support VoIP on port-security, please refer to the Network Configuration Guide.

MAC Authentication and 802.1X

On Cisco switches, we are looking at the multi-domain configuration. The multi-domain means that we can have one device on the VOICE domain, and one device on the DATA domain. The domain assignment is done using a Cisco Vendor-Specific Attributes (VSA). When the phone connects to the switchport, PacketFence will respond with the proper VSA only, no RADIUS tunneled attributes. CDP then tells the phone to tag its ethernet frames using the configured voice VLAN on the port. When a PC connects, the RADIUS server will return tunneled attributes, and the switch will place the port in the provided access VLAN.

On other vendor hardware, it is possible to make VoIP work using RADIUS VSAs. When a phone connects to a switchport, PacketFence needs to return the proper VSA to tell the switch to allow tagged frames from this device. When the PC will connect, we will be able to return standard RADIUS tunnel attributes to the switch, that will be the untagged VLAN.

NOTE

Again, refer to the Network Configuration Guide to see if VoIP is supported on your switch hardware.

28.5.3. What if CDP/LLDP feature is missing

It is possible that your phone doesn't support CDP or LLDP. If it's the case, you are probably looking at the "DHCP way" of provisioning your phone with a voice VLAN. Some models will ask for a specific DHCP option so that the DHCP server can give the phone a voice VLAN id. The phone will then reboot, and tag its ethernet frame using the provided VLAN tag.

In order to make this scenario work with PacketFence, you need to ensure that you tweak the registration and your production DHCP server to provide the DHCP option. You also need to

make sure there is a voice VLAN properly configured on the port, and that you auto-register your IP Phones (On the first connect, the phone will be assigned on the registration VLAN).

28.6. DHCP Option 82

PacketFence is able to locate a device on the network even if the switch port is not managed by PacketFence. To use this feature you need to add all the switches in PacketFence and enable SNMP read (switch and PacketFence side) and enable DHCP option 82 in *Configuration* → *Network Configuration* → *Networks* → *Network*. Once enabled, restart the `pfdhcplistener` and `pfmon` (or `pfcron`, if Packetfence version is ≥ 10.2) services. `pfmon` (or `pfcron`, if Packetfence version is ≥ 10.2) will query via SNMP all the switches to create a map (MAC \leftrightarrow switch) `pfdhcplistener` will parse the DHCP Option 82 and will use the map to resolve the MAC to the switch and will update the locationlog of the device.

29. Additional Integration

29.1. DHCP Remote Sensor

The DHCP remote sensor consists of a lightweight binary that is installed on your production DHCP server in order to replicate the DHCP traffic 1 to 1 to the PacketFence server. This solution is more reliable than the DHCP relaying since PacketFence receives a copy of all your DHCP traffic and not only the broadcasted DHCP traffic. Supported DHCP servers are Microsoft DHCP server and CentOS 6 and 7.

These sensors work by capturing the packets at the lowest level possible on your DHCP server and forward them to the PacketFence management interface

29.1.1. Microsoft DHCP Sensor

DHCP-Forwarder is an optimized version of precedent udp-reflector, which installs easily and only copy DHCPREQUESTS and DHCPACK packets to the destination.

[Download the installer here.](#)

It will install WinPCAP, nssm, launch a configurator to ask for interface, IP and port, save the configuration, install and launch DHCP-Forwarder service.

When you will be asked for a host IP and port, specify PacketFence management IP and 767 as the UDP port.

WARNING

On some versions of Windows, the `getmac` command will return invalid output when running the installer in a language other than English. In order to workaround the issue, change your Windows language to English, then logout/login and run the installer again.

The project page can be found [here](#).

29.1.2. Linux-based Sensor

First download the RPM on your DHCP server.

CentOS 6 and 7 servers

For CentOS 6:

```
# for x86_64
# wget http://inverse.ca/downloads/PacketFence/CentOS6/extra/x86_64/RPMS/udp-
reflector-1.0-6.1.x86_64.rpm
```

For CentOS 7:

```
# for x86_64
# wget http://inverse.ca/downloads/PacketFence/CentOS7/extra/x86_64/RPMS/udp-
reflector-1.0-6.1.x86_64.rpm
```

Now install the sensor:

```
# rpm -i udp-reflector-*.rpm
```

Compiling the sensor from source on a Linux system

First make sure you have the following packages installed:

- libpcap
- libpcap-devel
- gcc-c++

Get the source code of the sensor:

```
# mkdir -p ~/udp-reflector && cd ~/udp-reflector
# wget http://inverse.ca/downloads/PacketFence/udp-reflector/udp_reflector.cpp
# g++ udp_reflector.cpp -o /usr/local/bin/udp_reflector -lpcap
```

Configuring the Sensor

Place the following line in `/etc/rc.local`

- where `pcap0` is the pcap interface where your DHCP server listens on. (List them using `udp_reflector -l`)
- where `192.168.1.5` is the management IP of your PacketFence server

```
/usr/local/bin/udp_reflector -s pcap0:67 -d 192.168.1.5:767 -b 25000 &
```

Start the sensor:

```
# /usr/local/bin/udp_reflector -s pcap0:67 -d 192.168.1.5:767 -b 25000 &
```

The DHCP traffic should now be reflected on your PacketFence server.

29.2. Active Directory Integration

29.2.1. Deleted Account

Create the script `unreg_node_deleted_account.ps1` on the Windows Server with the following content. Make sure to change `@IP_PACKETFENCE` to the IP address of your PacketFence server.

You'll also need to change the username and password as they must match the credentials defined in the Web admin interface under *Configuration* → *Integration* → *Web Services*.

```
#####  
#####  
#Powershell script to unregister deleted Active Directory account based on the  
#UserName.#  
#####  
#####  
  
Get-EventLog -LogName Security -InstanceId 4726 |  
    Select ReplacementStrings,"Account name"|  
    % {  
        $url = "https://@IP_PACKETFENCE:9090/"  
        $username = "admin" # Username for the webservices  
        $password = "admin" # Password for the webservices  
        [System.Net.ServicePointManager]::ServerCertificateValidationCallback =  
    {$true}  
        $command = '{"jsonrpc": "2.0", "method": "unreg_node_for_pid", "params":  
    ["pid", "'+$_.ReplacementStrings[0]+'"]}'  
  
        $bytes = [System.Text.Encoding]::ASCII.GetBytes($command)  
        $web = [System.Net.WebRequest]::Create($url)  
        $web.Method = "POST"  
        $web.ContentLength = $bytes.Length  
        $web.ContentType = "application/json-rpc"  
        $web.Credentials = new-object System.Net.NetworkCredential($username,  
    $password)  
        $stream = $web.GetRequestStream()  
        $stream.Write($bytes,0,$bytes.Length)  
        $stream.close()  
  
        $reader = New-Object System.IO.Streamreader -ArgumentList  
    $web.GetResponse().GetResponseStream()  
        $reader.ReadToEnd()  
        $reader.Close()  
    }  
}
```

Create the scheduled task based on an event ID

Start → Run → Task sched.msc

Task Scheduler → Task Scheduler Library → Event Viewer Task → Create Task

General

Name: PacketFence-Unreg_node-for-deleted-account
Check: Run whether user is logged on or not
Check: Run with highest privileges

Triggers → New

Begin on the task: On an event
Log: Security
Source: Microsoft Windows security auditing.
Event ID: 4726

Actions → New

Action: Start a program
Program/script: powershell.exe
Add arguments (optional): C:\scripts\unreg_node_deleted_account.ps1

Settings:

At the bottom, select in the list "Run a new instance in parallel" in order to unregister multiple nodes at the same time.

Validate with Ok and give the account who will run this task. (Usually *DOMAIN\Administrator*)

29.2.2. Disabled Account

Create the script `unreg_node_disabled_account.ps1` on the Windows Server with the following content. Make sure to change `@IP_PACKETFENCE` to the IP address of your PacketFence server. You'll also need to change the username and password as they must match the credentials defined in the Web admin interface under *Configuration → Integration → Web Services*.

```
#####
#####
#Powershell script to unregister disabled Active Directory account based on the
UserName.#
#####
#####

Get-EventLog -LogName Security -InstanceId 4725 |
  Select ReplacementStrings,"Account name" |
  % {
    $url = "https://@IP_PACKETFENCE:9090/"
    $username = "admin" # Username for the webservises
    $password = "admin" # Password for the webservises
    [System.Net.ServicePointManager]::ServerCertificateValidationCallback =
    {$true}
    $command = '{"jsonrpc": "2.0", "method": "unreg_node_for_pid", "params":
    ["pid", "'+$_.ReplacementStrings[0]+'"]}'

    $bytes = [System.Text.Encoding]::ASCII.GetBytes($command)
    $web = [System.Net.WebRequest]::Create($url)
    $web.Method = "POST"
    $web.ContentLength = $bytes.Length
    $web.ContentType = "application/json-rpc"
    $web.Credentials = new-object System.Net.NetworkCredential($username,
    $password)
    $stream = $web.GetRequestStream()
    $stream.Write($bytes,0,$bytes.Length)
    $stream.close()

    $reader = New-Object System.IO.Streamreader -ArgumentList
    $web.GetResponse().GetResponseStream()
    $reader.ReadToEnd()
    $reader.Close()

  }

```

Create the scheduled task based on an event ID

Start → Run → Taskschd.msc

Task Scheduler → Task Scheduler Library → Event Viewer Task → Create Task

General

Name: PacketFence-Unreg_node-for-disabled-account
Check: Run whether user is logged on or not
Check: Run with highest privileges

Triggers → New

Begin on the task: On an event
Log: Security
Source: Microsoft Windows security auditing.
Event ID: 4725

Actions → New

Action: Start a program
Program/script: powershell.exe
Add arguments (optional): C:\scripts\unreg_node_disabled_account.ps1

Settings:

At the bottom, select in the list "Run a new instance in parallel"

Validate with Ok and give the account who will run this task. (Usually *DOMAIN\Administrator*)

29.2.3. Locked Account

Create the script `unreg_node_locked_account.ps1` on the Windows Server with the following content. Make sure to change `@IP_PACKETFENCE` to the IP address of your PacketFence server. You'll also need to change the username and password as they must match the credentials defined in the Web admin interface under *Configuration → Integration → Web Services*.

```
#####
#####
#Powershell script to unregister locked Active Directory account based on the
UserName.#
#####
#####

Get-EventLog -LogName Security -InstanceId 4740 |
  Select ReplacementStrings,"Account name" |
  % {
    $url = "https://@IP_PACKETFENCE:9090/"
    $username = "admin" # Username for the webservises
    $password = "admin" # Password for the webservises
    [System.Net.ServicePointManager]::ServerCertificateValidationCallback =
    {$true}
    $command = '{"jsonrpc": "2.0", "method": "unreg_node_for_pid", "params":
    ["pid", "'+$_.ReplacementStrings[0]+'"]}'

    $bytes = [System.Text.Encoding]::ASCII.GetBytes($command)
    $web = [System.Net.WebRequest]::Create($url)
    $web.Method = "POST"
    $web.ContentLength = $bytes.Length
    $web.ContentType = "application/json-rpc"
    $web.Credentials = new-object System.Net.NetworkCredential($username,
    $password)
    $stream = $web.GetRequestStream()
    $stream.Write($bytes,0,$bytes.Length)
    $stream.close()

    $reader = New-Object System.IO.Streamreader -ArgumentList
    $web.GetResponse().GetResponseStream()
    $reader.ReadToEnd()
    $reader.Close()

  }

```

Create the scheduled task based on an event ID

Start → Run → Taskschd.msc

Task Scheduler → Task Scheduler Library → Event Viewer Task → Create Task

General

Name: PacketFence-Unreg_node-for-locked-account
Check: Run whether user is logged on or not
Check: Run with highest privileges

Triggers → New

Begin on the task: On an event
Log: Security
Source: Microsoft Windows security auditing.
Event ID: 4740

Actions → New

Action: Start a program
Program/script: powershell.exe
Add arguments (optional): C:\scripts\unreg_node_locked_account.ps1

Settings:

At the bottom, select in the list "Run a new instance in parallel"

Validate with Ok and give the account who will run this task. (Usually *DOMAIN\Administrator*)

29.3. Switch Login Access

PacketFence is able to act as an authentication and authorization service on the port 1815 for granting command-line interface (CLI) access to switches. PacketFence currently supports Cisco switches and these must be configured using the following guide: <http://www.cisco.com/c/en/us/support/docs/security-vpn/remote-authentication-dial-user-service-radius/116291-configure-freeradius-00.html>. From the PacketFence's web admin interface, you must configure an Admin Access role (*Configuration* → *System Configuration* → *Admin Access*) that contains the action 'Switches CLI - Read' or 'Switches CLI - Write' and assign this role to an internal user or in an Administration rule in an internal source.

Then you need to enable **CLI Access Enabled** setting on switch(s) you want to manage in *Configuration* → *Network devices* → *Switches*.

NOTE

Any user that has the 'ALL' administrative role will be able to login into your switches. If you want to provide all PacketFence administrative access to some users without allowing them to login into the switches, then apply the 'ALL_PF_ONLY' administrative role which will contains all the necessary PacketFence roles without the switch login.

29.4. Syslog forwarding

Syslog forwarding feature allows you to forward PacketFence logs (all or specific log files) to a remote Syslog server using Syslog protocol.

You can configure this feature in *Configuration → Integration → Syslog Forwarding*

After you add a new Syslog server, you will need to perform following actions using CLI:

```
systemctl restart rsyslog
```

Logs will be kept on PacketFence **and** sent to your remote Syslog server.

29.5. Monit

monit is a utility for managing and monitoring processes, files, directories and filesystems on a Unix system. Monit conducts automatic maintenance and repair and can execute meaningful causal actions in error situations. E.g. Monit can start a process if it does not run, restart a process if it does not respond and stop a process if it uses too much resources.

For further reference the monit documentation is available at: <https://mmonit.com/monit/documentation/monit.html>

The monit configuration path is different between EL and Debian systems:

- **MONIT_PATH=/etc/monit.d** for EL-based systems
- **MONIT_PATH=/etc/monit/conf.d** for Debian-based systems

To simplify further documentation, we will use **\$MONIT_PATH** variable as a reference to these paths.

Starting from PacketFence 11.1, the Monit configuration is directly managed by PacketFence.

In order to enable Monit on your system, go in 'System Configuration→Main Configuration→Monit'. From there, you should configure the following settings:

- Set the 'Status' to 'enabled'
- Set the email addresses you wish to send the alerts to in 'Alert Email To'. If left empty, it will default to the email addresses defined in the 'Alerting' section
- Select the configurations you want monit to use. The defaults should be fine unless your deployment uses port-security enforcement or is an active/passive cluster

For the mailservr:

- You can specify a mailservr to use in the Monit configuration section. This can only be used for unauthenticated relaying. If using localhost, make sure you have postfix installed and properly configured on your server.
- If you do not specify a mailservr in the Monit section, you will want to make sure that the SMTP server settings in the 'Alerting' section are set properly as they will be used to generate the SMTP configuration of monit. Note that monit doesn't support STARTTLS so make sure you use 'none' or 'ssl' as your SMTP encryption in the 'Alerting' section. If you must use

STARTTLS, configure postfix on your local server for relaying and use 'localhost' in the mailserv configuration in the Monit section.

Once done, restart the monit service using:

```
systemctl restart monit
```

29.5.1. Monitoring scripts

Included with the monit configuration is a set of digitally signed scripts that are fetched from <http://inverse.ca/downloads/PacketFence/monitoring-scripts/v1/>. These scripts will be updated and run on a regular interval in order to make sure that your environment follows the best practices defined by Inverse and to alert you of any important change you may need to perform.

You can run them manually for troubleshooting purposes using:

```
/usr/local/pf/addons/monit/monitoring-scripts/update.sh  
/usr/local/pf/addons/monit/monitoring-scripts/run-all.sh
```

Ignoring some checks

Should you want to ignore one of the check that is being performed, add its script name in `$MONIT_PATH/packetfence/local-ignores`.

For example, if you would want to ignore the script that generated the following output

```
-----  
/usr/local/pf/var/monitoring-scripts/.check-epel.sh failed  
Result of /usr/local/pf/var/monitoring-scripts/.check-epel.sh  
The EPEL repository is enabled. This can cause disastrous issues by having the  
wrong versions of certain packages installed. It is recommended to disable it  
using the following command: sed -i 's/enabled\s*=\s*1/enabled = 0/g'  
/etc/yum.repos.d/epel.repo  
-----
```

You would add `/usr/local/pf/var/monitoring-scripts/.check-epel.sh` to the local-ignores file

Run some checks as root

Some scripts need to run as root but are disabled by default. If you want to run these checks, you need to add following content in `$MONIT_PATH/packetfence/local-vars`:

```
export RUN_ROOT_SCRIPTS=1
```

29.5.2. Monit Summary

View the monit summary to ensure all services are status **Running**, **Accessible**, or **Status ok**. Address any services that display any other failed status. Monit will display the services in the same order that they are processed. If the summary appears stuck, troubleshoot the next service in the list.

```
monit summary
```

TIP | More information on the monit command line arguments is available at <https://mmonit.com/monit/documentation/monit.html>

30. Advanced Topics

This section covers advanced topics in PacketFence. Note that it is also possible to configure PacketFence manually using its configuration files instead of its Web administrative interface. It is still recommended to use the Web interface.

In any case, the `/usr/local/pf/conf/pf.conf` file contains the PacketFence general configuration. For example, this is the place where we inform PacketFence it will work in VLAN isolation mode.

All the default parameters and their descriptions are stored in `/usr/local/pf/conf/pf.conf.defaults`.

In order to override a default parameter, define it and set it in `/usr/local/pf/conf/pf.conf`.

`/usr/local/pf/conf/documentation.conf` holds the complete list of all available parameters.

All these parameters are also accessible through the web-based administration interface under the Configuration tab. It is highly recommended that you use the web-based administration interface of PacketFence for any configuration changes.

30.1. Reports

Using the `report.conf` configuration file, you can define reports that create SQL queries to view tables in the PacketFence database. These reports will appear under the *Reports* menu of the administration interface.

PacketFence comes preloaded with several reports that are optimized for most common production use-cases in `reports.conf.defaults`. This file should not be modified, but can be used to provide working examples.

NOTE

Improperly formed reports can consume significant resources on the server. All queries should be profiled and optimized to avoid service outages when executed. Using `type=sql` (script/batch mode) allows increased query and transaction control.

TIP

Master/Slave replication can be used to offload query execution to a read-only database that is not an active member of the cluster. This will ensure that reporting does not degrade the production environment and will provide increased resources to generate the reports.

30.1.1. Configuration Attributes

In order to configure a report, you need to edit `/usr/local/pf/conf/report.conf` and add a section that will define the report. Then do a `/usr/local/pf/bin/pfcmd configreload hard`.

The administration interface builds the structured menu by splitting and separating all the section identifier's by double colons `::`. Identifiers without this separator are shown at the top level. Up

to a maximum of 2 sets of double colons can be used for a maximum menu depth of 3 levels. All identifiers must be unique and any identifier partially reused by another sibling will make it inaccessible. (ex: [A::B] will lose its place as a report and become a parent category for [A::B::*] if it is also defined. Either rename the former to include a 3rd part, or rename the latter to use a different 1st or 2nd part.

[Top Category::Sub Category::Report]

The following attributes are available to define a report (* mandatory attributes are marked with an asterisk):

- **type***: The type of report. Use **type=abstract** to use SQL Abstract and **type=sql** to use MySQL script/batch mode. Each of these types have their own additional attributes which are explained in more detail below.
- **description***: A user-friendly description that provides more details about the report. Used as a title for all charts.
- **charts**: A comma delimited list of charts to display. Each chart is displayed in its own tab above the table data. There is no limit to the number of charts that can be defined. Charts are explained in more detail below.
- **columns***: A comma separated list of columns or aliases that are displayed in the table from the SQL query (ex: **node.mac**, **Node MAC**). The table columns are displayed in the respective order. Columns can be aliased to a more friendly name, but these aliases must be used throughout the other attributes.
- **date_limit**: A PacketFence interval that defines the maximum date range allowed between **start_date** and **end_date**. The reports user is restricted from choosing a date range that exceeds this limit. This is used to prevent the MySQL query from consuming too much resources with large datasets. The duration is defined as **date_limit=[unit][interval]** (ex: **date_limit=1D**), where the unit is a positive integer and the interval is one of the following characters:
 - **s**: second(s)
 - **m**: minute(s)
 - **h**: hour(s)
 - **D**: Day(s)
 - **W**: Week(s)
 - **M**: Month(s)
 - **Y**: Year(s)
- **formatting**: A comma separated list of column or alias formatters. Each column is defined followed by a colon and the internal PacketFence function used to format the column value for every row (ex: **formatting=vendor:oui_to_vendor**). This is used to format the query result columns using a function to access internal PacketFence memory. The supported formatters:
 - **oui_to_vendor**: format a MAC OUI to a vendor.
- **has_date_range**: *[enabled|disabled]* Display a datetime range and provide **start_date** and **end_date** bindings. See **date_limit** to restrict the maximum date range.
- **has_limit**: *[enabled|disabled]* Display a limit selection and provide a **limit** binding.
- **node_fields**: A comma delimited list of fields (columns or aliases) that will be clickable from the table of the Report and linked to the specific Node - only clickable if the reports' user has

the "Node - View" admin role. All fields must be a valid PacketFence node identifier (*mac*).

- **person_fields**: A comma delimited list of fields (columns or aliases) that will be clickable from the table of the Report and linked to the specific User - only clickable if the reports' user has the "User - View" admin role. All fields must be a valid PacketFence user identifier (*pid*).
- **role_fields**: A comma delimited list of fields (columns or aliases) that will be clickable from the table of the Report and linked to the specific Role. All fields must be a valid PacketFence role identifier (*category_id*).

NOTE

Configuration attributes can optionally use a **columnName** reference with simple queries that use a single table (ex: **attribute=columnName,columnB**). The attributes must use **tableName.columnName** reference when using **joins** with 2+ tables (ex: **attribute=tableA.columnA,tableB.columnB**). Aliased **columns** can be used with the table reference (ex: **attribute=tableA.Alias A,tableB.Alias B**).

30.1.2. SQL Abstract

When **type=abstract** PacketFence uses Perl [SQL::Abstract::More](#) to automatically build the SQL query.

The following attributes are available when using **type=abstract**(* mandatory attributes are marked with an asterisk):

- **base_conditions**: A comma delimited list of conditions that is applied to the SQL query. Conditions should match the following format : **field:operator:value** (ex: **auth_log.source:=:sms,auth_log.status:=:completed**).
- **base_conditions_operator**: [*all|any*] The logical SQL operator (AND|OR respectively) used with the **base_conditions**.
- **base_table***: The base SQL table used in the SQL query.
- **date_field***: The table field (column) used to filter by the date range. When used the column will also be used for the default sorting, unless **order_fields** is explicitly defined.
- **group_field**: The field (column) to group the query results by. No grouping is performed if this field is empty or omitted.
- **joins** : The table(s), columns and aliases used to join on the **base_table**. See example below and [the following documentation](#). This attribute supports multi line blocks (heredoc), see below.
- **order_fields**: A comma delimited list of fields (columns) used to order the SQL query. The field should be prefixed of - if the sort should be made in descending order for the field (ex: **-node.regdate,locationlog.start_time,+iplog.start_time**).
- **searches**: A comma delimited list of searchable fields (columns) that are presented to the reports' user. This allows the user to optionally include additional criteria for the query. Each item is defined as **type:Friendly Name:tableName.columnName** (ex: **searches=string:Owner:person.pid,string:Node:node.mac**). Currently only the type *string* is supported.
 - **type** defines the type of the search, the only one currently supported is **string**.
 - **Display Name** is the user-friendly name of the field for display.
 - **field** is the SQL name of the field to search

WARNING

Replace operators **IS** and **<>** by **=** and **!=**, respectively.

NOTE

Prefix the fields with the table name and a dot (ex: `node.mac`, `locationlog.role`, ...) so that they are not ambiguous. Wrap table names and column names with backticks `` ` `` to avoid naming issues with current and future MySQL reserved words.

Examples

View of the `auth_log` table:

```
[auth_log]
description=Authentication report
# The table to search from
base_table=auth_log
# The columns to select
columns=auth_log.*
# The date field that should be used for date ranges
date_field=attempted_at
# The mac field is a node in the database
node_fields=mac
# Allow searching on the PID displayed as Username
searches=string:Username:auth_log.pid
```

In this simple example, you will be able to select the whole content of the `auth_log` table and use the date range on the `attempted_at` field as well as search on the `pid` field when viewing the report.

View of the opened security events:


```

[open_security_events]
description=Open security events
# The table to search from
base_table=security_event
# The columns to select
columns=security_event.security_event_id as "Security event ID",
security_event.mac as "MAC Address", class.description as "Security event
description", node.computername as "Hostname", node.pid as "Username",
node.notes as "Notes", locationlog.switch_ip as "Last switch IP",
security_event.start_date as "Opened on"
# Left join node, locationlog on the MAC address and class on the security
event ID
joins=<<<EOT
=>{security_event.mac=node.mac} node|node
=>{security_event.mac=locationlog.mac} locationlog|locationlog
=>{security_event.security_event_id=class.security_event_id} class|class
EOT
date_field=start_date
# filter on open locationlog entries or null locationlog entries via the
end_date field
base_conditions_operator=any
base_conditions=locationlog.end_time::=0000-00-00,locationlog.end_time:IS:
# The MAC Address field represents a node
node_fields=MAC Address
# The Username field represents a user
person_fields=Username

```

In the example above, you can see that the `security_event` table is *left joined* to the `class`, `node` and `locationlog` tables. Using that strategy we make sure all the security events are listed even on deleted nodes. Then, base conditions are added to filter out outdated `locationlog` entries as well as include devices without `locationlog` entries. Removing those conditions would lead to duplicate entries being shown since the report would reflect all the historical `locationlog` entries.

30.1.3. SQL

When `type=sql` PacketFence uses MySQL script/batch mode to manually build the SQL query including the execution of multiple statements. This provides complete query control as well as the ability to manage the SQL session and the SQL transaction. This is the preferred mode where SQL optimization is needed to execute complex queries, or for those more comfortable with raw (non-abstract) SQL.

```
sql=SELECT * FROM sponsors;
```

Multiline block (heredoc) is required when executing multiple statements. Each statement should be terminated with a semi-color ";".

NOTE

SQL execution exits on the first error and returns the result set of the last successful statement.

The following attributes are available when using `type=sql`:

- **bindings**: A comma delimited list of ordered bindings to send to the SQL script (ex: `bindings=tenant_id,start_date,end_date,cursor,limit`). See Bindings below.
- **cursor_type**: `[node|field_multi_field]` Adds a cursor binding to the sql script that implements pagination of the results. The cursor is automatically handled in the administration interface, but its use in the `sql` requires special attention. If omitted the default `none` is used. More information about cursors is provided below. There are 2 types of cursors:
 - **cursor_type=field**: Use a single field (column or alias) for the cursor.
 - **cursor_type=multi_field**: Use multiple fields (columns or aliases) for the cursor.
 - **cursor_type=offset**: Use integer based offset for the cursor.
 - **cursor_type=none**: No cursor is used.
- **cursor_default**: The default cursor used to conditionally query the results for the first page. On subsequent pages this is replaced with the results from N+1 row of the previous page, meaning the cursor for page 2 (with `default_limit=25`) will contain the value from the column of the 26th row from the previous page.
- **cursor_field**: A comma delimited list of fields (columns) used for pagination.
- **default_limit**: The default limit passed into the bindings of the SQL script. When `has_limit=enabled` the reports' user can override the default with a manual selection.
- **sql**: Either a single MySQL query, or a multi line block of statements within a heredoc (see Heredoc below).

30.1.4. Bindings

The **bindings** attribute defines an ordered comma delimited list of columns (or aliases) that are made available to the `sql` script. There is no limit with the number of bindings that can be used and a binding can be repeated more than once.

The available bindings are:

- **tenant_id**: The scoped tenant identifier of the reports' session.
- **start_date, end_date**: The start and end datetime. Formatted as "YYYY-MM-DD HH:mm:ss". Use native MySQL date functions to reformat it.
- **cursor**: On the first page this value is the **cursor_default**. On subsequent pages this value is taken from the **cursor_field** column of the last result row from the previous page. When using **cursor_type=multi_field** the cursor is split into the bindings as `cursor.0, cursor.1`, etc.
- **limit**: Uses **default_limit** (+1, see pagination) unless overridden by the user.

Bindings are consumed in the `sql` using "?" in the same order that they are defined.

```
[single binding]
type=sql
bindings=limit
sql=SELECT * FROM table LIMIT ?;
default_limit=100
has_limit=enabled
```

If a binding is needed more than once within the `sql`, it can either be defined multiple times, or defined once and consumed to SET a MySQL variable.

```
[many bindings]
type=sql
bindings=start_date,end_date,tenant_id,start_date,end_date,limit
sql= << EOT
    SELECT
        *
    FROM tableA
    JOIN tableB ON tableA.id = tableB.id
        AND date BETWEEN ? AND ?
    WHERE tenant_id = ?
        AND date BETWEEN ? AND ?
    LIMIT ?;
EOT
default_limit=100
has_date_range=enabled
has_limit=enabled
```

This can be improved as:

```

[many bindings]
type=sql
bindings=tenant_id,start_date,end_date,limit
sql= << EOT
BEGIN NOT ATOMIC
  SET @TENANT_ID = ?, @START_DATE = ?, @END_DATE = ?;
  SELECT
    *
  FROM tableA
  JOIN tableB ON tableA.id = tableB.id
    AND date BETWEEN @START_DATE AND @END_DATE
  WHERE tenant_id = @TENANT_ID
    AND date BETWEEN @START_DATE AND @END_DATE
  LIMIT ?;
END;
EOT
default_limit=100
has_date_range=enabled
has_limit=enabled

```

NOTE MySQL LIMIT will not accept a variable (`LIMIT @LIMIT`). A best practice is to always pass limit as the last field in `bindings` and to simply consume it at the end of the script (`LIMIT ?`) as shown in the last example above.

30.1.5. Pagination

Pagination is supported through the use of the `cursor_type`, `cursor_default`, `cursor_field`, `bindings` and `sql` attributes. Pagination supports the use of one to many columns. Special attention must be given to the order of the final result set in order to utilize the cursor properly. Symptoms of too few pages, or infinite loops through subsequent pages are signs of a mismatched cursor and/or query results order.

The `limit` binding always has +1 added to it as PacketFence always consumes an extra row to determine the cursor for the following page. Due to this all conditional statements must be inclusive (ex: Bad operators "<, >", Good operators: "`>=`", "`<=`"). If the column value is not unique then `cursor_type=multi_field` should be used instead to avoid infinite loops.

Examples of a single column cursor:

```

[all nodes in ascending order]
type=sql
sql= <<EOT
    SELECT mac FROM node WHERE mac >= ? ORDER BY mac LIMIT ?;
EOT
bindings=cursor,limit
cursor_type=field
cursor_field=mac
default_cursor=00:00:00:00:00:00

```

```

[all nodes in descending order]
type=sql
sql= <<EOT
    SELECT mac FROM node WHERE mac <= ? ORDER BY mac DESC LIMIT ?;
EOT
columns=mac
bindings=cursor,limit
cursor_type=field
cursor_field=mac
default_cursor=ff:ff:ff:ff:ff:ff

```

Example of a multi column cursor:

```

[all ip4log logs]
type=sql
sql= <<EOT
    SELECT
        ip4log.ip,
        ip4log.start_time,
        node.mac
    FROM ip4log
    INNER JOIN node
        ON ip4log.mac = node.mac
    WHERE ip4log.start_time >= ?
        AND node.mac >= ?
    ORDER BY ip4log.start_time, node.mac
    LIMIT ?;
EOT
columns=mac
bindings=cursor.0,cursor.1,limit
cursor_type=multi_field
cursor_field=start_time,mac
default_cursor=0000-00-00 00:00:00:00,00:00:00:00:00:00

```

30.1.6. Charts

Charts are defined as a comma delimited list using the `chart` attribute. An optional "@" symbol can be used to delimit a chart name. A mandatory pipe (vertical-bar) `|` is used to delimit the chart type and the fields. Within the fields a colon ":" is used to delimit each of the fields (if more than one field is necessary). The general syntax is:

```
charts=[pie,bar,parallel,scatter] [@ Chart Name] | field1 [:fieldN:...]
```

There are 4 types of charts available:

- **pie**: A pie chart with 2 dimensions. Must contain 2 fields (`charts=pie|field1:field2`):
 - `field0`: The dimensions label.
 - `field1`: The dimensions value.
- **bar**: A bar chart with 2 dimensions. Must contain 2 fields (`charts=bar|field1:field2`):
 - `field0`: The dimensions label.
 - `field1`: The dimensions value.
- **parallel**: A parallel category (sankey) diagram with 2+ dimensions. Must contain 3+ fields (`charts=parallel|field1:field2:field3[...:fieldN]`):
 - `fieldN`: The N dimensions label of 2+ fields. A category is created for each field and order is maintained. The palette is applied to the last field (right-most).
 - `fieldLast`: The last field always contains the dimensions value.
- **scatter**: A date/time based line graph with 1+ dimensions. The date/time column is always defined in the first field and the query should return this using the "YYYY-MM-DD HH:mm:ss" format.
 - When only one field is defined (`charts=scatter|field1`) then a value of 1 is implied for each row.
 - When 2 fields are defined (`charts=scatter|field1:field2`) then the 2nd field is used as the dimensions value. The query results are automatically aggregated to produce dimensions for several terms (year/month/week/day/hour/minute).
 - When 3+ fields are defined (`charts=scatter|field1:field2:field3[...:fieldN]`) the automatic aggregation is disabled and a dimension is used for each field.

NOTE | All charts use the same color palette to provide a visual continuity.

30.1.7. Heredoc

The `joins` and `sql` attribute support multi line block statements. All whitespace characters are preserved. All multi line statements are pure SQL, thus the `--` prefix can be used as a remark.

```
attribute= <<EOT
  -- multi-line
  -- block
  -- statement
EOT
```

Heredoc supports multiple SQL statements. When running multiple statements they should be wrapped in `BEGIN NOT ATOMIC ... END;`, which is not required when using a single statement:

```
[multi statement example]
type=sql
sql= <<EOT
BEGIN NOT ATOMIC
  -- consume bindings and SET friendly SQL variables (except LIMIT)
  SET @TENANT_ID = ?, @START_DATE = ?, @END_DATE = ?;
  SELECT
    *
  FROM tableA
  WHERE tenant_id = @TENANT_ID
    AND columnA BETWEEN @START_DATE AND @END_DATE
  -- SQL LIMIT doesn't support SQL variables
  LIMIT ?;
END;
EOT
bindings=tenant_id,start_date,end_date,limit
```

30.1.8. Troubleshooting

- If the API request returns an error or an empty response refer to the `packetfence.log` to obtain the full MySQL error message.
- SQL scripts are transactional. After the script is run any variables or stored procedures created or temporary tables created are destroyed. Any locks obtained are released.
- Modification to the configuration file only requires a `/usr/local/pf/bin/pfcmd configreload hard` for the changes to take effect. The administration interface will begin using the new script on its next request.

30.2. Admin Access

You can manage which access you give to PacketFence administrators. To do that go through *Configuration* → *System Configuration* → *Admin Access*. Then go to your source which authenticate administrator and create an *administration* rule and assign the wanted Admin role. This functionality allows you to have a granular control on which section of the admin interface is available to whom.

30.2.1. Built-in roles

- ALL: Provides the user with all the admin roles without any exception.
- ALL_PF_ONLY: Provides the user with all the admin roles related to the PacketFence deployment (excludes switch login rights).
- Node Manager: Provides the user the ability to manage the nodes.
- User Manager: Provides the user the ability to manage other users.
- Security Event Manager: Provides the user the ability to manage the security events (trigger, open, close) for the nodes.

30.3. Guest pre-registration

Pre-registration is disabled by default. Once enabled, PacketFence's firewall and Apache ACLs allow access to the `/signup` page on the portal even from a remote location. All that should be required from the administrators is to open up their perimeter firewall to allow access to PacketFence's management interface IP on port 443 and make sure a domain name to reach said IP is configured (and that the SSL cert matches it). Then you can promote the pre-registration link from your extranet web site: <https://<hostname>/signup>.

To minimally configure guest pre-registration, you must make sure that the following statement is set under `[guests_self_registration]` in `/usr/local/pf/conf/pf.conf`:

```
[guests_self_registration]
preregistration=enabled
```

This parameter should be configured from the *Configuration* → *Policies and Access Control* → *Connection Profiles* → *Profile Name* section.

- | | |
|----------------|--|
| CAUTION | A valid MTA configured in PacketFence is needed to correctly relay emails related to the guest module. If <code>localhost</code> is used as <code>smtpserver</code> , make sure that a MTA is installed and configured on the server. |
| CAUTION | Pre-registration increases the attack surface of the PacketFence system since a subset of it's functionality is exposed on the Internet. Make sure you understand the risks, apply the critical operating system updates and apply PacketFence's security fixes. |
| NOTE | A 'portal' interface type is required to use this feature. A 'portal' interface type can be added to any network interface using the web admin GUI. |

30.4. Content-Security-Policy (CSP)

The Content-Security-Policy HTTP response header tells modern browsers what can be accessed from a generated web page. The default policy is pushed for the captive portal and enforces that everything the browser executes comes from within PacketFence, with the exception of the configured network detection host that is by default the Inverse IP address.

If, for some reason the portal is modified with content that needs to be accessed from

PacketFence generated web pages, CSP can be deactivated through *Configuration* → *System Configuration* → *Main Configuration* → *Advanced* → *CSP Security Headers*.

30.5. pfacct: track bandwidth usage

Starting from v10, **pfacct** daemon is used to track bandwidth usage of nodes using **RADIUS Accounting** or NetFlow v5 traffic. It is enabled by default and replaced **packetfence-radiusd-acct** service. **pfacct** will store data into **bandwidth_accounting** table. Using a security event with a bandwidth limit trigger, you can limit data usage of your nodes. GUI also use **bandwidth_accounting** table informations to display online/offline status of nodes. Bandwidth usage reports are available in *Reports* menu under *Accounting* section.

30.5.1. NetFlow traffic

pfacct can get NetFlow traffic from two kind of sources:

- network devices which send directly NetFlow traffic to PacketFence
- inline L2/L3 networks (using NetFlow kernel module)

By default, **pfacct** listens NetFlow traffic on localhost, using **udp/2056** port to not conflict with the **fingerbank-collector** (which listens NetFlow traffic on all interfaces).

pfacct must be able to map an IP address to a MAC address (from NetFlow traffic) in order to create a record in **bandwidth_accounting** table. It means that PacketFence needs to be aware of IP addresses of your nodes (default behavior on inline L2/L3 networks).

You need to adjust **pfacct** configuration based on your NetFlow traffic source.

NetFlow traffic from network devices

You need to:

- make **pfacct** listens on IP address where you want to receive NetFlow traffic using **netflow_address** setting in *Configuration* → *System configuration* → *Services* menu
- enable *NetFlow on all networks* in *Configuration* → *System configuration* → *Advanced* menu

Then restart **packetfence-iptables** and **packetfence-pfacct** services for it to take effect.

NetFlow traffic from inline L2/L3 networks

You need to enable *Netflow Accounting Enabled* setting when defining an inline network.

If you enable *NetFlow on all networks* in *Configuration* → *System configuration* → *Advanced* menu, **pfacct** will collect NetFlow bandwidth usage for all networks instead of the ones defined in **/usr/local/pf/conf/networks.conf**.

Then restart **packetfence-iptables** and **packetfence-pfacct** services for it to take effect.

31. Export/Import mechanism

This section covers export/import mechanism available since PacketFence 11.0.0. It can be used to automate parts of upgrades or to restore PacketFence installations.

31.1. Assumptions and limitations

- You can export on any PacketFence version above 10.3.0
- You can import on any PacketFence version above 11.0.0
- The import process needs to be done on a **standalone** server. Restoring directly to clusters is currently unsupported
 - NOTE: Once you restored to your standalone server, you can make it a cluster by joining other machines to it and creating your **cluster.conf** but this is relatively advanced and out of scope of this document
- Restoring on a fresh install of PacketFence is recommended although restoring on an existing instance can work but your mileage may vary
- The import process will not modify network cards configuration of your server: it will only update PacketFence IP configuration. We recommend you to define targeted IP addresses on network cards before running import process even if you can do it at end of import process.
- The import process will not join automatically server to Active Directory domains. You need to rejoin server manually.
- The import process will only restore the files that can be edited via the admin interface which include:
 - Standard configuration files in **/usr/local/pf/conf/*.conf**
 - Connection profiles HTML templates in **/usr/local/pf/html/captive-portal/profile-templates/**
 - Standard certificates
 - **/usr/local/pf/conf/ssl/***
 - **/usr/local/pf/raddb/certs/***
- Here is a short list of the configuration files that will not be restored. Changes to these files need to be migrated manually. This list is not meant to be complete:
 - **/usr/local/pf/conf/radiusd/***
 - **/usr/local/pf/conf/log.conf**
 - **/usr/local/pf/conf/log.conf.d/***
 - **/usr/local/pf/conf/iptables.conf**
 - **/usr/local/pf/conf/cluster.conf**

31.2. Export on current installation

NOTE

When you are in a cluster, you need to perform this process on the first member of the incoming addresses of your database cluster. To find the member, run `show status like 'wsrep_incoming_addresses'`; inside your MariaDB instance and the first IP will be the one where you need to perform the export process.

31.2.1. Installation (for PacketFence version 10.3.0 only)

On PacketFence version 10.3.0, you need to install `packetfence-export` package using following instructions:

RHEL / CentOS based systems only

```
yum localinstall
http://packetfence.org/downloads/PacketFence/RHEL8/packetfence-export-
11.1.e18.noarch.rpm
```

Debian 9 systems only

```
wget http://packetfence.org/downloads/PacketFence/debian/packetfence-
export_11.1.deb
dpkg -i packetfence-export_11.1.deb
```

31.2.2. Start the export process

The export process will try to use files created by the nightly backup done at 00:30am everyday. If this fine for you and you don't need the latest data, then you can skip this step. Otherwise to have the latest data and configuration in your export, run:

```
/usr/local/pf/addons/backup-and-maintenance.sh
```

Next, run the export script:

```
/usr/local/pf/addons/full-import/export.sh /tmp/export.tgz
```

The command above will create your export archive in `/tmp/export.tgz`. You will now need to copy this file to your new server using SCP or your preferred mechanism.

31.3. Import on new installation

You first need to have a PacketFence installation with latest version done on a standalone server following the instructions in our install guide. You don't need to go through the configurator unless you want to modify IP settings of the server.

WARNING

If you want to use the first step of the configurator to configure your server, you need to do it **before** running your import.

31.3.1. Preliminary steps

If your export archive used MariaDB-backup instead of `mysqldump` (your DB backup filename in `/root/backup/` contains `xbstream`), then you need to install identical MariaDB-backup version on your **new** installation.

If you are restoring from PacketFence 10.3.0 only

RHEL / CentOS based systems

```
yum remove MariaDB-backup
yum localinstall
https://www.packetfence.org/downloads/PacketFence/CentOS7/x86_64/RPMS/MariaDB-
backup-10.2.37-1.el7.centos.x86_64.rpm
```

Debian based systems

```
wget -O /root/mariadb-backup-10.2_10.2.37.deb
https://www.packetfence.org/downloads/PacketFence/debian-
lastrelease/pool/stretch/m/mariadb-10.2/mariadb-backup-
10.2_10.2.37+maria~stretch_amd64.deb
dpkg-deb -xv /root/mariadb-backup-10.2_10.2.37.deb /root/mariadb-backup
mv /root/mariadb-backup/usr/bin/mariabackup /usr/local/bin/mariabackup
mv /root/mariadb-backup/usr/bin/mbstream /usr/local/bin/mbstream
```

If you are restoring from PacketFence 11.0 or above

RHEL / CentOS based systems

```
yum install MariaDB-backup --enablerepo=packetfence
```

Debian based systems

```
apt install mariadb-backup
```

31.3.2. Start the import process

The import script will guide you through the restore of the database, the configuration files and will help adjust the PacketFence IP configuration if necessary.

To start the import process using the export archive you made on the current installation:

```
/usr/local/pf/addons/full-import/import.sh /tmp/export.tgz
```

Once the process is completed, you should see the following:

```
Completed import of the database and the configuration! Complete any necessary
adjustments and restart PacketFence
```

If that's not the case, check the output above to understand why the process failed.

If all goes well, you can restart services using [following instructions](#).

Additional steps to build or rebuild a cluster

If you want to build or rebuild a cluster, you need to follow instructions in [Cluster setup section](#).

If your previous installation was a cluster, some steps may not be necessary to do. Your export archive will contain your previous `cluster.conf` file.

31.3.3. Final steps

If you restored from PacketFence 10.3.0 and you installed MariaDB-backup for your restore during the 'Preliminary steps', you need to update it to the latest version:

RHEL / CentOS based systems

```
yum update MariaDB-backup --enablerepo=packetfence
```

Debian based systems

```
rm /usr/local/bin/mariabackup
rm /usr/local/bin/mbstream
```

32. Automation of upgrades

This section covers automation of upgrades available since PacketFence 11.0.0.

32.1. Assumptions and limitations

- You can perform automated upgrades on **standalone** servers only. Cluster upgrades must use the procedure described in the [Clustering Guide](#)
- You can perform automated upgrades starting from PacketFence 11.0.0
- A backup and an export of your configuration are performed before doing upgrade

32.2. Full upgrade (for PacketFence version 11.0.0 only)

32.2.1. Preliminary steps

On PacketFence version 11.0.0, you need to install `packetfence-upgrade` package using following instructions:

RHEL / CentOS based systems only

```
yum install packetfence-upgrade --enablerepo=packetfence
```

Debian systems only

```
apt update  
apt install packetfence-upgrade
```

Then you can perform a full upgrade using following command:

```
/usr/local/pf/addons/full-upgrade/run-upgrade.sh
```

32.3. Full upgrade (for PacketFence versions 11.1.0 and later)

Run following script to perform a full upgrade:

```
/usr/local/pf/addons/upgrade/do-upgrade.sh
```

33. Additional Information

For more information, please consult the mailing archives or post your questions to it. For details, see:

- packetfence-announce@lists.sourceforge.net: Public announcements (new releases, security warnings etc.) regarding PacketFence
- packetfence-devel@lists.sourceforge.net: Discussion of PacketFence development
- packetfence-users@lists.sourceforge.net: User and usage discussions

34. Commercial Support and Contact Information

For any questions or comments, do not hesitate to contact us by writing an email to: support@inverse.ca.

Inverse (<https://inverse.ca>) offers professional services around PacketFence to help organizations deploy the solution, customize, migrate versions or from another system, performance tuning or aligning with best practices.

Hourly rates or support packages are offered to best suit your needs.

Please visit <https://inverse.ca/> for details.

35. GNU Free Documentation License

Please refer to <http://www.gnu.org/licenses/fdl-1.2.txt> for the full license.

36. Appendix

Appendix A: Administration Tools

36.A.1. pfcmd

`pfcmd` is the command line interface to most PacketFence functionalities.

When executed without any arguments `pfcmd` returns a basic help message with all main options:

Usage:

```
pfcmd <command> [options]
```

Commands

cache	manage the cache subsystem
checkup	perform a sanity checkup and report any problems
class	view security event classes
configreload	reload the configuration
connectionprofileconfig	query/modify connection profile
configuration parameters	
fingerbank	Fingerbank related commands
fixpermissions	fix permissions on pf tree
floatingnetworkdeviceconfig	query/modify floating network devices
configuration parameters	
generatedomainconfig	generate the domain configuration
generatemariadbconfig	generate the MariaDB configuration
generatemonitconfig	generate the monit configuration
generatesyslogconfig	generate the syslog configuration
help	show help for pfcmd commands
import	bulk import of information into the database
database	
ipmachistory	IP/MAC history
locationhistorymac	Switch/Port history
locationhistoryswitch	Switch/Port history
networkconfig	query/modify network configuration
parameters	
node	manipulate node entries
pfconfig	interact with pfconfig
pfcron	run pfcron tasks
pfqueue	query/modify pfqueue tasks and counters
reload	rebuild fingerprint or security events
tables without restart	
service	start/stop/restart and get PF daemon status
schedule	Nessus scan scheduling
switchconfig	query/modify switches.conf configuration
parameters	
version	output version information
security_event	manipulate security events
security_eventconfig	query/modify security_events.conf
configuration parameters	
tenant	manipulate tenants

Please view "pfcmd help <command>" for details on each option

The node view option shows all information contained in the node database table for a specified MAC address

```
# /usr/local/pf/bin/pfcmd node view 52:54:00:12:35:02
mac|pid|detect_date|regdate|unregdate|lastskip|status|user_agent|computername|notes|last_arp|last_dhcp|switch|port|vlan|dhcp_fingerprint
52:54:00:12:35:02|1|2008-10-23 17:32:16||||unreg||||2008-10-23 21:12:21|||||
```

Appendix B: Restoring a Percona XtraBackup or Mariabackup dump

If you enabled Percona XtraBackup or Mariabackup for your nightly backup, you can use the following instructions to restore it. In this example we will be restoring `/root/backup/packetfence-db-dump-innobackup-2016-12-20_00h31.xbstream.gz`

First, create a restore directory, move the backup to it and gunzip the backup:

```
# mkdir /root/backup/restore
# cd /root/backup/restore
# cp ../packetfence-db-dump-innobackup-2016-12-20_00h31.xbstream.gz .
# gunzip packetfence-db-dump-innobackup-2016-12-20_00h31.xbstream.gz
```

Then extract the xbstream data (for XtraBackup):

```
# xbstream -x < packetfence-db-dump-innobackup-2016-12-20_00h31.xbstream
```

Then extract the xbstream data (for Mariabackup):

```
# mbstream -x < packetfence-db-dump-innobackup-2016-12-20_00h31.xbstream
```

Once done, you should have a lot of files that were extracted in the restore dir. Now, lets place the xbstream back in the backup directory

```
# mv packetfence-db-dump-innobackup-2016-12-20_00h31.xbstream ../
```

Next, apply the innodb log (for XtraBackup):

```
# innobackupex --apply-log ./
```

Next, apply the innodb log (for Mariabackup):

```
# mariabackup --prepare --target-dir=./
```

We will now stop MariaDB, move the existing data directory and replace it by the data that was

extracted:

NOTE | Make sure you adjust the commands above to your environment.

For XtraBackup:

```
# service packetfence-mariadb stop
# mv /var/lib/mysql /var/lib/mysql.bak
# mkdir /var/lib/mysql
# mv * /var/lib/mysql
# chown -R mysql: /var/lib/mysql
# service packetfence-mariadb start
```

For Mariabackup:

```
# service packetfence-mariadb stop
# mv /var/lib/mysql /var/lib/mysql.bak
# mkdir /var/lib/mysql
# mariabackup --innobackupex --defaults
-file=/usr/local/pf/var/conf/mariadb.conf --move-back --force-non-empty
-directories ./
# chown -R mysql: /var/lib/mysql
# service packetfence-mariadb start
```

Should the service fail to start, make sure you look into the MariaDB error logs.

Appendix C: How to restore a standalone PacketFence server ?

Starting from PacketFence 11.0.0, you can use [the export/import mechanism](#).