



# Installation Guide

for PacketFence version 8.1.0

---

# Installation Guide

by Inverse Inc.

Version 8.1.0 - Jul 2018

Copyright © 2018 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".

9279VnJ

# Table of Contents

About this Guide .....	1
Other sources of information .....	1
Introduction .....	2
System Requirements .....	3
Assumptions .....	3
Minimum Hardware Requirements .....	3
Operating System Requirements .....	3
Installation .....	5
Installing PacketFence from the ZEN .....	5
Installing PacketFence on existing Linux .....	6
Getting Started .....	9
Going Through the Configurator .....	10
Connecting PacketFence to Microsoft Active Directory .....	10
Configuring Cisco Catalyst 2960 Switch .....	11
Adding the Switch to PacketFence .....	12
Configuring the Connection Profile .....	13
Configuring Microsoft Windows Supplciant .....	13
Testing .....	13
Enabling the Captive Portal .....	15
Creating Authentication Source for Guests .....	15
Configure switchport for Web Authentication .....	15
Adjust Switch Configuration in PacketFence .....	16
Enabling Portal on Management Interface .....	16
Configuring the Connection Profile .....	17
Testing .....	17
Adding SMS Authentication for Guests .....	18
Authentication Sources .....	18
Alerting .....	19
Adding SMS Authentication Source .....	19
Configuring the Connection Profile .....	20
Testing .....	20
Introduction to Role-based Access Control .....	21
Adding Roles .....	21
Using the Employee Role .....	22
Using the Corporate_Machine Role .....	22
Supported Enforcement Modes .....	24
Technical Introduction to Inline Enforcement .....	24
Technical Introduction to Out-of-band Enforcement .....	25
Technical Introduction to Hybrid Enforcement .....	30
Technical Introduction to RADIUS Enforcement .....	31
Technical Introduction to DNS Enforcement .....	31
Adding Inline Enforcement to Existing Installation .....	33
Introduction .....	33
Preparing the Operating System .....	33
Adding Inline Interface .....	34
Network Devices .....	35
Adding Connection Profile for Inline .....	35
Testing the Inline Configuration .....	36
Advanced Inline Topics .....	36
Adding VLAN Enforcement to Existing Installation .....	37
Introduction .....	37

Adding the Registration, Isolation and Other Interface .....	38
Network Devices .....	39
Adding Connection Profile for Registration .....	41
Troubleshooting PacketFence .....	42
RADIUS Audit Log .....	42
Log files .....	42
RADIUS Debugging .....	42
Authentication Mechanisms .....	44
Microsoft Active Directory (AD) .....	44
OAuth2 Authentication .....	49
Eduroam .....	53
SAML Authentication .....	53
Billing Engine .....	55
External API Authentication .....	69
Advanced Portal Configuration .....	71
Portal Modules .....	71
Portal Surveys .....	79
Devices Registration .....	84
Passthroughs .....	84
Proxy Interception .....	85
Parked Devices .....	86
Advanced Access Configuration .....	88
Connection Profiles .....	88
VLAN Filter Definition .....	90
RADIUS Filter Definition .....	95
Advanced RADIUS Configuration .....	97
Local Authentication .....	97
Authentication against Active Directory (AD) .....	97
EAP Authentication against OpenLDAP .....	97
EAP Guest Authentication on Email, Sponsor and SMS Registration .....	98
EAP Local User Authentication .....	100
Limit Brute Force EAP Authentication .....	100
Testing .....	100
RADIUS Accounting .....	101
Fingerbank Integration .....	103
Onboarding .....	103
Update Fingerbank Database .....	103
Submit Unknown Data .....	104
Upstream Interrogation .....	104
Local Entries .....	104
Settings .....	104
Intrusion Detection System Integration .....	105
Regex Syslog Parser .....	105
Suricata IDS .....	106
Security Onion .....	108
ERSPAN .....	110
StreamScan Compromise Detection System (CDS) .....	111
Firewall SSO Integration .....	115
Barracuda .....	115
Checkpoint .....	116
FortiGate .....	120
iBoss .....	123
JSON-RPC .....	123
Palo Alto .....	124

Performing Compliance Checks .....	129
Installation .....	129
Configuration .....	130
Integrating Provisioning Agents .....	136
PacketFence Apple, Android and Windows Wireless Provisioning .....	136
MobileIron .....	140
OPSWAT .....	146
SentinelOne .....	152
Symantec SEPM .....	155
PKI Integration .....	162
Microsoft PKI .....	162
PacketFence PKI .....	173
Best Practices .....	186
RHEL7 systemd early swapoff bug mitigation .....	186
IPTables .....	187
Log Rotations .....	187
Performance Optimizations .....	188
NTLM Authentication Caching .....	188
SNMP Traps Limit .....	192
MySQL optimizations .....	192
Captive Portal Optimizations .....	196
Dashboard Optimizations (statistics collection) .....	197
Troubleshooting .....	197
Advanced Network Topics .....	199
Floating Network Devices .....	199
Production DHCP access .....	200
Routed Networks .....	202
Network Devices Definition (switches.conf) .....	205
More on VoIP Integration .....	209
DHCP Option 82 .....	210
Additional Integration .....	211
DHCP Remote Sensor .....	211
Active Directory Integration .....	212
Switch Login Access .....	217
Cisco Mobility Services Engine (MSE) .....	217
Advanced Topics .....	222
Custom Reports .....	222
Admin Access .....	224
Content-Security-Policy (CSP) .....	225
Additional Information .....	226
Commercial Support and Contact Information .....	227
GNU Free Documentation License .....	228
A. Administration Tools .....	229
pfcmd .....	229
B. Restoring a Percona XtraBackup dump .....	231
Optional components .....	233
Blocking malicious activities with violations .....	233
Guests Management .....	239

# 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/>

## Other sources of information

---

The following documents are included in the package and release tarballs.

<i>Network Devices Configuration Guide</i> (pdf)	Covers switches, WiFi controllers and access points configuration.
<i>Developer's Guide</i> (pdf)	Covers API, captive portal customization, application code customizations and instructions for supporting new equipment.
<b>CREDITS</b>	This is, at least, a partial file of PacketFence contributors.
<b>NEWS.asciidoc</b>	Covers noteworthy features, improvements and bug fixes by release.
<b>UPGRADE.asciidoc</b>	Covers compatibility related changes, manual instructions and general notes about upgrading.

# Introduction

---

PacketFence is a fully supported, trusted, Free and Open Source network access control (NAC) system. Boasting 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>.

# System Requirements

---

## Assumptions

---

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

- database server (MySQL or 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.

## Minimum Hardware Requirements

---

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

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

## Operating System Requirements

---

PacketFence supports the following operating systems on the x86\_64 architecture:

- Red Hat Enterprise Linux 7.x Server



## Chapter 3

- Community ENTERprise Operating System (CentOS) 7.x
- Debian 8.0 (Jessie)

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 Fedora and Gentoo are known to work but this document does not cover them.

# 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.

## 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.

### Virtual Machine

This setup has been tested using VMWare ESXi, Fusion and Workstation products with 8 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).

The virtual appliance passwords are:

Management (SSH/Console) and MySQL

- Login: root
- Password: [p@ck3tf3nc3](#)

Captive Portal / 802.1X Registration User

- Login: demouser
- Password: demouser

## 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.

## Import to VMWare Player/Workstation for Linux

Newer version of VMWare Player handles the VLAN trunking a lot better. Having that said, 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.

## Installing PacketFence on existing Linux

---

PacketFence provides packages repository for RHEL / CentOS as well as package repository for 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

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

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 execute the following commands:

```
update-rc.d -f apparmor stop
update-rc.d -f apparmor teardown
update-rc.d -f apparmor remove
```

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

## Red Hat-based Systems



### Note

Applies to CentOS and Scientific Linux but only the x86\_64 architecture is supported.

Explicitly instruct NetworkManager to never interact with your DNS configuration ([Source](#)):

```
echo "[main]
dns=none" > /etc/NetworkManager/conf.d/99-no-dns.conf
service NetworkManager restart
```

## RHEL 7.x



### Note

These are extra steps are required for RHEL 7 systems only, excluding derivatives such as CentOS or Scientific Linux.

RedHat Enterprise Linux users need to take an additional setup step. If you are not using the RHN Subscription Management from RedHat you need to enable the optional channel by running the following as root:

```
subscription-manager repos --enable rhel-7-server-optional-rpms
```

## Software Installation

### RHEL / CentOS

In order to use the PacketFence repository:

```
yum localinstall http://packetfence.org/downloads/PackageFence/RHEL7/`uname -i`/
RPMS/packetfence-release-1.2-6.e17.centos.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

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 jessie jessie' \  
> /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:

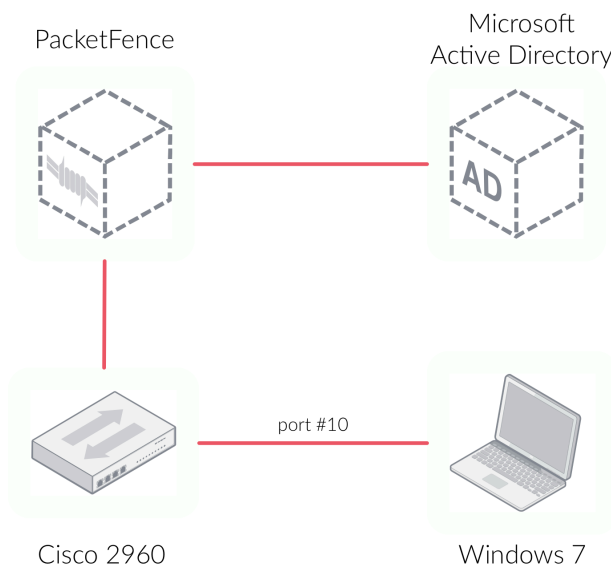
```
sudo apt-key adv --keyserver keys.gnupg.net \  
--recv-key 0xFE9E84327B18FF82B0378B6719CDA6A9810273C4  
sudo apt-get update  
sudo apt-get install packetfence
```

# 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.

## Going Through the Configurator

---

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

- Step 1 - Enforcement, choose *RADIUS enforcement* - this will make PacketFence act as simple RADIUS server
- Step 2 - Networks, make sure you define only one interface with the Management type. That network interface will be the one for 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 3 - Database - provide the required information to properly create the PacketFence database
- Step 4 - PacketFence - provide your domain name, DHCP servers list and other required information
- Step 5 - Administration - provide the PacketFence's admin username and password to be used
- Step 6 - 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 7 - Confirmation - start all services.

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/admin](https://@ip_of_packetfence:1443/admin)

Open that link and log in using the username/password specified in Step 5.

## 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 *Add 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 *Save and 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 *Add source*→*Internal*→*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.

Next, 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.

## 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.

### 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
```

### 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

```

## 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.

## 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 *Add 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.

## 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 on *Add profile*. Specify the following information:

- Profile Name: 8021x
- 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 *Save* to save all configuration changes.

## 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.

## 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 (+) button to see the RADIUS exchanges. If the 802.1X authentication is successful, you should have *Accept* as an *Auth Status*.

# 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.

## 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 *Add Source*→*External*→*Null*. As *Name* and *Description*, specify *null-source*. 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 *Save* to save the new authentication source.

## 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
```

## 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 *Save* to save all configuration changes.

## 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

## 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 *Add Profile*. Specify the following information:

- Profile Name: guest
- Filters: If any of the following conditions are met:
  - Connection Type: WIRED\_MAC\_AUTH
- Sources: null-source

Click on *Save* to save all configuration changes.

## 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.

# 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
- Alerting

## 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
- Email
- External HTTP API
- Facebook (OAuth 2)
- Github (OAuth 2)
- Google (OAuth 2)
- Instagram (OAuth 2)
- Kerberos
- Kickbox
- LDAP
- LinkedIn (OAuth 2)
- Null
- OpenID Connect (OAuth 2)
- Pinterest (OAuth 2)
- RADIUS
- SMS
- Sponsored Email
- Twitter (OAuth 2)
- Windows Live (OAuth 2)

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 `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.

## 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.

## 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 *Add source→External→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 *Save* to save the new authentication source.

## 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.

## 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.

# 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 another 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.

## 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 two algorithms: one based on a hash of the username (default one) and the other one based on a round-robin (last registered device +1).

**Configuration → Policies and Access Control → Roles**, click on *Add 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.

## Using the Employee Role

---

From the **Configuration** → **Policies and Access Control** → **Authentication Sources**, we select **Add 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**:
  - Set role employee
  - Set unregistration date January 1st, 2020

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 an unregistration date set to January 1st, 2020.

## 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**:
  - Set role corporate\_machine
  - Set unregistration date January 1st, 2020

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 httpasswd 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.

# 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 you how to properly configure PacketFence to use each enforcement mode.

## Technical Introduction to Inline Enforcement

---

### 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.

### Device Configuration

No special configuration is needed on the unmanageable device. That's the beauty of it. 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.

### 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.

## Limitations

Inline enforcement because of its nature has several limitations that one 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.

# Technical Introduction to Out-of-band Enforcement

---

## 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.

## 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.

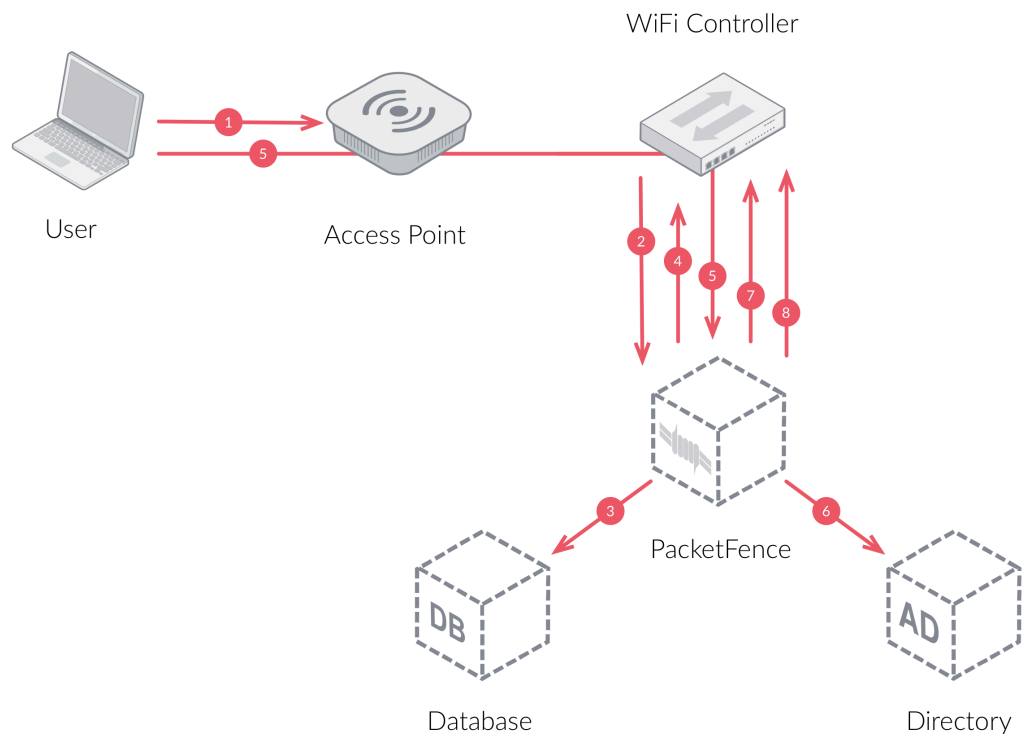
MAC Authentication is a new mechanism introduced by some switch vendor to handle the cases where a 802.1X supplicant does not exist. Different vendors have different names for it. Cisco calls it MAC Authentication Bypass (MAB), Juniper calls it MAC RADIUS, Extreme Networks calls it Netlogin, etc. After 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.

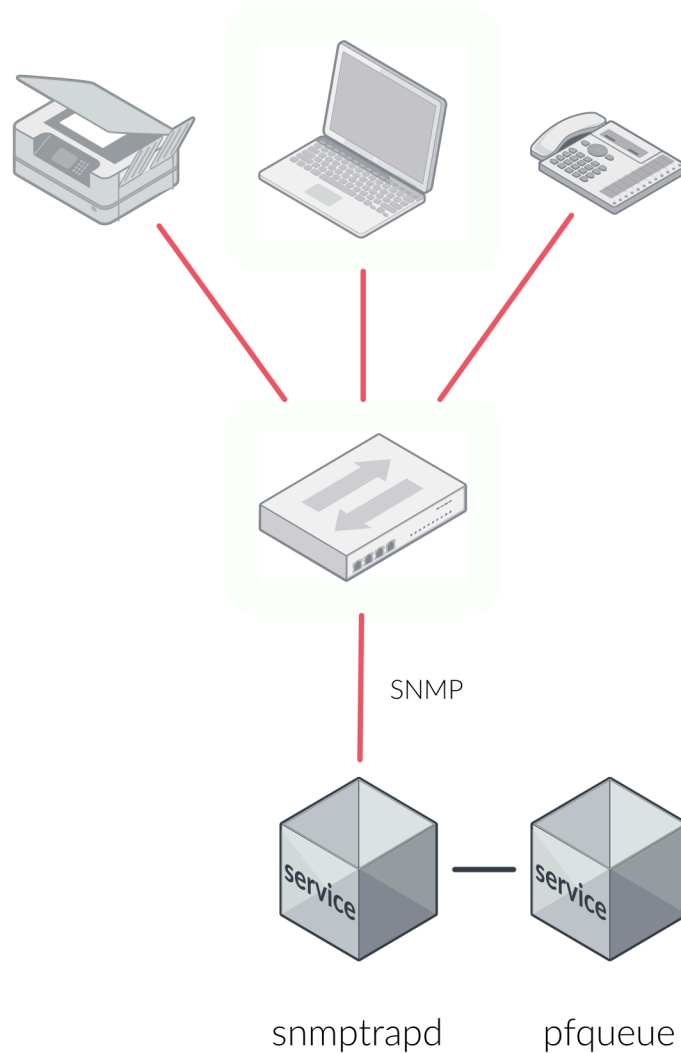
## Port-security and SNMP

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.

## 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 violations in a separate VLAN, an isolation VLAN needs also to be created.

### Link Changes (deprecated)

This is the most basic setup and it needs a third VLAN: the MAC detection VLAN. There should be nothing in this VLAN (no DHCP server) and it should not be routed anywhere; it is just a void VLAN.

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 MAC detection VLAN in which the device will send DHCP requests

(with no answer) 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 violations ?) 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 MAC detection 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 MAC detection 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.

# Technical Introduction to Hybrid Enforcement

---

## 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.

## 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 , where ALWAYS means that the device is always in inline mode, PORT specify the  
 PORT , MAC ifIndex of the port which will use inline enforcement, MAC a mac address that will  
 , SSID 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.

## Technical Introduction to RADIUS Enforcement

---

### 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*→*Interfaces*. When doing so, you must select **Other** as the type of interface. Moreover, you must select **radius** as an additional 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*.

## Technical Introduction to DNS Enforcement

---

### 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* → *Interfaces* 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 *Add 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.

# Adding Inline Enforcement to Existing Installation

---

## 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).

## 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

## 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 :

Network Name	Subnet	Gateway	PacketFence Address
ens160 Management Card	172.20.100.0/16	172.20.0.1	172.20.100.2

Network Name Card	Subnet	Gateway	PacketFence Address
ens192 Inline	192.168.2.0/24	192.168.2.1	192.168.2.1

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

## 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.

## 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*.



## 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.

## Advanced Inline Topics

---

### 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.

# Adding VLAN Enforcement to Existing Installation

---

## 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

VLAN ID	VLAN Name	PacketFence Address	DHCP	DNS
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

## 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`, `httpd.portal`, `pfdhcp`, `iptables`, `pfdhcplistener`, `pfdns` services.

## 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.

## 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.

### 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
```

### 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. The latest documentation is available here: [http://www.cisco.com/en/US/docs/switches/lan/catalyst2960/software/release/15.0\\_1\\_se/configuration/guide/sw8021x.html](http://www.cisco.com/en/US/docs/switches/lan/catalyst2960/software/release/15.0_1_se/configuration/guide/sw8021x.html)

## Save the Configuration

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

## 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*.

## 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.

# Troubleshooting PacketFence

---

## RADIUS Audit Log

---

TODO

## 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/`.

## 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/raddb -n auth
```

For the accounting radius process:

```
# radiusd -X -d /usr/local/pf/raddb -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:

- a. 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**
- b. 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.



# 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.

## 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 **Add Domain** and fill in the information about your domain.

**New Domain** ✕

Identifier ⓘ

Specify a unique identifier for your configuration.  
This doesn't have to be related to your domain

Workgroup ⓘ

DNS name of the domain ⓘ

The DNS name (FQDN) of the domain.

This server's name ⓘ

This server's name (account name) in your Active Directory.

Active Directory server ⓘ

The IP address or DNS name of your Active Directory server

DNS server ⓘ

The IP address of the DNS server for this domain.

Username ⓘ

Password ⓘ

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.
- **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.
- **Username** is the username that will be used for binding to the server. This account must be a domain administrator.
- **Password** is the password for the username defined above.



### 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.

## 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.

## 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**

**New Realm** ✕

Realm ⓘ

Realm Options

You can add options in the realm definition

Domain  ▼

The domain to use for the authentication in that realm

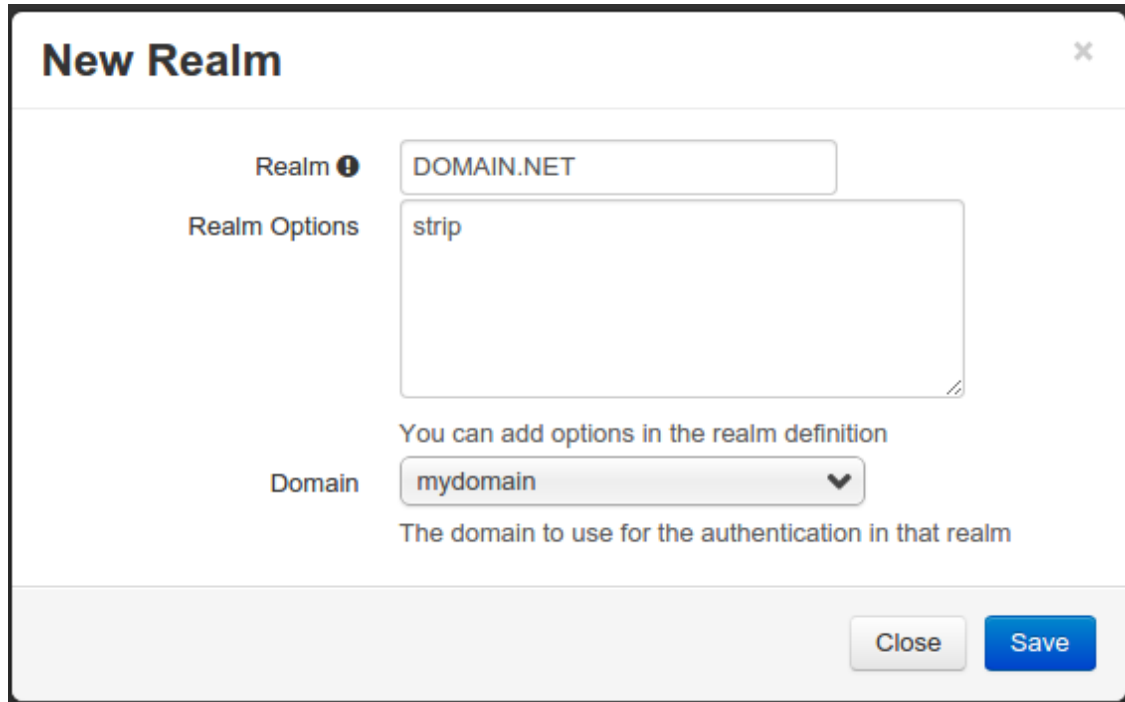
Next, restart PacketFence in **Status** → **Services**

## 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 and DOMAIN.

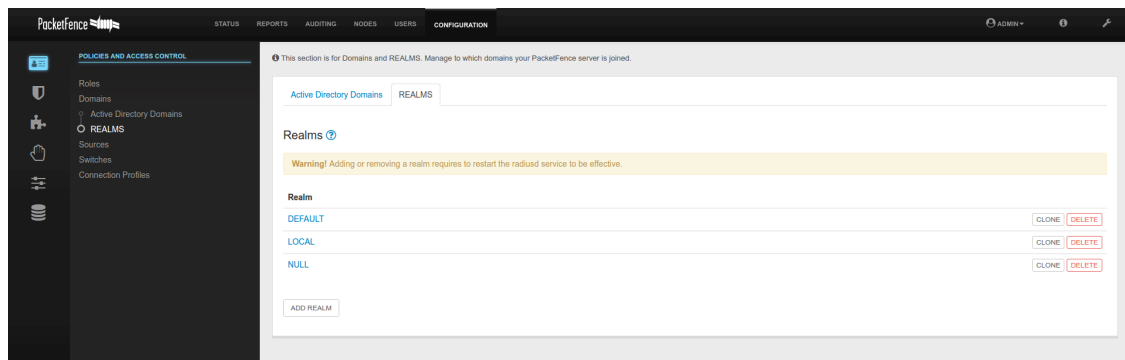


Where :

- **Realm** is either the DNS name (FQDN) of your domain or the workgroup
- **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

Now create the two other realms associated to your other domains.

You should now have the following realm configuration



## 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, Twitter, Instagram, Pinterest, 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).

## 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](https://YOUR_PORTAL_HOSTNAME/oauth2/callback). Of course, replace the hostname with the values from `general1.hostname` and `general1.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](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 and Pages**.

## 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](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](https://YOUR_PORTAL_HOSTNAME/oauth2/callback) and then save changes.

Also, add the following Authorized domains : `*.facebook.com`, `*.fbcdn.net`, `*.akamaihd.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](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 and Pages**.



### Caution

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

## 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>. When you create your App, make sure you specify the following as the Callback URL [https://YOUR\\_PORTAL\\_HOSTNAME/oauth2/callback](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 and Pages**.

## Instagram

To use Instagram, you also need an API code and a secret key. To get one, go here: <https://www.instagram.com/developer/clients/manage/>. When you create your App, make sure you specify the following as the Website URL: [https://YOUR\\_PORTAL\\_HOSTNAME/oauth2/callback](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 Instagram 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 Instagram as a **Source** from your connection profile definition, available from **Configuration → Policies and Access Control → Connection Profiles and Pages**.

## 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](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 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 and Pages**.

Also, LinkedIn requires a *state* parameter for the authorization URL. If you modify it, make sure to add it at the end of your URL.

## 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](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 and Pages**.

## Pinterest

To use Pinterest, you also need an API code and a secret key. To get one, go here: <https://developers.pinterest.com/apps>. When you create your App, make sure you specify the following as the Redirect URL: [https://YOUR\\_PORTAL\\_HOSTNAME/oauth2/callback](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 Pinterest 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 Pinterest as a **Source** from your connection profile definition, available from **Configuration → Policies and Access Control → Connection Profiles and Pages**.

## Twitter

To use Twitter, you also need an API code and a secret key which Twitter calls *consumer key* and *consumer secret*. Obtain this information by creating an new application from your [Twitter Apps Management page](#). When you create your App, make sure you specify the following as the *Callback URL* [https://YOUR\\_PORTAL\\_HOSTNAME/oauth2/callback](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 Twitter 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 Twitter as a **Source** from your connection profile definition, available from **Configuration → Policies and Access Control → Connection Profiles and Pages**.

## 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://account.live.com/developers/applications>. When you create your App, make sure you specify the following as the Callback URL [https://YOUR\\_PORTAL\\_HOSTNAME/oauth2/callback](https://YOUR_PORTAL_HOSTNAME/oauth2/callback) replacing the hostname with the values from `general.hostname` and `general.domain`, and check **Live SDK support**.

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).

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

## 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.

In order for PacketFence to allow Eduroam authentication, you need to configure an exclusive Eduroam source in **Configuration → Authentication Sources** and restart the RADIUS and IPTables services.

## 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.

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.

User Sources > New Source

### New Source SAML

Name

Description

Service Provider entity ID

Path to Service Provider key (x509)

Path to Service Provider cert (x509)

Path to Identity Provider metadata

Identity Provider entity ID

Path to Identity Provider cert (x509)

Path to Identity Provider CA cert (x509)   
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

Authorization source  The source to use for authorization (rule matching)

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.

## 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 *Passthrough* and add the Identity Provider domain name to the *Passthroughs* list.

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

## 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 four payment gateways: Authorize.net, Mirapay, Paypal and Stripe.

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

- Billing source(s)

- Billing tier(s)

## 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*.

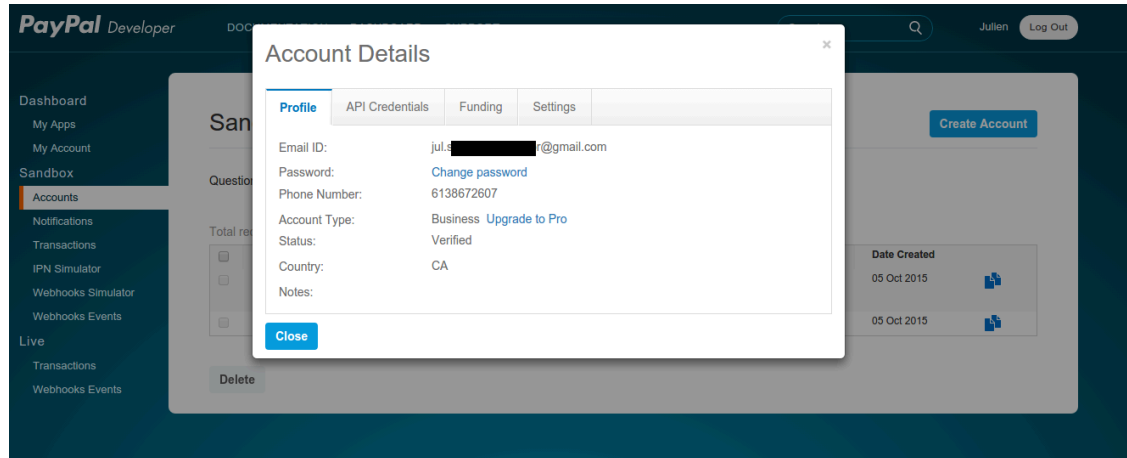
Afterwards, go back into accounts, and expand the business account, then click *Profile*

The screenshot shows the PayPal Developer 'Sandbox Test Accounts' page. The page has a dark blue header with 'PayPal Developer' and navigation links for 'DOCUMENTATION', 'DASHBOARD', and 'SUPPORT'. A search bar and user information 'Julien Log Out' are also present. The left sidebar contains a menu with 'Dashboard', 'My Apps', 'My Account', 'Sandbox', 'Accounts', 'Notifications', 'Transactions', 'IPN Simulator', 'Webhooks Simulator', 'Webhooks Events', 'Live', 'Transactions', and 'Webhooks Events'. The main content area is titled 'Sandbox Test Accounts' and includes a 'Create Account' button. Below the title, there is a link to the 'Testing Guide' and an 'FAQ'. A table shows two records:

Email Address	Type	Country	Date Created
Jul. [redacted]@gmail.com	BUSINESS	CA	05 Oct 2015
Jul. [redacted]@gmail.com	PERSONAL	CA	05 Oct 2015

The BUSINESS account row is expanded to show a 'Profile' link. A 'Delete' button is located at the bottom left of the table.

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](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.

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

Overview
Add Funds
Withdraw
History
Resolution Centre
Profile

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

#### Auto Return for Website Payments

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](#)

Auto Return:  On  
 Off

**Return URL:** Enter the URL that will be used to redirect your customers upon payment completion. This URL must meet the guidelines detailed below. [Learn More](#)

Return URL:

**Return URL Requirements:** The following items are required in order to set up Auto Return.

- 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.
- You must provide verbiage on the page displayed by the Return URL that explains that payment transaction details will be emailed to the buyer.
- 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](http://www.sandbox.paypal.com/ca) to view details of this transaction.

---

#### Payment Data Transfer (optional)

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.

Payment Data Transfer:  On  
 Off

Identity Token: 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 it's 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*.



User Sources > New Source

### New Source Paypal

Name

Description

Identity token

Cert id

Payment type 
The type of transactions this source will do (donations or sales).

Authorized domains 
Comma separated list of domains that will be resolve with the correct IP addresses.

Email address 
The email address associated to your paypal account.

Cert file 
The path to the certificate you submitted to Paypal.

Key file 
The path to the associated key of the certificate you submitted to Paypal.

Paypal cert file 
The path to the Paypal certificate you downloaded.

Currency

Test mode

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

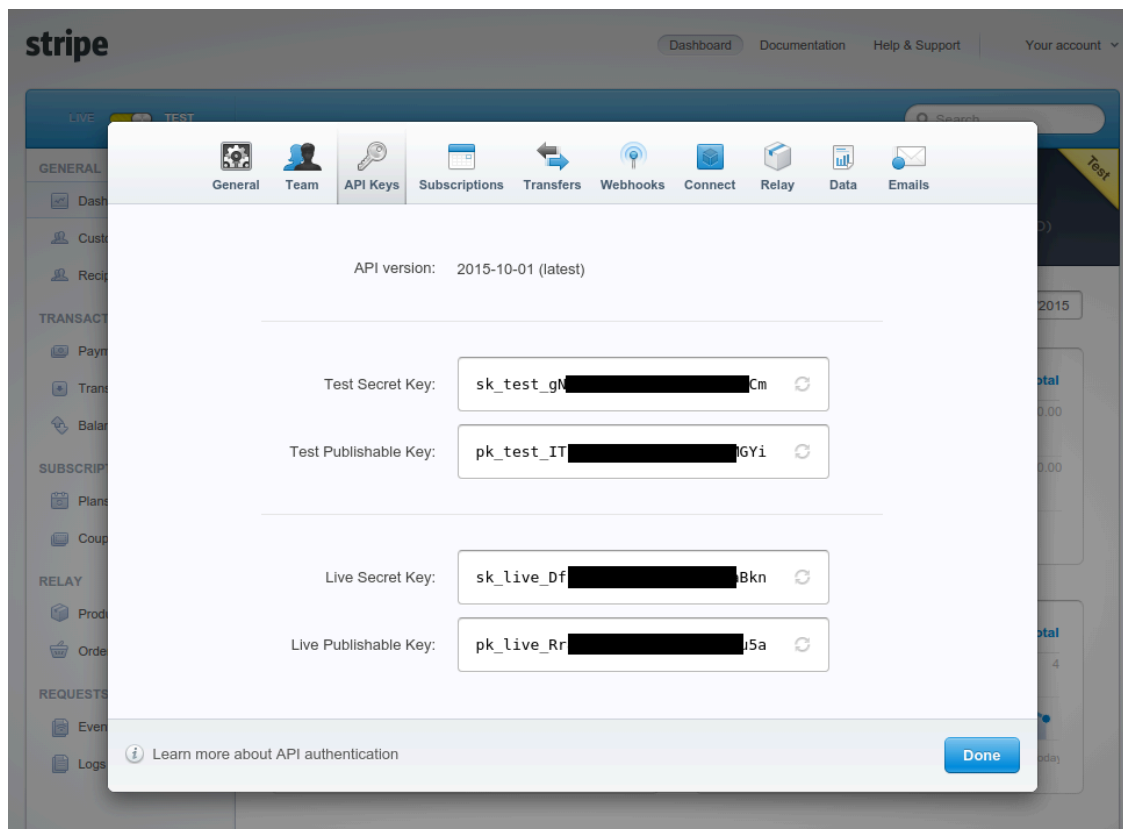
## Stripe

### 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*

User Sources > New Source

### New Source Stripe

Name

Description

Secret key

Publishable key

Style

The type of payment the user will make. Charge is a one time fee, subscription will be arecurring fee.

Currency

Test mode

[Save](#)

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

## Authorize.net

### Creating an account

First go on <https://account.authorize.net> to signup for a merchant account or <http://developer.authorize.net/> for a sandbox account.

After you created your account you will be shown your *API login ID* and *Transaction key*. Note both of these information for usage in the PacketFence configuration.

Then login into your new account.

Then under *Account* click *Settings*.

On the settings page in the section *Security settings*, click *MD5-Hash*

Now enter a secret that will be shared between authorize.net and PacketFence.

The screenshot shows the Authorize.Net user interface. At the top, there are navigation links for FEEDBACK, CONTACT US, HELP, and LOG OUT. The main header displays the Authorize.Net logo and a welcome message: "Welcome: TestFirstName TestLastName". Below the header is a navigation bar with tabs for HOME, TOOLS, REPORTS, SEARCH, and ACCOUNT. On the left side, there is a "Settings" menu with options: Billing Information, Statements, User Administration, User Profile, and Digital Payment Solutions. The main content area is titled "MD5 Hash" and includes a "Help" link. The text reads: "If you would like your transaction responses to be encrypted by the Payment Gateway, enter and confirm a MD5 Hash Value below. For more information about the MD5 Hash, please refer to [Reference & User Guides](#)." Below this, there are two input fields for "New Hash Value" and "Confirm Hash Value", both containing the text "useStrongerSecret". A "Submit" button is located at the bottom of the form.

### PacketFence configuration

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

The screenshot shows the "New Source" configuration page in the PacketFence administration interface. The breadcrumb trail is "User Sources > New Source". The page title is "New Source" with "AuthorizeNet" as a subtitle. The form contains the following fields and options:

- Name:** authorize-test
- Description:** authorize.net
- Api login id:** 65 [redacted] 37
- Transaction key:** f8 [redacted] df
- MD5 hash:** useStrongerSecret
- Authorized domains:** \*.authorize.net

Below the "Authorized domains" field, there is a note: "Comma separated list of domains that will be resolve with the correct IP addresses." Below this, there is a "Currency" dropdown menu set to "CAD" and a "Test mode" checkbox which is checked. At the bottom of the form is a blue "Save" button.

Where :

- **API login ID** is the one you got earlier while creating your account.
- **Transaction key** is the one you got earlier while creating your account.
- **MD5 hash** the one you configured in your Authorize.net account.
- **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

## Mirapay

To be contributed...

## 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.

**New billing tier** ×

Billing tier ⓘ

Name ⓘ

Description ⓘ 

The short name of the billing tier  
This tier will grant you basic access to the network for a duration of 24 hours.

A description of the billing tier

Price ⓘ

The price that will be charged to the customer

Role ⓘ

The target role of the devices that use this tier.

Access duration ⓘ

The access duration of the devices that use this tier.

Use time balance

Check this box to have the access duration be a real time usage.  
This requires a working accounting configuration.

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*.

## 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.

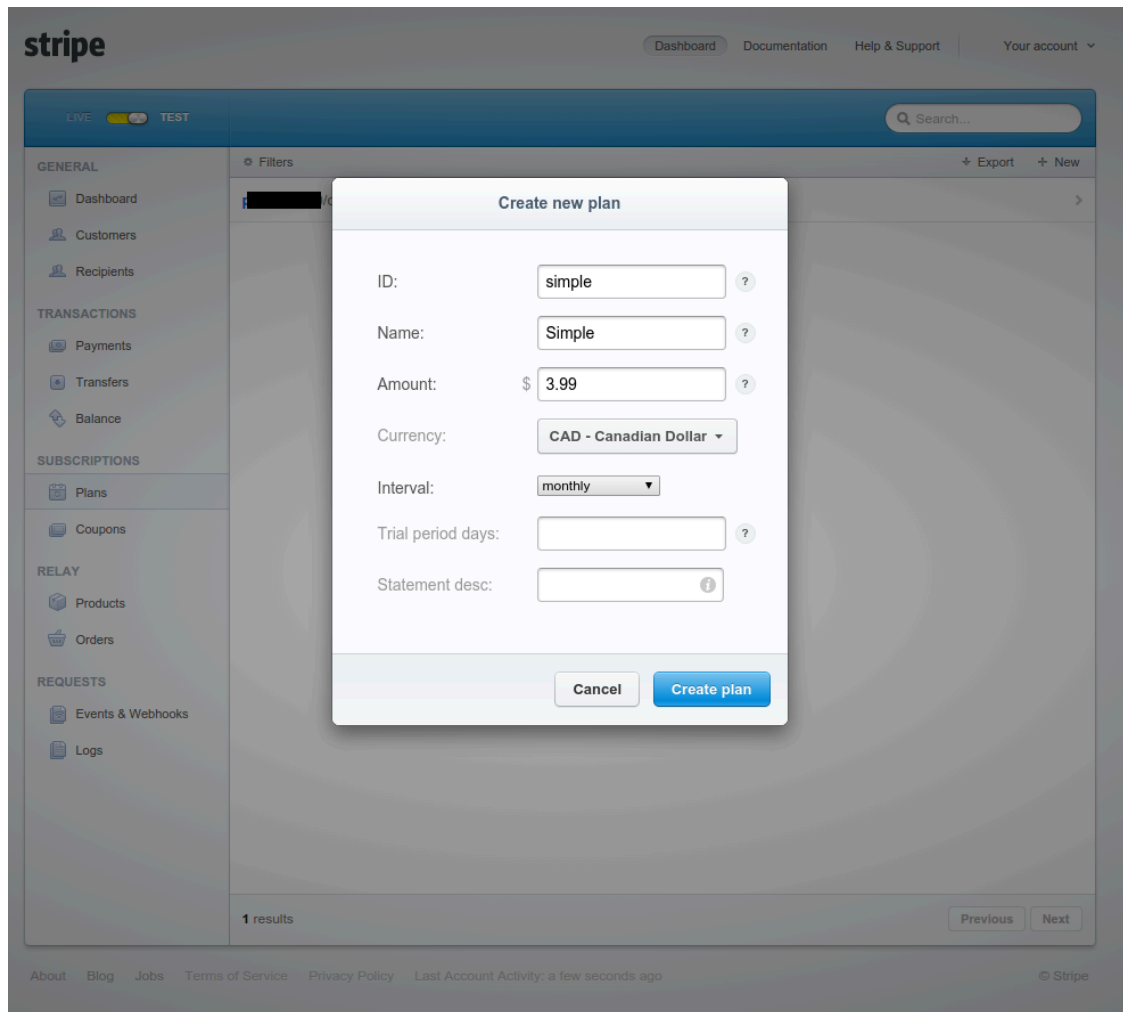
```
[simple]
name=Simple network access
description=Click here if you are poor
price=3.99
role=guest
access_duration=10Y
use_time_balance=disabled
```

```
[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.

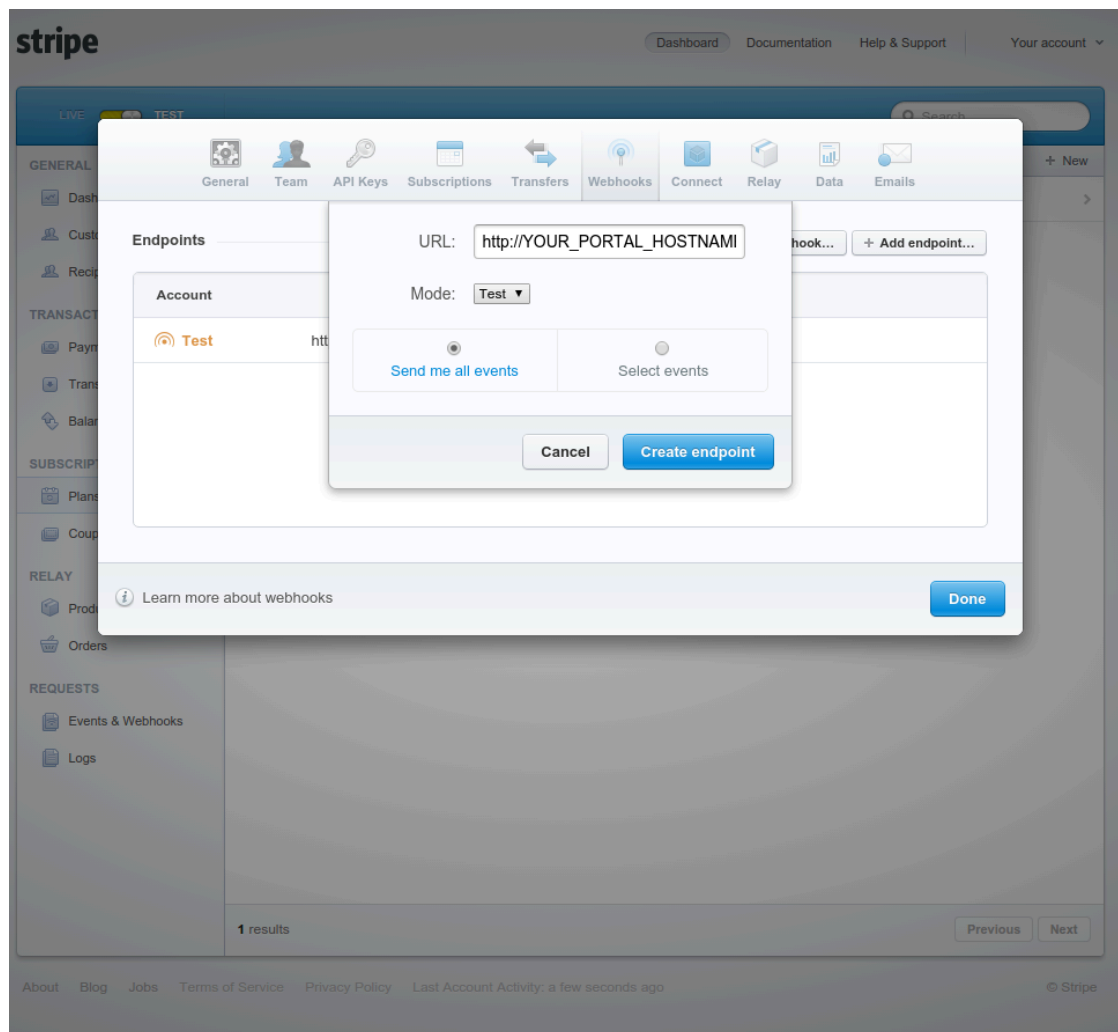
Updates are sent using HTTP requests on a public IP.

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](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.

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

## 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.

### 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"}
```

### 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.

## 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.

# Advanced Portal Configuration

---

## 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.

## 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.


Portal Modules >

### New portal module


Identifier 

Description 

The description that will be displayed to users

Sources  

The sources to use in the module. If no sources are specified, all the sources on the Portal Profile will be used

PID field  


Which field should be used as the PID.

Mandatory fields

The additional fields that should be required for registration

Require AUP

Require the user to accept the AUP

Signup template 

The template to use for the signup

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.

Portal Modules >

### New portal module

Identifier ⓘ

Description ⓘ

The description that will be displayed to users

Sources  ▼

The sources to use in the module. If no sources are specified, all the sources on the Portal Profile will be used

PID field  ▼

Which field should be used as the PID.

Mandatory fields

The additional fields that should be required for registration

Require AUP

Require the user to accept the AUP

Signup template ⓘ

The template to use for the signup

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

Portal Modules >

## New portal module

Identifier ⓘ

Description ⓘ

The description that will be displayed to users

Modules

1	<input type="text" value="default_oauth_policy"/>
2	<input type="text" value="sms"/>

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.

Portal Modules >

## New portal module

Identifier ⓘ

Description ⓘ

The description that will be displayed to users

Show first module when none is selected

Template

The template to use to display the choices

Modules

1	<input type="text" value="default_login_policy"/>
2	<input type="text" value="secure_boarding"/>

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>
```

Portal Modules >

## New portal module

Identifier **i**

Description **i**

The description that will be displayed to users

Message **i**

The message that will be displayed to the user. Use with caution as the HTML contained in this field will NOT be es

Template

The template to use to display the message

Skipable

Whether or not, this message can be skipped

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.

Portal Modules > my\_first\_root\_module

## Root my\_first\_root\_module

Description **i**

The description that will be displayed to users

Modules

1	<input type="text" value="default_registration_policy"/>
2	<input type="text" value="hello_world"/>

## 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.

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.

## 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.

## 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.

## 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`.

## Portal Surveys

---

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

## 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 MySQL CLI as root and execute the following:

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

## 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 -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 -Mpf::person -MData::Dumper -e 'print Dumper(person_view("admin"))'`
- **ip**: The IP address of the client

## 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 form is titled 'New portal module' and is located under the 'Portal Modules' section. It contains several input fields and a dropdown menu:

- Identifier**: A text input field containing 'survey1'.
- Description**: A text input field containing 'My first survey'. Below this field is a note: 'The description that will be displayed to users'.
- Domain**: A dropdown menu with 'survey1' selected. Below this field is a note: 'The survey to use in this portal module. Surveys are defined in survey.conf'.
- Template**: A text input field containing 'survey.html'. Below this field is a note: 'The template to use to display the message'.
- actions**: A section with a gear icon and the text 'No actions selected [Add one.](#) If none are specified, the default ones of the module will be used.'

At the bottom of the form is a 'SAVE' button.

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

Portal Modules >

### New portal module

Identifier \*

Description \*

The description that will be displayed to users

modules 1  ⊖ ⊕

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*.

## 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 *Custom 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 configreload hard`

Refer to the *Custom Reports* section of this document for advanced configuration.

## 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.

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



## 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.

The section **Configuration** → → **Advanced Access Configuration** → **Device Registration** is where to configure it. The portal page can be accessed by the following URL: [https://YOUR\\_PORTAL\\_HOSTNAME/device-registration](https://YOUR_PORTAL_HOSTNAME/device-registration) This URL is accessible from within the network, in any VLAN that can reach the PacketFence server.

As from PacketFence 7.3 and onwards you need to create a device registration profile and associate it with the connection profile.

This will allow you to configure who is able to register which type of device. For instance a person connecting from the public WiFi can only register an Android device but a person from the internal network can register Windows and MacOS computers.



### 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.

## 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.

## 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.

## 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`.

## Proxy Interception

---

PacketFence enables you to intercept proxy requests and forward them to the captive portal. It only works one 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.

## 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 violation **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 violation **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.

## Violation 1300003

This violation 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 violations 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*

# Advanced Access Configuration

---

## 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`:

```
[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
- 
- **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**

Examples

- `last_switch =~ "^JAMES" && extended.mse_inout.bob == "bobby"`

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`).

## 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 match IF an authentication source is configured for it.

## 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.

These rules are available in different scopes:

```
ViolationRole
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)
```

Below are several common examples of VLAN filters usage.

EXAMPLE: prevent 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 as a registered device

```
[category]
filter = node_info.category
operator = is
value = default
```

```
[ssid]
filter = ssid
operator = is
value = SECURE
```

```
[time]
filter = time
operator = is
value = wd {Mon Tue Wed Thu Fri} hr {11am-2pm}
```

```
[1:category&ssid&time]
scope = RegisteredRole
role = nointernet
```

EXAMPLE: create a violation if the SSID is OPEN and the owner is igmout

```
[igmout]
filter = owner.pid
operator = is
value = igmout
```

```
[open]
filter = ssid
operator = is
value = OPEN
```

```
[2:igmout&ssid]
scope = RegisteredRole
action = trigger_violation
action_param = mac = $mac, tid = 1100012, type = INTERNAL
```

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

```
[igmout]
filter = username
operator = is
value = igmout
```

```
[secure]
filter = ssid
operator = is
value = SECURE
```

```
[3:igmout&secure]
scope = AutoRegister
role = staff
```



```
[4:igmout&secure]
scope = NodeInfoForAutoReg
role = staff
```

EXAMPLE: autoregister all Windows devices and assign them the role default.

```
[Windows_AutoReg]
filter = fingerbank_info.device_hierarchy_names
operator = includes
value = Windows
```

```
[AUTOREG_WINDOWS:Windows_AutoReg]
scope = AutoRegister
role = default
```

```
[WINDOWS_ROLE:Windows_devices]
scope = NodeInfoForAutoReg
role = default
```

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

```
[MAC]
filter = node_info.mac
operator = match
value = aa:bb:cc:dd:ee:ff
```

```
[SSID_Open]
filter = ssid
operator = is
value = Open
```

```
[REJECT:MAC&SSID_Open]
scope = RegistrationRole
role = REJECT
```

EXAMPLE: autoregister devices using dot1x/EAP over wired or wireless and that are not pending

```
[not_pending]
filter = node_info.status
operator = is_not
value = pending
```

```
[wireless_dot1x]
filter = connection_type
operator = is
value = Wireless-802.11-EAP
```

```
[wired_dot1x]
filter = connection_type
operator = is
value = Ethernet-EAP
```

```
[autoreg_wireless_dot1x:wireless_dot1x&not_pending]
scope = AutoRegister
role = default
```

```
[autoreg_wired_dot1x:wired_dot1x&not_pending]
scope = AutoRegister
role = default
```

EXAMPLE: auto set Avaya and Polycom as phones by matching vendor MAC, and set to default role

```
[Polycom]
filter = node_info.mac
operator = regex
value = ^(00:04.f2).*
```

```
[avaya]
filter = node_info.mac
operator = regex
value = ^(00:04:0d|00:1b:4f|00:e0:07|04:8a:15|24:d9:21|2c:f4:c5|34:75:c7|3c:3a:73|
3c:b1:5b|44:32:2a|58:16:26|6c:fa:58|70:30:18|70:38:ee|70:52:c5|80:1d:aa|84:83:71|
90:fb:5b|a0:51:c6|b4:b0:17|c8:f4:06|cc:f9:54|d4:ea:0e|fc:83:99|fc:a8:41|b0:ad:aa|
10:cd:ae|50:cd:22|b4:47:5e|d4:78:56|c0:57:bc|38:bb:3c|e4:5d:52|a4:25:1b|6c:a8:49|
a0:12:90|f8:15:47|50:61:84|bc:ad:ab|b4:a9:5a|f8:73:a2|64:c3:54|64:a7:dd|64:6a:52|
00:09:6e|00:07:3b|00:0d:18|00:0d:28|00:04:2e).*
```

```
[set_voip:(avaya|Polycom)]
scope = IsPhone
role = default
```

EXAMPLE: Refuse user authentication without prior machine authentication

```
[EthernetEAP]
filter = connection_type
operator = is
value = Ethernet-EAP
```

```
[EAPTLS]
filter = radius_request
attribute = EAP-Type
operator = is
value = TLS
```

```
[machine]
filter = node_info
attribute = machine_account
operator = defined
value = robert
```

```
[REJECT_NO_MACHINE_AUTH_NO_EAPTLS:EthernetEAP&!machine&!EAPTLS]
scope = RegisteredRole
role = REJECT
```

```
[REJECT_NO_MACHINE_AUTH:EthernetEAP&!machine]
scope = RegistrationRole
role = REJECT
```

EXAMPLE: autoregister printers and scanners

```
[printers]
filter = node_info.device_class
operator = is
value = Printers/Scanners
```

```
[HP_Printer]
filter = node_info.mac
operator = regex
value = ^(00:1e:0b:0e:22:4c|00:23:7d:91:0e:86|00:1f:29:19:1d:4f|00:18:fe:a0:26:e2|
00:1e:0b:1b:d2:e7|2c:76:8a:3f:21:cd|00:14:38:d8:a1:14|00:14:38:de:b6:0a)$
```

```
[autoreg_printers:(printers)]
scope = AutoRegister
role = printers
```

```
[printers_role:(printers)]
scope = NodeInfoForAutoReg
role = printers
```

You can also add notes to the device using filters. Here you can see how to add a note to all HP Printers.

```
[Add_Note_HP_Printer:HP_Printer]
scope = RegisteredRole
role = printers
action = modify_node
action_param = mac = $mac, notes = HP PRINTER
```

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

## RADIUS Filter Definition

---

We added the ability to specify filters directly in the portion of code that return the radius answer or do a call to the API.

These rules are only available in one scope:

```
returnRadiusAccessAccept
```

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)
violation
user_role
vlan
```

For example, lets define a rule that return Access Accept when the connection is Ethernet-EAP and when there is no violation (merge\_return means that the original answer of PacketFence will be merge with the filter answer automatically):

```
[violation]
filter = violation
operator = defined
```

```
[ethernetetap]
filter = connection_type
operator = is
value = Ethernet-EAP
```

```
[1:ethernetetap&!violation]
merge_answer = no
scope = returnRadiusAccessAccept
```

In this other example we just add a new attribute to the original answer in the same conditions (here \$user\_role will be replaced by the real user role of the device and \${switch.\_portalURL} will be replaced by the value of \_portalURL defined in the switch config):

```
[1:ethernetep&!violation]
merge_answer = yes
scope = returnRadiusAccessAccept
answer1 = Cisco-AVPair => url-redirect-acl=$user_role;url-redirect=
${switch._portalURL}/cep$session_id
```

Here is an example on filtering by switch IP addresses and using MAC authentication, and then changing session and idle timeouts.

```
[switch_ip]
filter = switch._ip
operator = regex
value = ^(172\.24\.237\.112|172\.24\.237\.114|172\.24\.237\.128)$
```

```
[macauth]
filter = connection_type
operator = is
value = WIRED_MAC_AUTH
```

```
[1:switch_ip&macauth]
scope = returnRadiusAccessAccept
merge_answer = no
#Setting the timeout value as random between 10620 and 12600
answer1 = Session-Timeout => 10620..12600
#Terminate-Action to 0 means Terminate session
#Terminate-Action to 1 means Reauthenticate
answer2 = Termination-Action => 1
answer3 = Idle-Timeout => 240
```

Fingerbank can also be used to target specific devices as seen below. We can then apply Cisco ACLs to them.

```
[ipad_by_name]
filter = fingerbank_info.device_hierarchy_names
operator = includes
value = Apple iPad
```

```
[1:ipad_by_name]
scope = returnRadiusAccessAccept
merge_answer = yes
answer1 = Cisco-AVPair => ip:inacl#101=permit ip any any
```

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

# 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.

## 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"
```

## 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.

## 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.

```

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
}

```

## 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.

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. For SMS the phone number and the PIN code should be used.



## Note

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

In `/usr/local/pf/conf/radiusd/packetfence-tunnel` uncomment the line `# packetfence-local-auth` and restart radiusd.

This will activate the feature for any local account on the PacketFence server. You can restrict which accounts can be used by commenting the appropriate line in `/usr/local/pf/raddb/policy.d/packetfence`. For example, if you would want to deactivate this feature for accounts created via SMS, you would have the following :

```
packetfence-local-auth {
    # Disable ntlm_auth
    update control {
        &MS-CHAP-Use-NTLM-Auth := No
    }
    # Check password table for local user
    pflocal
    if (fail || notfound) {
        # Check password table with email and password for a sponsor registration
        pfguest
        if (fail || notfound) {
            # Check password table with email and password for a guest
            registration
            pfsponsor
            if (fail || notfound) {
                # *Don't* check activation table with phone number and PIN code
                # pfsms <--- This line was commented out
                if (fail || notfound) {
                    update control {
                        &MS-CHAP-Use-NTLM-Auth := Yes
                    }
                }
            }
        }
    }
}
```



## Note

For this feature to work, the users' passwords must be stored in clear text in the database. This is configurable via `advanced.hash_passwords`.



## EAP Local User Authentication

---

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

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

```
# Uncomment the following line to enable local PEAP authentication
packetfence-local-auth
```

Restart the radiusd service in order to apply the change.

`/usr/local/pf/bin/pfcmd service radiusd restart`



### Caution

You will need to disable password hashing in the database for local authentication to work. In the administration interface, go in *Configuration* → *System Configuration* → *Main Configuration* → *Advanced* and set *Database passwords hashing method* to **plaintext** or **ntlm**. Don't use **bcrypt**.

## 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`

## 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
```

## 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.

### Violations

Using PacketFence, it is possible to add violations 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 violation 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.

# 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 violations 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.

## 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.

## 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.

## 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 in the global database.

## 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 from 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.

## 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.

## 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 allows easier understanding.

# Intrusion Detection System Integration

---

## 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 seperated by a comma. The parameters that are to be replaced by a named capture.

Example Action

Regex -

```
mac\s*:\s*(?&lt;mac&gt;[a-zA-Z0-9]{2}(:[a-zA-Z0-9]{2}){5}), notes\s*:\s*(?&lt;notes&gt;.*)
```

Action -

```
modify_node: mac, $mac, notes, $notes
```

## Suricata IDS

---

PacketFence already contains a syslog parser for Suricata. This is an example to raise a violation from a syslog alert on the Suricata SID.

The first step is to create the syslog regex parser and then create the violation.

### 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 *: (?<date>\d{2}\/\d{2}\/\d{4}-\d{2}:\d{2}:\d{2}.*) \[\*\*\] \[\d+:(?<sid>
\d+):\d+\] (?<message>.*) \[\*\*\].* (?<srcip>\d{1,3}(\.\d{1,3}){3}):(?<srcport>
\d+) -> (?<ip>\d{1,3}(\.\d{1,3}){3}):(?<port>\d+)
Action: trigger_violation 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\_violation(mac, 00:11:22:33:44:55, tid, 2000340, type, detect)

We can see that PacketFence will execute the violation on the MAC address 00:11:22:33:44:55.

## Violation Creation

Now you will need to create the violation with the trigger id 2000340 in order to isolate the device. In order to do so, go to **Configuration → Compliance → Violation → ADD VIOLATION**

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.



# Security Onion

---

## 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>

## PacketFence Integration

Once Security Onion is installed and minimally configured, integration with PacketFence is required to be able to raise violations 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_sguil { file("/var/log/nsm/securityonion/sguild.log"
    program_override("securityonion_ids")); };
# This line filters on the string "Alert Received"
filter f_sguil { match("Alert Received"); };
# 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_sguil source, apply the f_sguil
# filter and send it to the d_packetfence_alerts destination
log { source(s_sguil); filter(f_sguil); destination(d_packetfence_alerts); };
```

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 violations 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 violation 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
```

## 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/PackageFence/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 sosetup
```

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

```

## StreamScan Comprimise Detection System (CDS)

---

This is an example to raise a violation from a syslog alert on a StreamScan alert ID.

The first step is to create the syslog regex parser and then create the violation.

## 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_violation 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
- violation : trigger_violation('mac', '00:11:22:33:44:55', 'tid', '2022985',
'type', 'detect')
```

We can see that PacketFence will execute the violation on the MAC address 00:11:22:33:44:55.

## Violation creation

Now you will need to create the violation with the trigger id 2000340 in order to isolate the device. In order to do so, go to **Configuration → Compliance → Violation → ADD VIOLATION**

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.

## 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
```

# 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.

## Barracuda

---

### 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



**Firewall SSO** [X]

Hostname or IP Address ⓘ

Username ⓘ

Password ⓘ

Port of the service  [↑] [↓]

If you use an alternative port, please specify

Roles  [X]

Nodes with the selected roles will be affected

[Close] [Save]

## 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
```

## Checkpoint

### Enabling Identity Awareness on the Security Gateway

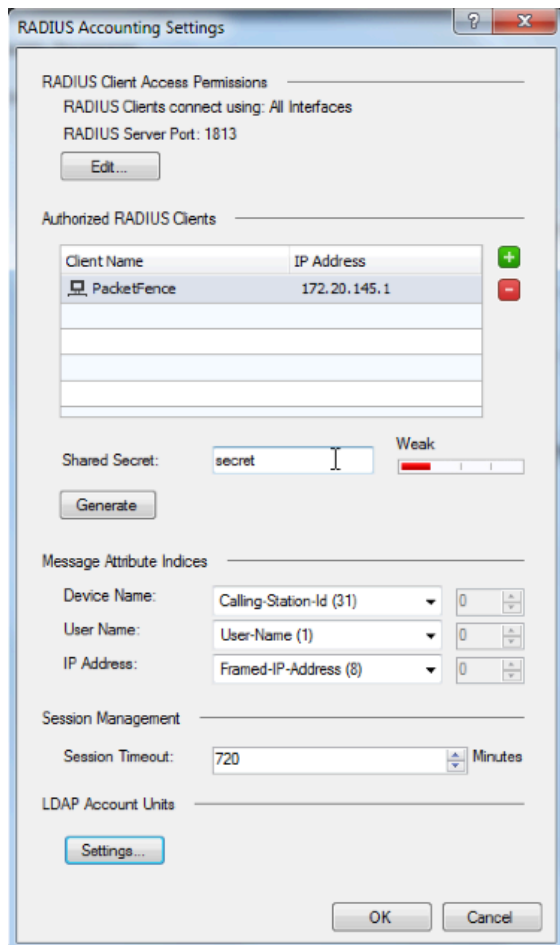
To enable Identity Awareness:

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*.

## 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*.

## 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)

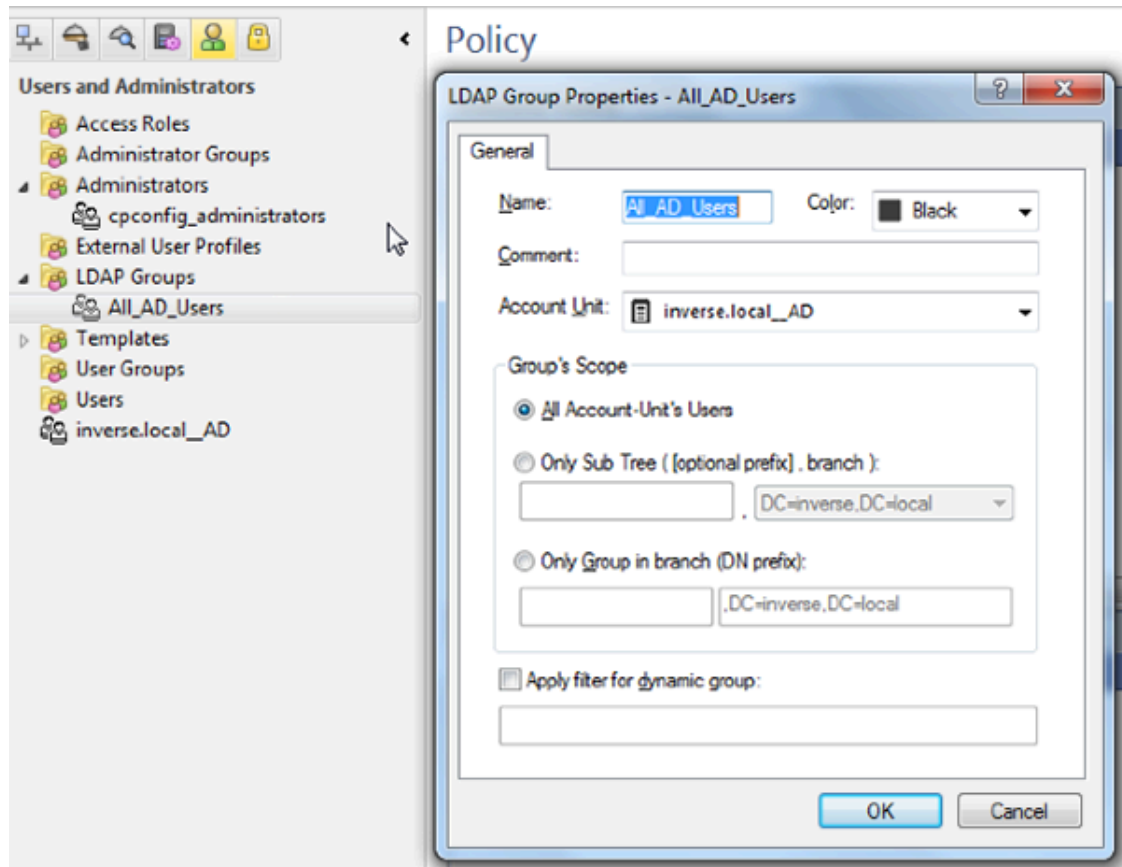
## RADIUS Client Access Permissions

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.

## LDAP Groups

Make sure that you have the correct LDAP Objects created on the Checkpoint.



## 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

**Firewall SSO** [X]

Hostname or IP Address ⓘ 192.168.100.2

Secret ⓘ .....

Port of the service 1813 ⓘ

If you use an alternative port, please specify

UID type PID

Roles staff [X]

Nodes with the selected roles will be affected

Close Save

## Verification

You can check the correct log in with the SmartView Tracker under **Network & Endpoint Queries** → **Predefined** → **Identity Awareness Blade** → **Login Activity**

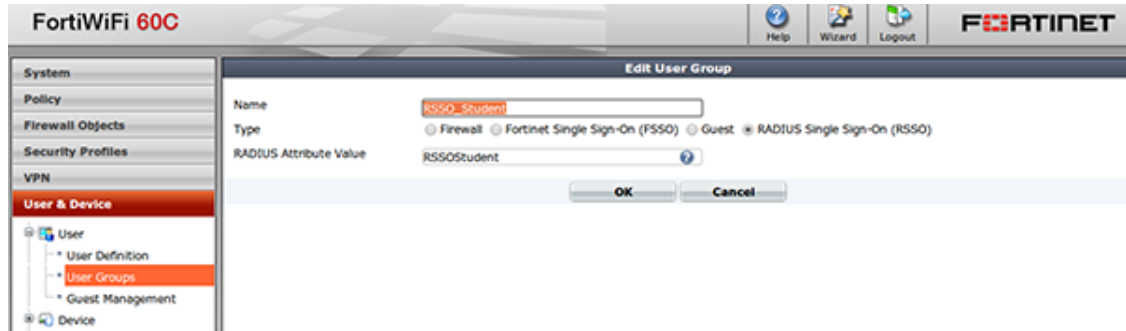
## FortiGate

---

### 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)



You can also see that in the webpage at **User & Device** → **Monitor** → **Firewall**

## Configure the endpoint attribute

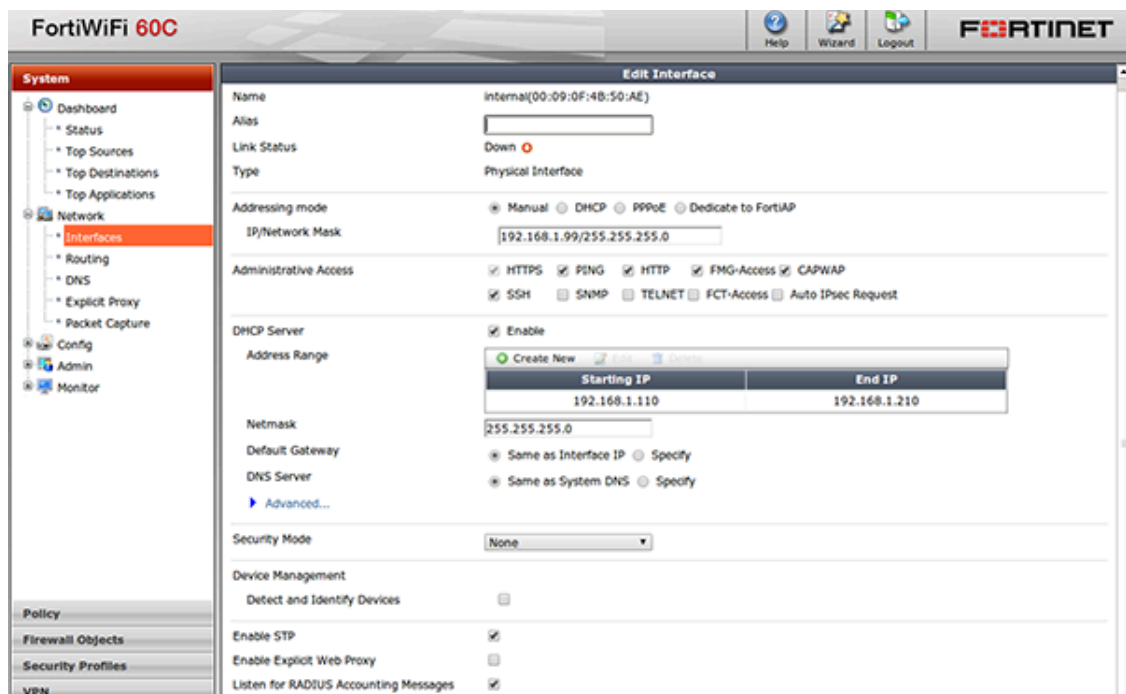
The default endpoint attribute is the Calling-Station-Id so the MAC address shows up under User Name, we can change that in CLI:

```
config user radius
edit RSSO_agent
set rso-endpoint-attribute User-Name
end
```

## Activate the Accounting Listening

Go to **System** → **Network** → **Interfaces**.

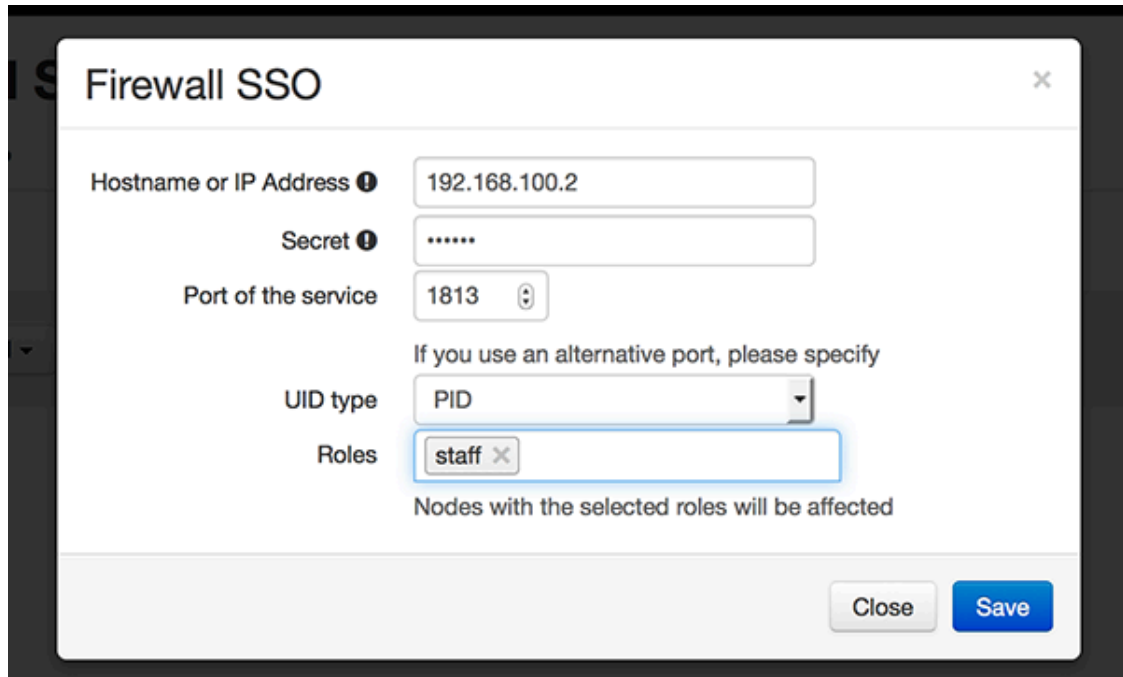
Select the interface that will communicate with PacketFence and check *Listen for RADIUS Accounting Messages* then confirm.



## 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



**Firewall SSO**

Hostname or IP Address ⓘ 192.168.100.2

Secret ⓘ .....

Port of the service 1813 ⓘ

If you use an alternative port, please specify

UID type PID

Roles staff x

Nodes with the selected roles will be affected

Close Save

## 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
```

# iBoss

---

## JSON-RPC

---

### 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>.

### 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



**New Firewall SSO JSONRPC**

Hostname or IP Address \*

Username \*

Password \*

Port of the service

If you use an alternative port, please specify

Roles

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

## Palo Alto

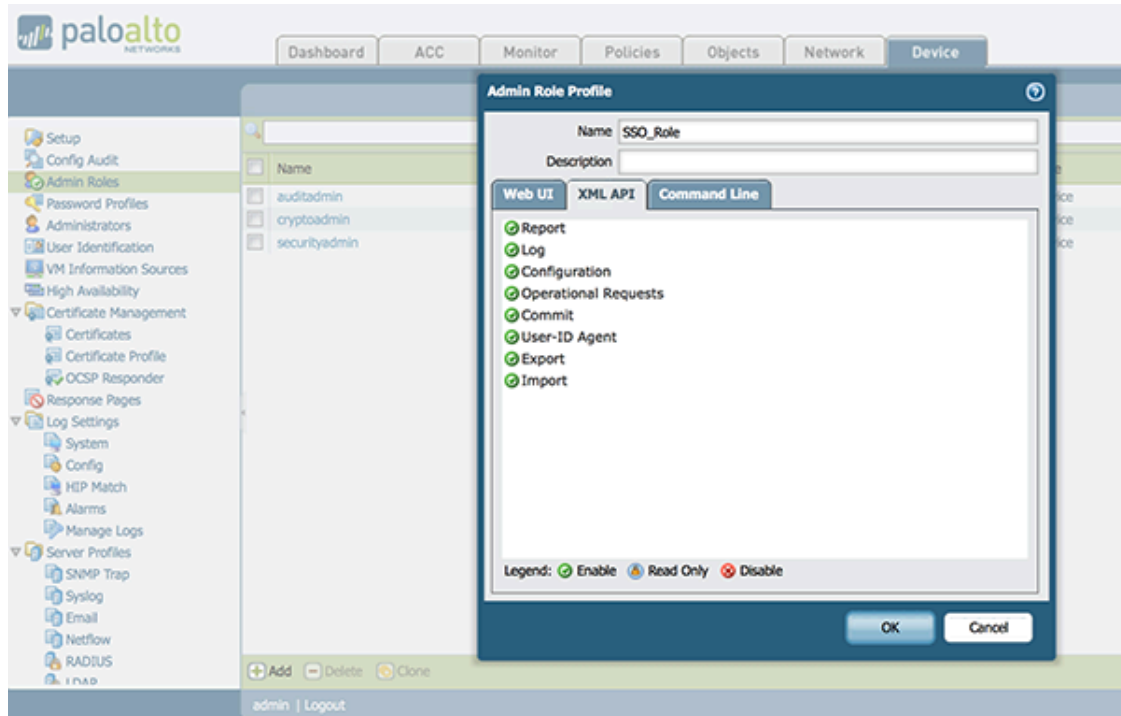
### Installation using XMLAPI

#### 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`.

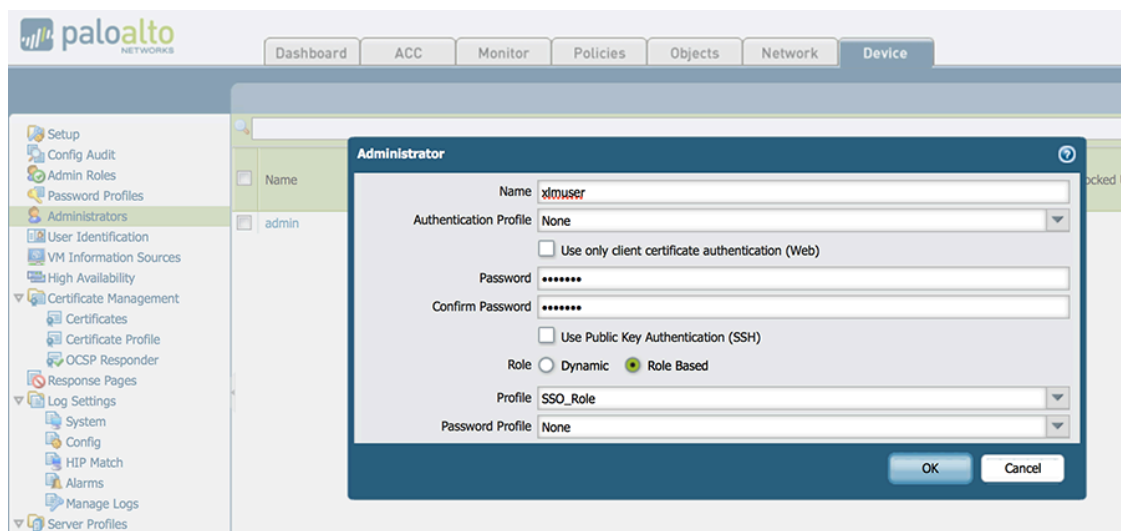


## Create the account in PAN-OS

Now you have created the role, you will associate an user with it.

Go to **Device** → **Administrator** → **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>
LUFRT1jeFV6SHd1QnJHaU55dnYvR1FNSkJNeTR6Uzg9TDgzNVlj0=
</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:** LUFRT1jeFV6SHd1QnJHaU55dnYvR1FNSkJNeTR6Uzg9TDgzNVlj0= (use the key previously generated)
- **Port of the service:** 443
- **Roles:** add the roles that you want to do SSO with

The screenshot shows the 'New Firewall SSO' configuration window. The fields are filled as follows:

- Hostname or IP Address:** 192.168.100.1
- Firewall Type:** PaloAlto Firewall
- Secret or Key:** [Masked]
- Port of the service:** 443
- Roles:** staff, gaming

Below the roles field, it says: "If you use an alternative port, please specify". At the bottom, there is a note: "Nodes with the selected roles will be affected". The 'Save' button is highlighted in blue.

## 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
Never				

## Installation using syslog



### Note

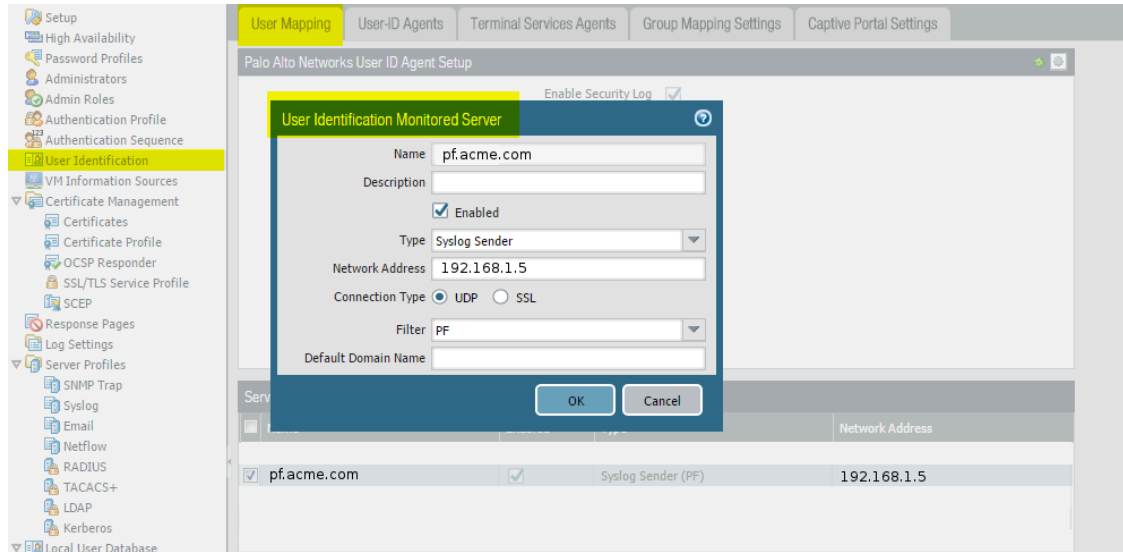
This installation mode is not suggested unless you use the SSO for informational purposes (no enforcement). PacketFence will use easily spoofable UDP packets to communicate with the Palo Alto firewall. If you require encryption and origin validation of the SSO messages, please use the XML API.

## 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.

# 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.

## Installation

---

### 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.

### OpenVAS

Please visit [http://www.openvas.org/install-packages.html#openvas4\\_centos\\_atomic](http://www.openvas.org/install-packages.html#openvas4_centos_atomic) to configure the correct repository to be able to install the latest OpenVAS scanning engine.

Once installed, please make sure to follow the instructions to correctly configure the scanning engine and create a scan configuration that will fit your needs. You'll also need to create a user for PacketFence to be able to communicate with the server.

It is important to get the correct scan config ID and NBE report format ID to populate the parameters in the PacketFence configuration file. The easiest way to get these IDs is by downloading both of the scan configuration and report format from the OpenVAS web gui and retrieve the IDs in the filenames.

For example `report-format-f5c2a364-47d2-4700-b21d-0a7693daddab.xml` gives report format ID `f5c2a364-47d2-4700-b21d-0a7693daddab`.

## WMI

You just have to enable WMI on each Microsoft Windows devices with a GPO from Active Directory.

## Configuration

---

In order for the compliance checks to correctly work with PacketFence (communication and generate violations inside PacketFence), you need to configure these sections:

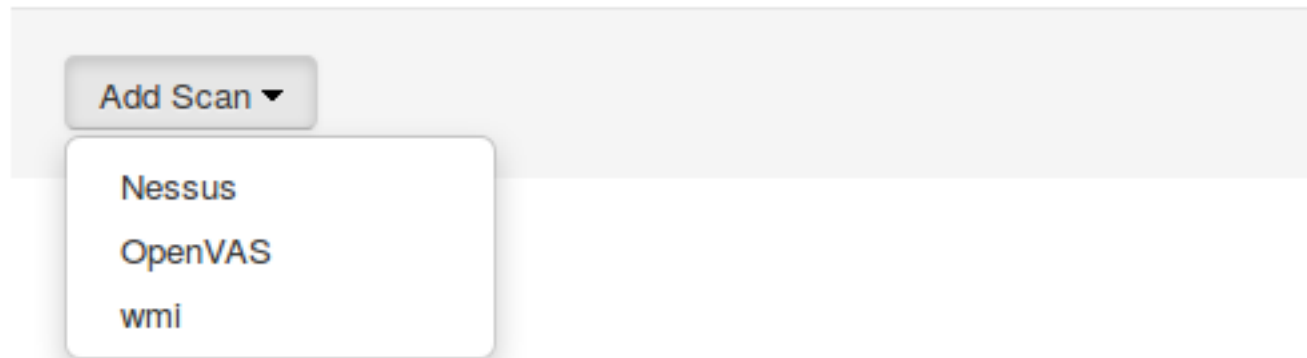
### Scanner Definition

First go in Configuration and Scanner Definition: *Configuration*→*Compliance*→*Scans Engines*

Then add a scan:

## Scan Engine

**Name**



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 after 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)  
802.1X: Even if the auto-registration has been enabled, the scan will be trigger on a EAP connection  
802.1X types: comma delimited EAP type that will trigger the scan if 802.1X above has been enabled

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)  
OpenVAS config ID: the ID of scanning configuration on the OpenVAS server

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

## 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.

Go in configuration → WMI Rules Definition:

There are already 3 rules defined:



```
Software_Installed
logged_user
Process_Running
```

Let's take the Software\_Installed rule:

```
request: select * from Win32_Product
```

```
Rules Actions:
```

```
[Google]
attribute = Caption
operator = match
value =Google
```

```
[1:Google]
action=trigger_violation
action_param = mac = $mac, tid = 888888, type = INTERNAL
```

This rule will do the following:

```
retrieve all the installed software on the device and test if the attribute
Caption contain Google.
if it matched then we will trigger a violation (with the trigger internal::888888)
for the mac address of the device.
```

The second one, logged\_user:

```
request: select UserName from Win32_ComputerSystem
```

```
Rules Actions:
```

```
[UserName]
attribute = UserName
operator = match
value = (.*)
```

```
[1:UserName]
action = dynamic_register_node
action_param = mac = $mac, username = $result->{'UserName'}
```

This rule will do the following:

```
retrieve the current logged user on the device and register the device based on
the user account.
```

The last one, Process\_Running:

```
request: select Name from Win32_Process
```

Rules Actions:

```
[explorer]
attribute = Name
operator = match
value = explorer.exe
```

```
[1:explorer]
action = allow
```

This rule will do the following:

```
retrieve all the running process on the device and if one match explorer.exe then
we bypass the scan.
```

- Rules syntax

the syntax of the rules are simple to understand:

the 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 we define 2 sorts of blocs:  
The test bloc (ie [explorer]) and the action bloc (ie [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
- value is the value you want to compare

Feel free to define multiples test blocs

The action bloc is where you will define your logic, per exemple let's take this one [1:google&explorer], this mean that if the google test is true and explorer is true then we execute the action.

The logic can be more complex and can be something like that [1:!google| (explorer&memory)] that mean if not google or (explorer and memory)

## 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.

## Violations Definition

You need to create a new violation section and have to specify:

Using Nessus:

```
trigger=Nessus::<violationId>
```

Using OpenVAS:

```
trigger=OpenVAS::<violationId>
```

Where **violationId** 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 violation related database contents using:

```
$ pfcmd reload violations
```



### Note

Violations will trigger if the plugin is higher than a low severity vulnerability.

## 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.

# Integrating Provisioning Agents

---

## 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.

### 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: 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-Enterprise to integrate with the PacketFence PKI.
- PKI Provider: This 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
```

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
```

## 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 configreload 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 **Internal->**, 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 **Policies and Access Control->Connection Profiles, Add 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

### New Provisioning Entry Windows

Provisioning ID **!**

Description

Roles

Nodes with the selected roles will be affected

SSID

Broadcast network

Uncheck this box if you are using a hidden SSID

EAP type

Select the EAP type of your SSID

Security type

Select the type of security applied for your SSID

PKI Provider

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".

### New Provisioning Entry Apple Devices

**Settings** **Signing**

Sign Profile

Check this box if you want the profiles signed

The certificate for signing profiles

The Certificate for signing in pem format

The private key for signing profiles

The Private Key for signing in pem format



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 included between the begin and end tag, not including the delimiters themselves.

Certificate file example:

```
----- BEGIN CERTIFICATE -----  
1234567890asdfghjkl  
zxcvbnmqwertyuiop78  
----- END CERTIFICATE -----
```

Copy everything between the BEGIN and END lines, but not the lines themselves. Repeat this operation for the certificate key and intermediate certificate.

```
----- BEGIN PRIVATE KEY -----  
1234567890asdfghjkl  
zxcvbnmqwertyuiop78  
----- END PRIVATE KEY -----
```

## 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.

# MobileIron

---

## 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.

The screenshot shows the MobileIron Admin Portal interface. The top navigation bar includes 'DASHBOARD', 'USERS & DEVICES', 'ADMIN', 'APPS', 'POLICIES & CONFIGS', 'SETTINGS' (highlighted in red), and 'LOGS & EVENTS'. Below this, a sub-navigation bar shows 'Preferences' (selected), 'Sentry', 'Connector', 'LDAP', 'Operators', 'Local CA', 'Service Diagnostics', and 'Templates'. The main content area is titled 'Preferences' and contains several sections:

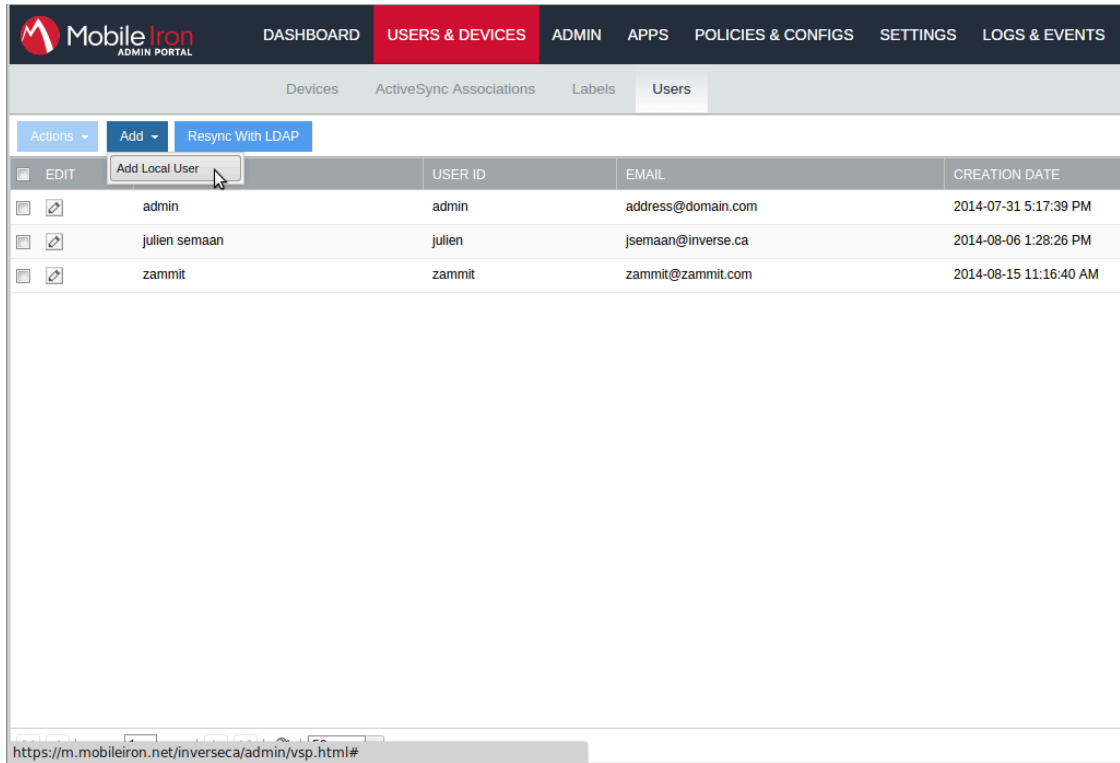
- Keep MDM Settings:**
  - Remove Wi-Fi settings for cellular-enabled devices
  - Remove Wi-Fi settings for cellular-enabled and Wi-Fi only devices
  - Clear passcode
- APNs Preferences:**
  - Use Enterprise Issued APNs certificate [i](#)
  - [View Certificate](#)
- MDM Preferences:**
  - Enable MDM Profile
  - View MDM Alerts
  - Enable Profile Encryption
  - Permit expired client certificates [MDM Certificate Report](#)
  - Enable MDM for iOS 4.1 and greater [i](#)
  - [Install MDM Certificate](#) [View Certificate](#)
- Save User Password Preferences:**
  - Save User Password:  Yes  No
- Registration PIN code Preferences:**
  - Passcode Length:  (6-12)

At the bottom of the page, there are 'Save' and 'Cancel' buttons.

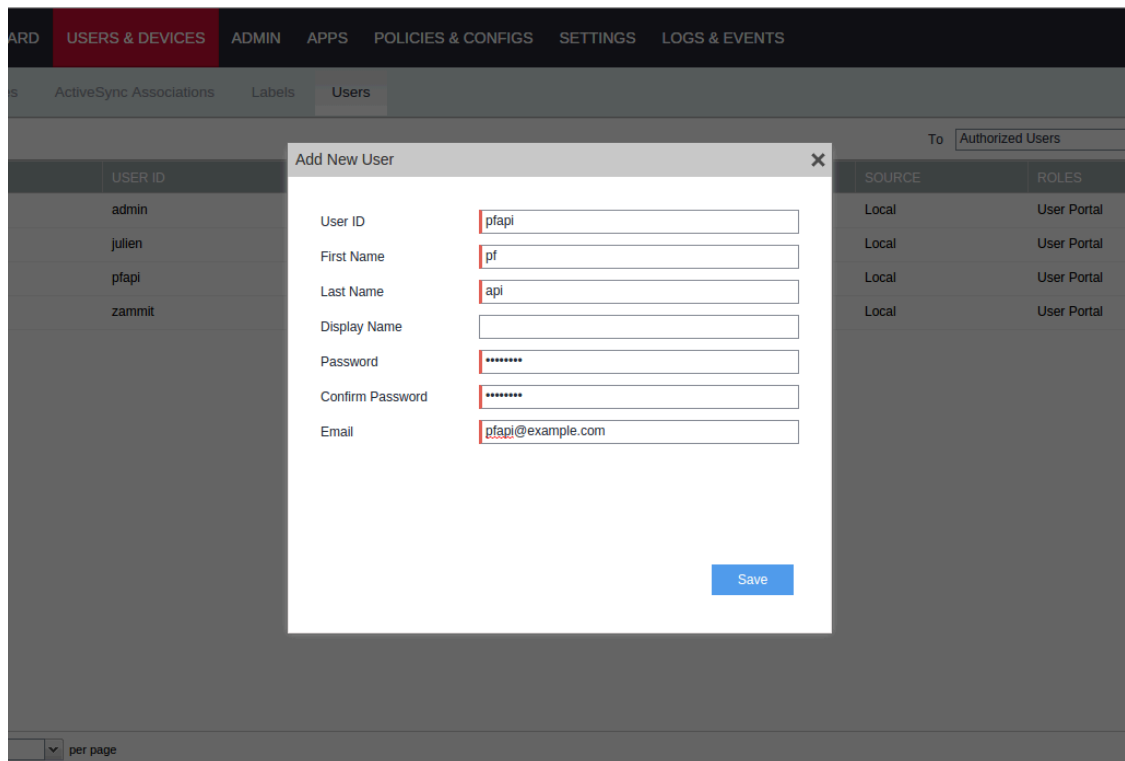
## Create an API user

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*.

The screenshot shows the MobileIron Admin Portal interface. The top navigation bar includes 'DASHBOARD', 'USERS & DEVICES', 'ADMIN' (highlighted), 'APPS', 'POLICIES & CONFIGS', 'SETTINGS', and 'LOGS & EVENTS'. Below this, there are tabs for 'ADMINS' and 'DEVICE SPACES'. A table lists users with columns for 'USER ID', 'EMAIL', 'SOURCE', and 'ROLES'. The 'pfapi' user is selected, and a dropdown menu is open over it, showing 'Assign to Space' with 'API' selected. The URL at the bottom is <https://m.mobileiron.net/inverseca/admin/vsp.html#>.

	USER ID	EMAIL	SOURCE	ROLES
<input type="checkbox"/>	admin	address@domain.com	Local	API, Add dev
<input type="checkbox"/>	julien semaan	jsemaan@inverse.ca	Local	
<input checked="" type="checkbox"/>	pfapi	pfapi@inverse.ca	Local	
<input type="checkbox"/>	zammit	zammit@zammit.com	Local	

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.

## 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**.

The screenshot shows the MobileIron Admin Portal interface with the 'USERS & DEVICES' tab selected. A dialog box titled 'Registration instructions for: zammit' is open. The dialog contains the following text:

An SMS containing a link to the registration page has been sent to  
Also, if you selected the 'Email User' option, then zammit@zammit.com will receive an email containing registration instructions and the following information:

- Server Address: **m.mobileiron.net:50291**
- User Name: zammit
- Password: your password

If the text message does not arrive or the recipient is not able to follow the link, then the recipient should go directly to the following site using the device's web browser:

<https://m.mobileiron.net/inverseca/cid/android.html>

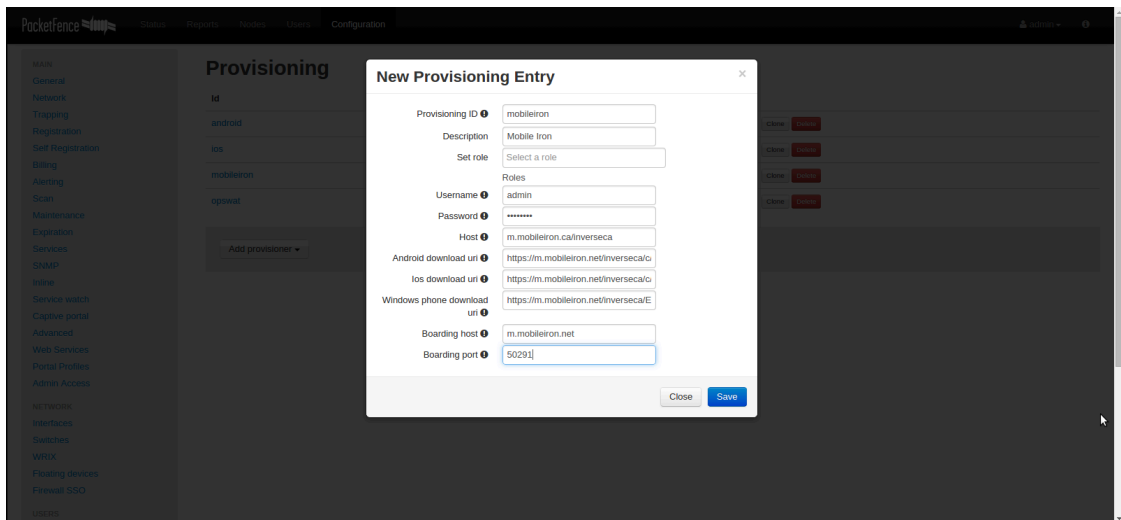
The dialog has an 'OK' button at the bottom right.

## 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*.



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.

The screenshot shows the PacketFence Configuration page for Portal Profiles. The left sidebar contains a navigation menu with options like Registration, Billing, and Portal Profiles. The main content area is titled 'Configuration' and includes the following fields:

- Profile Name: default
- Profile Description: Default Profile
- Logo: /common/packetfence-cp.png
- Redirection URL: http://www.packetfence.org/
- Force redirection URL:  (disabled)
- Reuse dot1x credentials:  (disabled)
- Enable Billing Engine:  (disabled)
- Number of Registration Pages: 0
- Languages: 1 en\_US
- Sources: 1 null
- Provisioners: 1 mobileiron

## 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

The screenshot shows the PacketFence Configuration page for Passthroughs. The left sidebar contains a navigation menu with options like Sources, Provisioners, and Passthroughs. The main content area is titled 'Configuration' and includes the following fields:

- Wireless IPS:  (disabled)
- Wireless IPS threshold: 90
- Passthrough:  (checked)
- Passthroughs: m.mobileiron.net, \*.itunes.apple.com, itunes.apple.com, play.google.com, \*.play.google.com
- Proxy Passthroughs: (empty field)
- Proxy Interception:  (disabled)
- Proxy Interception Port: 8080, 3128

## 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*.

## 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.

## OPSWAT

---

### 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.

### 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.

### 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



## 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=
```

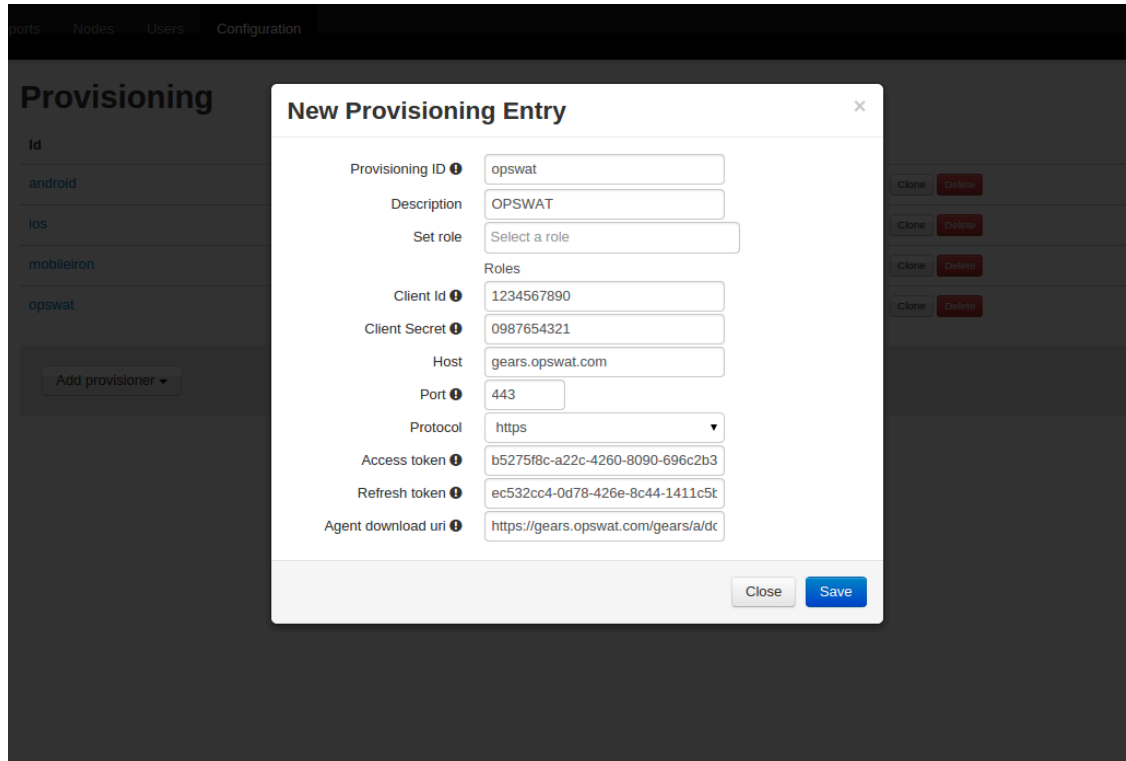
You should now be presented with a JSON response that contains the access and refresh token. Take note of both of these values for the PacketFence configuration. Example:

```
{"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"}
```

## 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*.

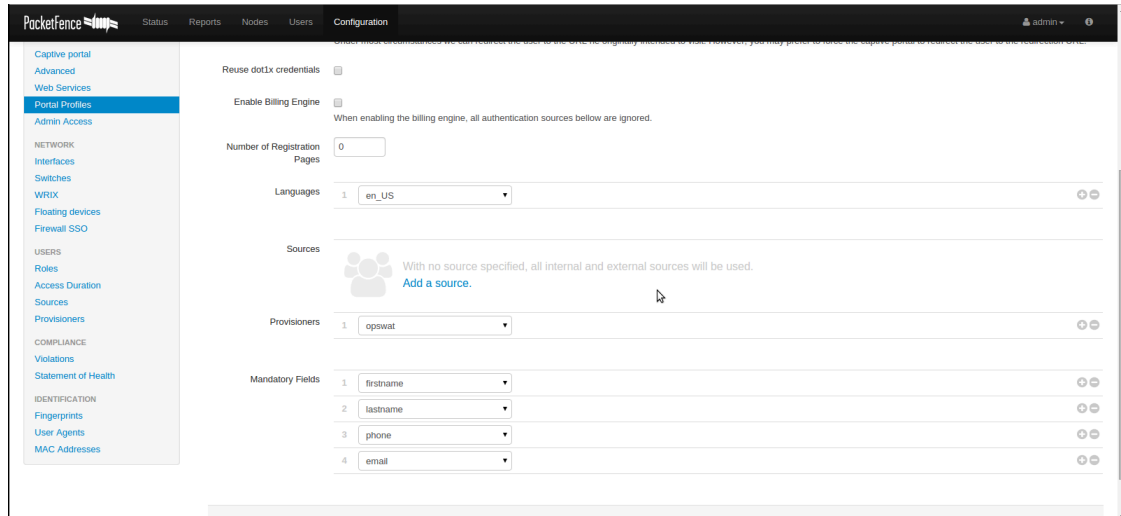


Now configure this new provisioner with the information you got above.

- 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

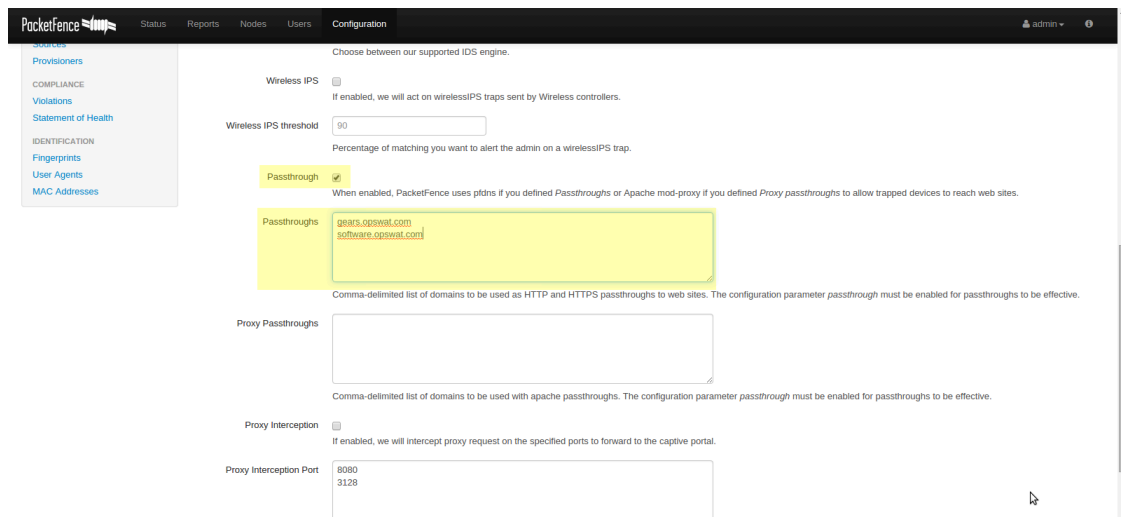
Now that you have created the provisioner, go in the *Connection Profiles* menu on the left and select the default portal. Click *Add Provisioner* and select the new OPSWAT Metadefender Endpoint provisioner that was created earlier.



## 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



## 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.

## Compliance enforcement

PacketFence polls the OPSWAT Metadefender Endpoint API in order to trigger violations 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	VM's	Servers
<b>Antiphishing</b>		<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"/>					
<b>Antivirus</b>		<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 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 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"/>					
<b>Backup</b>		<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"/>					
<b>Encryption</b>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

## Configure PacketFence

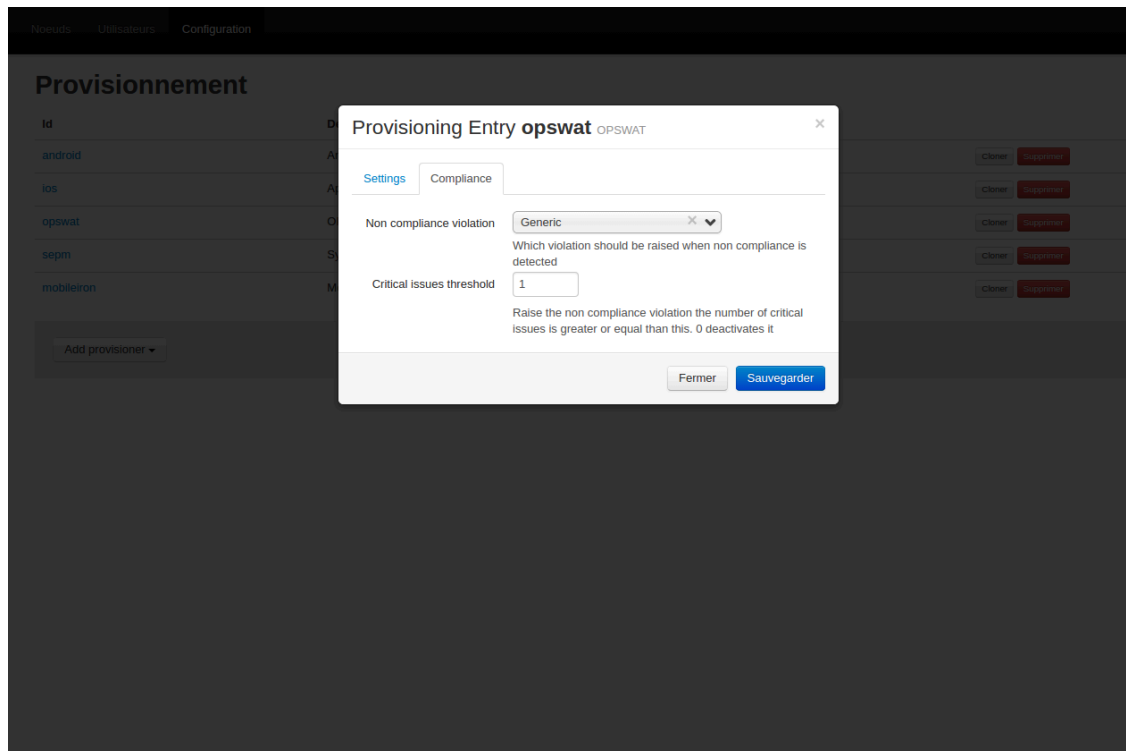
Now in order to enforce the compliance of the devices using the flagged critical issues above, you will need to configure the provisioner you created above to activate the enforcement.

Back in the provisioner configuration, go in the *Compliance* tab.

You now have to configure the violation that is raised when a device is noncompliant. Using the violation *Generic* 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 template the violation is using in the *Connection Profile* section. Simply select your connection profile and then go in the *Files* tab.

You can then modify the template violations/generic.html so it displays additional information.

You can also customize this violation in the *Violations* section of the administration interface. Refer to the PacketFence Administration Guide for additional information about this.

## SentinelOne

---

### 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.

**Choose update platform:**

**Latest available installer versions:**

PLATFORM	FILENAME	SIZE	SHA1
Windows	SentinelOne_windows_v1.8.4	62.79 MB	45279d1a2
Windows	SentinelOne_windows_v1.8.4	61.91 MB	23b6a72ba...
OS X	SentinelOne_osx_v1.8.4_20x	2.91 MB	2fc358c4b8...

[SAVE](#) [DISCARD CHANGES](#)



### 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.

## 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.

**Users** [+ USER](#) [DELETE](#)

ADMIN	NAME	ROLE
██████	Admin	Admin
██████	Admin	Admin
██████	Admin	Admin

**DETAILS**

Username:

Full name:

Email:

Role:

Password:

Confirm password:

[CANCEL](#) [CREATE](#)

**User Details**

Username - Unique identifier of the user in the system, must contain only alpha numeric characters.  
 Full name - Must contain only alpha numeric characters.  
 Role - Admin - An administrator with ability to perform any action. Viewer - Is not allowed to perform any actions / commands, but can still access every page. When trying to perform any action, the system will return an appropriate error message.

**Password**

Password must be at least 8 characters long.

**API Token**

API Token is used by REST API clients that require access to the system on behalf of a user. Use the API token by adding it to every request without the need for a login. Only a single token is supported for every user, which can always be revoked.

**Two Factor Authentication**

Two factor authentication, ZFA, is used to increase the security of the login process. Use [Google Authenticator](#), Duo or any other compliant application as the second method of authentication. The usage of ZFA can be enforced by the administrator for all users. In that case, they will be able to configure it on their next login. Alternatively, you can enable ZFA on a specific user even if the administrator hasn't forced the usage of ZFA for all users. To setup ZFA, first enable it and then use your phone QR code scanning within Google Authenticator or enter the code manually on your phone. This will establish trust with Google Authenticator which you can use later to login.

## 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 Provisioning Entry' form for SentinelOne. The form fields are as follows:

- Provisioning ID:** sentinelone
- Description:** SentinelOne
- Roles:** Click to add a role
- OS:** Add OS from Fingerbank (with a blue plus button) and Click to add an OS
- Host:** packetfence.sentinelone.net
- Port:** 443
- Protocol:** https
- API username:** packetfence
- API password:** [masked]
- Windows agent download URI:** /common/SentinelOne.exe
- Mac OSX agent download URI:** /common/SentinelOne.pkg

Buttons at the bottom right: Close, Save.

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 Mapf::errorc OSX 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 pfdns in order for the SentinelOne specific passthroughs to be taken into consideration.

```
# /usr/local/pf/bin/pfcmd service pfdns restart
```

## 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 violation 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.

## Symantec SEPM

---

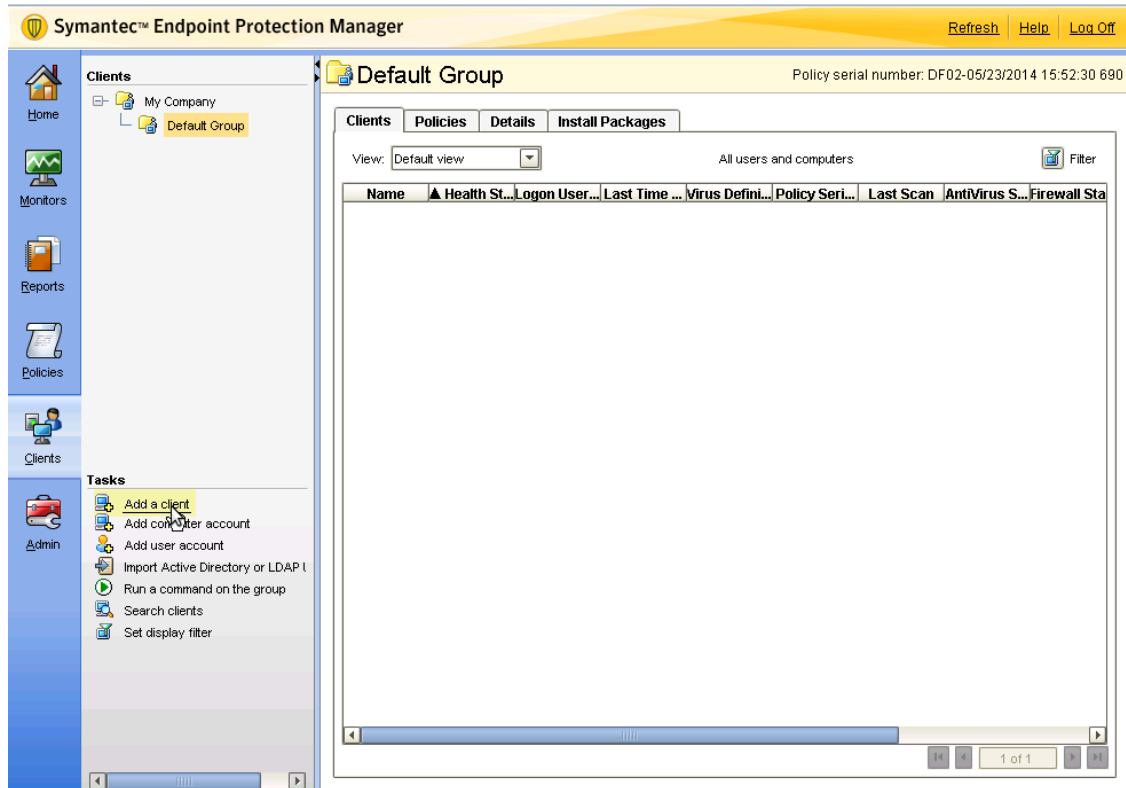
### Configure the SEPM

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.

### Create the install package

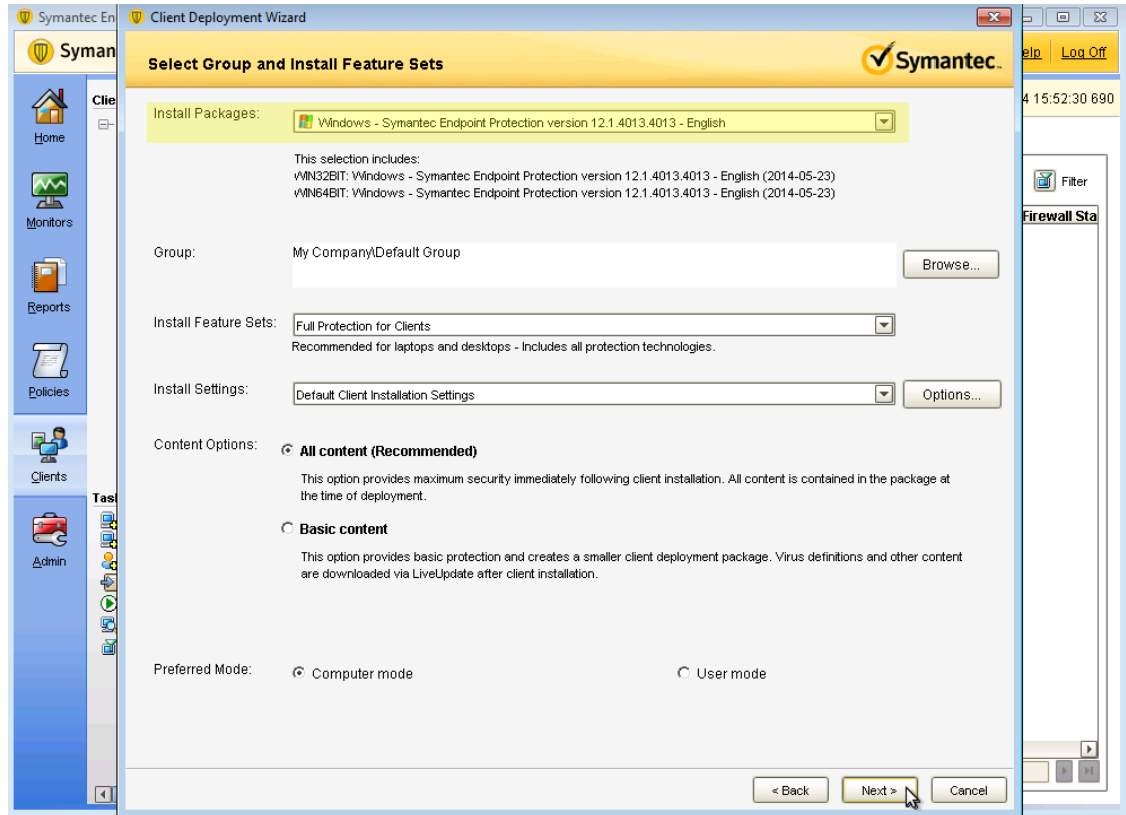
Login in your SEPM console and then go in the *Clients* tab on the left. Select the group your clients should belong and then click *Add a client*.





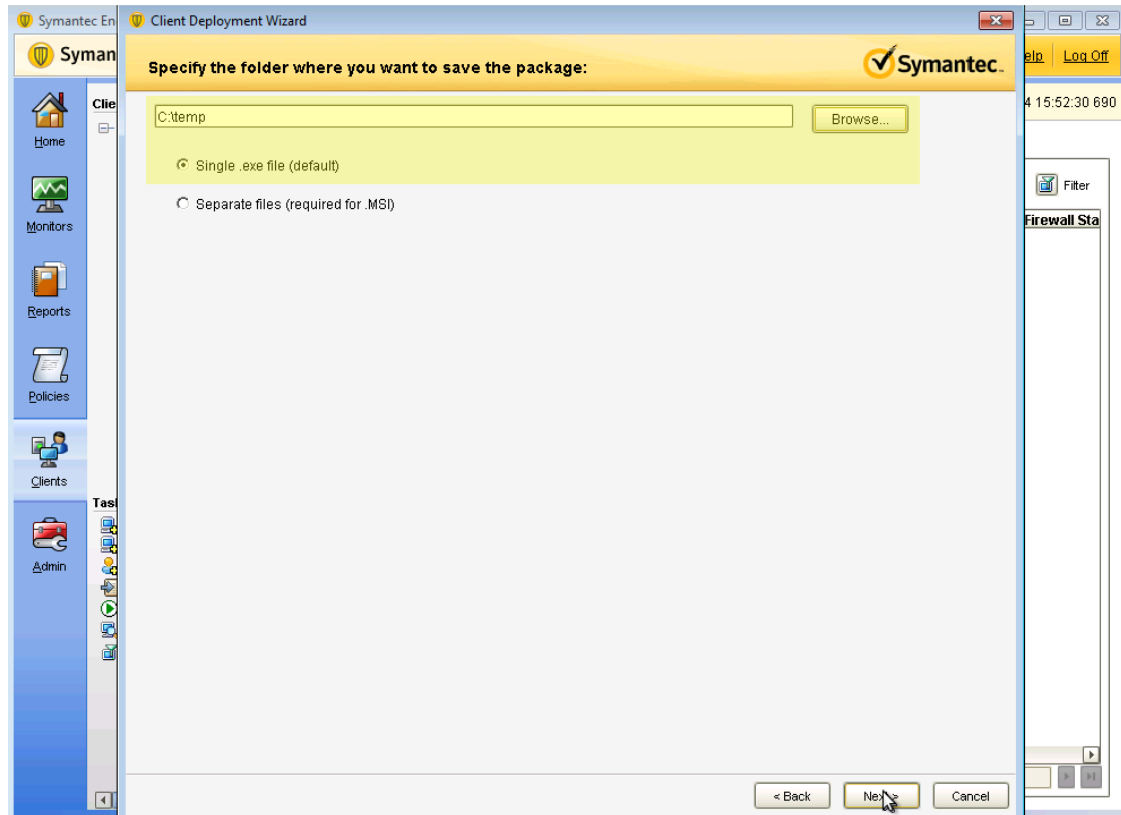
The wizard for the package creation will open. On the first page, make sure *New Package Deployment* is selected and click *Next*.

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`.

## 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

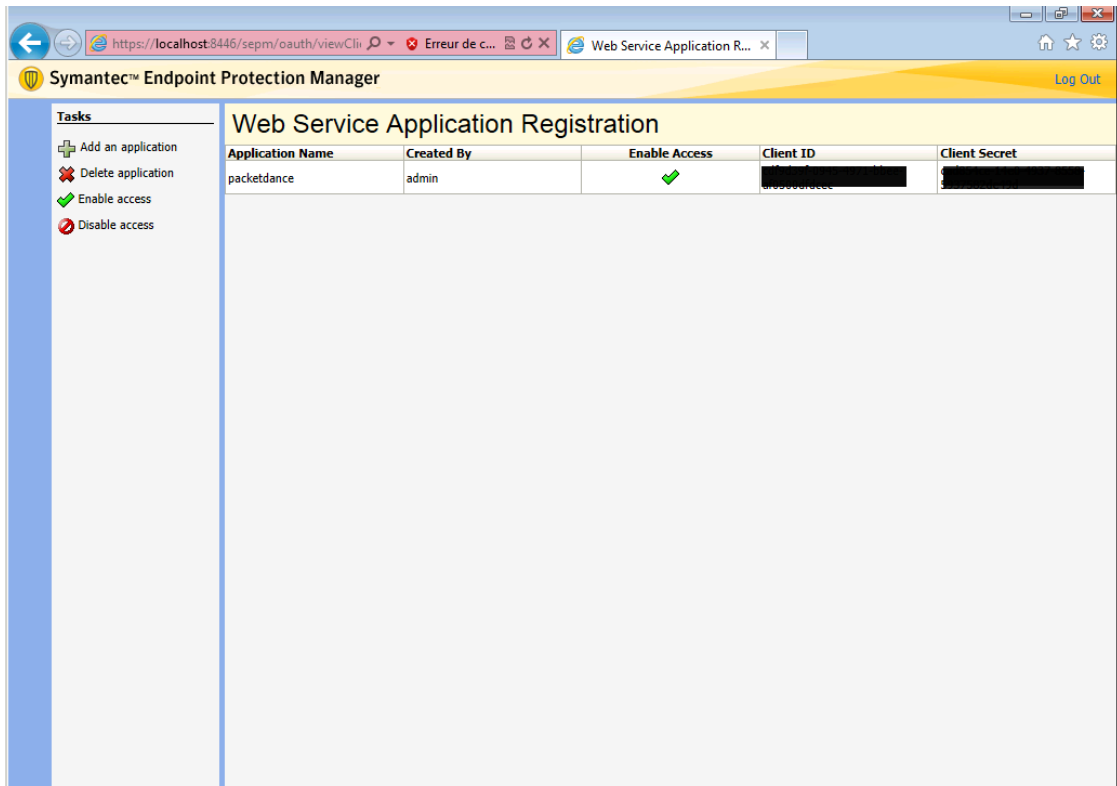
On your computer open a browser and access <https://192.168.1.100:8446/sep.m>.

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/sepm/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://172.21.2.186:8446/sepm/oauth/token?grant_type=authorization_code&client_id=-clientid-&client_secret=-clientsecret-&redirect_uri=http://localhost/&code=
```

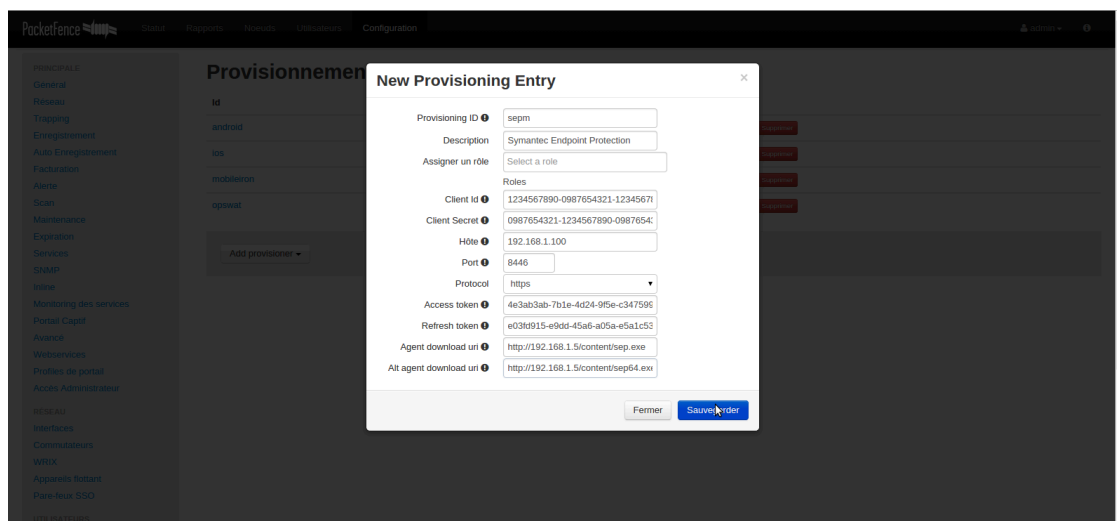
You should now be presented with a JSON response that contains the access and refresh token. Take note of both of these values for the PacketFence configuration. Example:

```
{"access_token": "4e3ab3ab-7b1e-4d24-9f5e-c347599a8a72", "token_type": "bearer", "refresh_token": "e03fd915-e9dd-45a6-a05a-e5a1c53c1ccd", "expires_in": 43199}
```

## 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*.



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.

PacketFence Configuration

Enable Billing Engine  When enabling the billing engine, all authentication sources below are ignored.

Number of Registration Pages:

Languages: 1 en\_US

Sources: With no source specified, all internal and external sources will be used. [Add a source.](#)

Provisioners: 1 sepm

Mandatory Fields: 1 firstname, 2 lastname, 3 phone, 4 email

[Save](#) [Reset](#)

## 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*.

## 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.

# PKI Integration

---

## 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.

### 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.

### 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



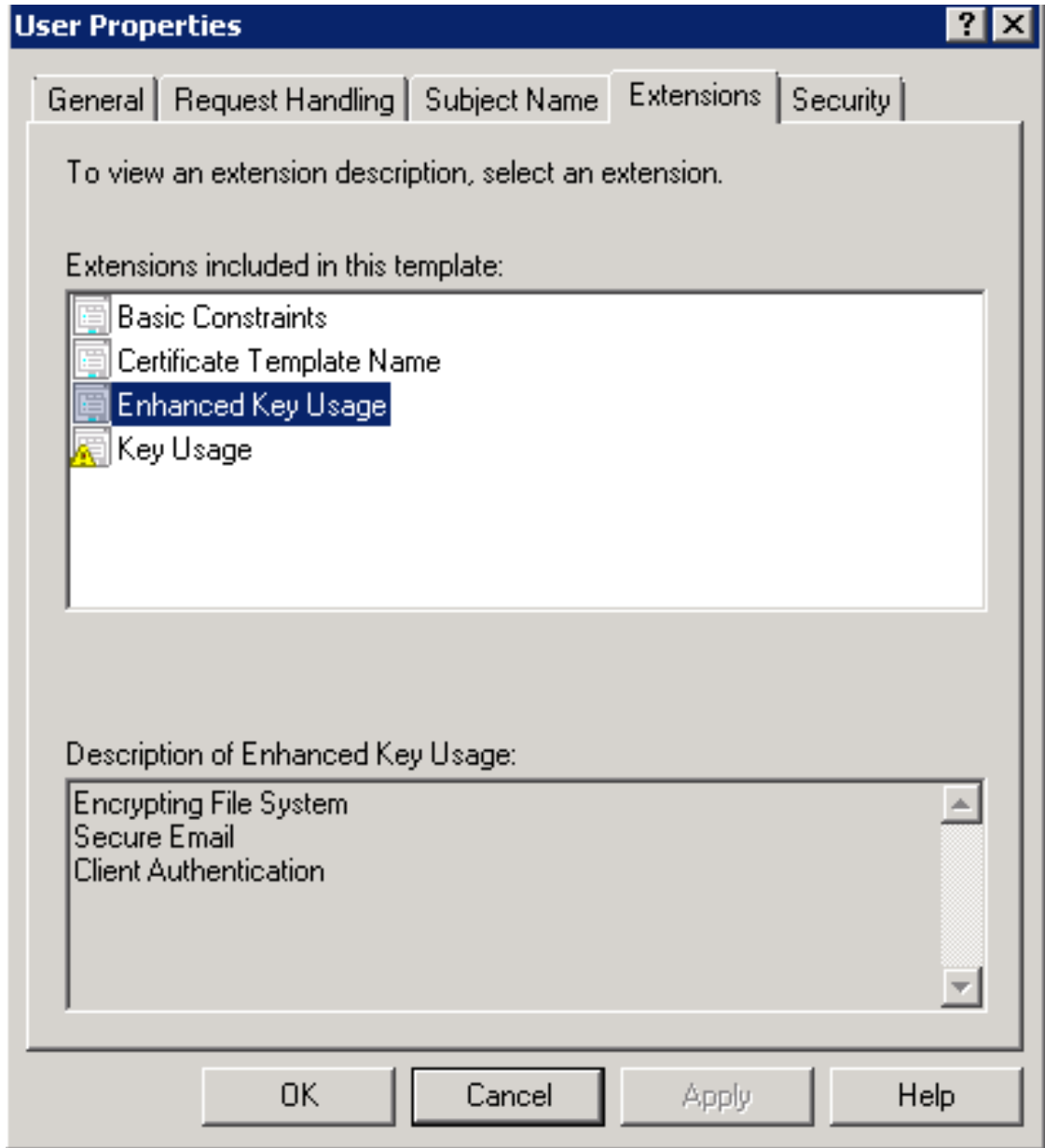
#### Caution

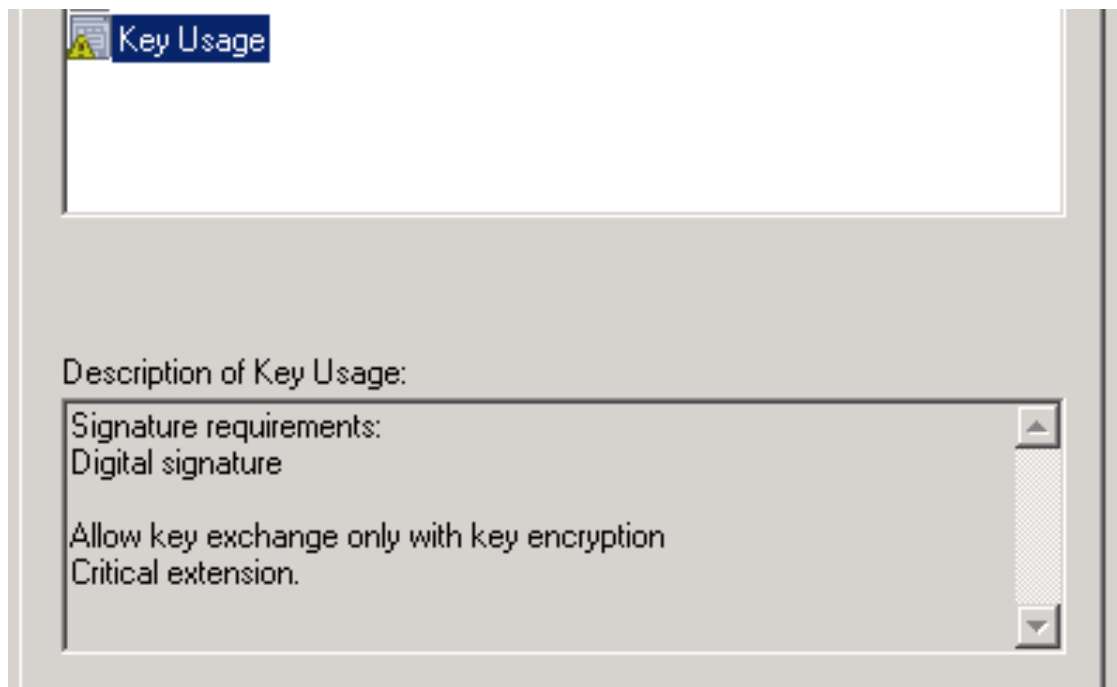
Remember that the validity of your CA can impact your whole certificate architecture.



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.

## 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->Users->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.

**PKI Provider**

PKI Provider Name   
The unique id of the PKI provider

Url   
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   
Country for the certificate

State   
State for the certificate

Locality   
Locality for the certificate

Organization

For the URL it will be `http://<ServerDNSName>/CertSrv/mscep/`.

You do not need any Username/Password combination for this configuration.

Organization   
Organization for the certificate

Organizational unit   
Organizational unit for the certificate

Common name Attribute   
Defines what attribute of the node to use as the common name during the certificate generation.

Ca cert path   
Path of the CA that will generate your certificates

Server cert path   
Path of the RADIUS server authentication certificate

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->Users->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

**New Provisioning Entry** Windows

Provisioning ID ⓘ

Description

Roles  ✕

Nodes with the selected roles will be affected

SSID

Broadcast network

Uncheck this box if you are using a hidden SSID

EAP type  ▼

Select the EAP type of your SSID

Security type  ▼

Select the type of security applied for your SSID

PKI Provider  ✕ ▼

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->Main->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 the following command in PacketFence CLI

```
# openssl req new -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 path/MyCA
```

Now you need to use the "CEP encryption" certificate and the "Enrollment agent". Both were obtained when doing the getca. You should have at least three certificates with the same name and a different number at the end. e.g. MyCA-0, MyCA-1, etc.

To display the content of a certificate use the following command:

```
# openssl x509 -in path/MyCA-0 -text
# openssl x509 -in path/MyCA-1 -text
```

In the output search for **X509v3 extensions:**. When using the scep enroll command you will need the "CEP Encryption" certificate as an argument for -e and the "Enrollment agent" certificate as an argument for -c. -d is use for the debug output. -l is the local file where your certificate will be save.

```
# sscep enroll -c path/MyCA-0 -e path/MyCA-1 -k path/local.key -r path/local.csr -
l path/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.

## PacketFence PKI

---

This section has been created to give a quick start to install 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.

### Assumptions

- You have at least one server with PacketFence 5.2 or later.
- The server already has a properly configured switch with 802.1X support.
- The PacketFence RADIUS server is working in your environment.
- A Mail Transport Agent is properly configured on the PacketFence server.
- The PacketFence management IP will be 192.168.1.5.
- The RADIUS shared secret is "useStrongerSecret".

## Installation

### Install the PKI



#### Note

The PacketFence PKI application is available on the PacketFence repository as the package "packetfence-pki". This is available for CentOS 6, CentOS 7 and Debian 7 and Debian 8.

PacketFence PKI can be installed on a standalone machine or on the PacketFence server itself. Communication between the PKI and PacketFence proper will take place over a REST API over port 9393.

## Preparing to install in standalone mode

- Open the ports 9393 and 9292 over TCP to be able to reach the PKI. This has to be done in the firewall (iptables).
- Disable SELinux:

```
vim /etc/selinux/config
```

- Change the line:

```
SELINUX=enforcing
```

to:

```
SELINUX=disabled
```

- Reboot for the change to take effect.

## Preparing to install with PacketFence

Allow TCP traffic over ports 9393 and 9292 through iptables:

```
vim /usr/local/pf/conf/iptables.conf
```

Uncomment the following lines.

```
# -A input-management-if --protocol tcp --match tcp --dport 9393 --jump ACCEPT
# -A input-management-if --protocol tcp --match tcp --dport 9292 --jump ACCEPT
```

Restart the iptables service:

```
bin/pfcmd service iptables restart
```

## Installation

### Debian

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 wheezy wheezy' > \
/etc/apt/sources.list.d/packetfence.list
```

Once the repository is defined, you can install `packetfence-pki` using:

```
sudo apt-key adv --keyserver keys.gnupg.net \
--recv-key 0xFE9E84327B18FF82B0378B6719CDA6A9810273C4
sudo apt-get update
sudo apt-get install packetfence-pki
```

## CentOS/RHEL

Installing on RHEL or CentOS requires the `packetfence` and `packetfence-extra` repositories. Install them first and then install the `packetfence-pki` package itself.

```
yum localinstall http://inverse.ca/downloads/PacketFence/CentOS6/x86_64/RPMS/
packetfence-release-1-2.centos6.noarch.rpm
yum install packetfence-pki --enablerepo=packetfence-extra, packetfence
```

## Configuring the PKI

### PacketFence-PKI Configuration



#### Note

The following configuration will allow you to generate certificates for an EAP-TLS configuration, if you want to use the PKI for another purpose you will have to adjust the Key Usage/Extended Key Usage fields.

1. Ensure that the service has started:

```
service packetfence-pki start
```

2. Log on to the PKI application:

- URL: <https://192.168.1.5:9393>
- Login: admin
- Default Password: [p@ck3tf3nc3](#)

After you login there are few steps to follow to have the PKI working for an EAP-TLS configuration with PacketFence. You will need to configure the following items:

- A Certification Authority (CA)
- Profiles for each purpose, e.g. server/client auth are 2 different profiles
- The REST API authentication and authorization
- A RADIUS server certificate
- A PKI Provider (optional)
- A Provisioner (optional)

Follow the four step wizard to create your certification authority, your client and server profile and your REST API configuration.

## Certification Authority

### • CA Initialize

# Welcome to PacketFence PKI

This wizard will help you to configure your PKI

- Below you have to fill the form to initialize and create the Certificate Authority
- This CA will sign your certificate

Step 1 of 4

#### Cn

Common Name

#### Mail

Email address of the contact for your organisation

#### Organisation

Organisation

#### Country

Country

#### State

State or Province

**Locality**

Locality

**Key type**

**Key size**

**Digest**

**Key usage**

Optional. One or many of: digitalSignature, nonRepudiation, keyEncipherment, dataEncipherment, keyAgreement, keyCertSign, cRLSign, encipherOnly, decipherOnly

**Extended key usage**

Optional. One or many of: serverAuth, clientAuth, codeSigning, emailProtection, timeStamping, msCodeInd, msCodeCom, msCTLSign, msSGC, msEFS, nsSGC

**Days**

Number of day the CA will be valid

Fill out the following fields that will be used to generate the certification authority.

- CN: Subject of the certification authority
- Mail: Email address of the admin or any other mail that suit your situation
- OU: Organization Unit
- Organization: Name of your company
- Country: Country (select in the list)
- State: state code (i.e. NY, CA, QC, etc...)
- Locality: City where is the organization
- Key type, size and digest: we recommend to use the following RSA, 2048, sha1
- Key Usage and Extended Key Usage are not necessary for the certification authority
- Days: Number of validity days, i.e. 10y = 3650



## Caution

Remember that after the expiration date of your certification authority, every certificate generated by it will be invalidated. We recommend at least 10 years for the CA.

## Server authentication Profile

This profile will be used to generate the RADIUS server certificate and key. The server certificate is used by the RADIUS server to authenticate its end of the connection to the client.

Fill out the following fields that will be used to generate the certificate profile to use for generating server certificates.

- Name: A name by which to identify this profile
- Validity: Number of validity days, i.e. 2y = 730
- Key type, size and digest: we recommend using the following: RSA, 2048, sha256
- Key Usage: Optional
- Extended Key Usage: "serverAuth"
- The P12 mail setup is mandatory for the server and client authentication profile. This is required to send the certificate and password by email using the **send certificate** button.
  - If your mail alerts are already working with PacketFence you should use the following:
    - P12 smtp server: 127.0.0.1
    - Tick P12 mail password

The following fields should be configured according to your preferences and will fill out the email sent when exporting the certificate:

- P12 mail subject: A descriptive line indicating that this is the certificate required to authenticate
- P12 mail from: The email address of the CA manager
- P12 mail header: Text that will appear in all emails sent with the certificate
- P12 mail footer: Optional, e.g. "This email has been generated automatically, please do not reply."

## • Server Profile Configuration

- Below you have to fill the form to initialize and create the Server Certificate Profile
- This Profile information will be use to create your server certificate and will be associated to the previous CA

Step 2 of 4

### Name

Profile Name

### Validity

Number of day the certificate will be valid

### Key type

### Key size

### Digest

### Key usage

Optional. One or many of: digitalSignature, nonRepudiation, keyEncipherment, dataEncipherment, keyAgreement, keyCertSign, cRLSign, encipherOnly, decipherOnly

### Extended key usage



### Caution

We recommend using a long validity for your RADIUS server certificate to avoid it expiring too frequently (i.e. two to five years).

## Client Authentication Profile

This profile will be used to generate the RADIUS client certificate and key. The client certificate is used by the 802.1X supplicant to authenticate its end of the connection to the server.



## • Client Profile Configuration

- Below you have to fill the form to initialize and create the Client Certificate Profile
- This Profile information will be use to create your client certificate and will be associated to the previous CA

Step 3 of 4

### Name

Profile Name

### Validity

Number of day the certificate will be valid

### Key type

### Key size

### Digest

### Key usage

Optional. One or many of: digitalSignature, nonRepudiation, keyEncipherment, dataEncipherment, keyAgreement, keyCertSign, cRLSign, encipherOnly, decipherOnly

### Extended key usage

## REST API Configuration

### • REST API Configuration

- Below you have to fill the form to initialize and create the REST API
- This REST API will be associated to the client profile and will be used to communicate with PacketFence

Step 4 of 4

#### Name

REST Profile name

#### Allowed users

User allowed to use this API from PacketFence Hold down "Control", or "Command" on a Mac, to select more than one.




The fields shown above are required to allow use of the REST API over which PacketFence and the PKI exchange authentication information. A username and password are mandatory.

## Common tasks in PacketFence-PKI interface

### Change PKI default password



#### Caution

Please be sure to change the default password to the PKI.

To change PKI default password, you will need to change `admin` user password. Password can be changed in *Configuration* → *Users* after editing a user.

### User Creation

Additional users for specific tasks may be created under *Configuration* → *Users*.

All fields are mandatory. Users can be associated with the REST API configuration.

### Creating RADIUS server certificate



#### Caution

After the creation of your certificate you need to sign it.

1. Click on *Add Certificate* button in *Certificate* → *Certificates* menu

2. Fill out form fields:
  - a. Mail: mail address that will receive certificate file and password
  - b. Profile: your server profile defined above
3. Click on *Submit* button
4. Click on *Sign* button to sign your server certificate
5. Click on :
  - a. *Send certificate* to send certificate by mail with password
  - b. *Download certificate* to download certificate and receive certificate's password by mail

## Manage certificates profiles

You can manage certificates profiles under *Certificate* → *Profiles*.

# 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 private key.

The CA certificate generated by the PacketFence PKI will be placed in `/usr/local/packetfence-pki/ca/`. Copy the CA certificate (and not its private key) to the directory created above and make sure it is readable by the "pf" user.

In the case where the PKI was installed on the same server as PacketFence, this will mean for example:

```
cp /usr/local/packetfence-pki/ca/YourCA.pem /usr/local/pf/conf/ssl/tls_certs/
chown pf:pf /usr/local/pf/conf/ssl/tls_certs/*
```

Since the server certificate is stored in the PKI database, you will have to sign and export it to the PacketFence server.

On the PKI web interface, under Certificates click on the "sign" icon for the certificate for your RADIUS server. This will automatically sign the certificate with your CA. Use the *Send certificate* or *Download certificate* to export it. The certificate will be exported in p12 format which combines both the certificate and its key. The password to decrypt the file will be send by email.

Copy the p12 formatted file to the `tls_certs` directory on the PacketFence server:

```
scp /path/to/your/downloads/YourCert.* root@192.168.1.5:/usr/local/pf/conf/ssl/
tls_certs/
```

Then, convert the p12 file to the pem format using the `openssl` tool:

```
cd /usr/local/pf/conf/ssl/tls_certs/
openssl pkcs12 -in YourCert.p12 -nocerts -out YourCert.key -nodes
openssl pkcs12 -in YourCert.p12 -out YourCert.pem -clcerts -nokeys
```

Ensure that the files are readable by `pf`:

```
chown pf:pf /usr/local/pf/conf/ssl/tls_certs/*
```

## RADIUS EAP-TLS and packetfence-pki

Using the PKI generated certificates requires editing the RADIUS EAP configuration file.

Edit the `/usr/local/pf/conf/radiusd/eap.conf` file and replace the following lines with references to your new certificates in the `tls-common` configuration block:

```
private_key_file = [% install_dir %]/raddb/certs/server.key
certificate_file = [% install_dir %]/raddb/certs/server.crt
ca_file = [% install_dir %]/raddb/certs/ca.pem
```

E.g.

```
private_key_file = [% install_dir %]/conf/ssl/tls_certs/YourCert.key
certificate_file = [% install_dir %]/conf/ssl/tls_certs/YourCert.pem
ca_file = [% install_dir %]/conf/ssl/tls_certs/YourCA.pem
```

Certificate revocation checks also have to be configured using OCSP in the same block.

For example:

```
ocsp {
    enable = yes
    override_cert_url = yes
    url = "http://192.168.1.5:9292/pki/ocsp/"
}
```

Regenerate the new configuration files and restart `radiusd` to enable EAP-TLS using your CA signed certificates:

```
/usr/local/pf/bin/pfcmd service radiusd generateconfig
/usr/local/pf/bin/pfcmd service radiusd restart
```

## PacketFence provider configuration

Using the PKI requires configuring the PKI providers section in the PacketFence's web admin interface under *Configuration* → *Advanced Access Configuration* → *Users*. The provider configuration defines how PacketFence connects to the PKI REST API and which profile will be used.

Add a new PKI provider and select PacketFence PKI.

Fill out the form for a PKI provider according to the PKI configuration profile you created earlier. Pay attention to the username and password which have to match an authorized user in the PKI configuration.

### PKI Provider ×

PKI Provider Name \*

The unique ID of the PKI provider

Protocol

Protocol to use to contact the PacketFence PKI API

Host

Host which hosts the PacketFence PKI

Port

Port on which to contact the PacketFence PKI API

Username

Username to connect to the PKI

Password

Password for the username filled in above

Profile

Profile used for the generation of certificates

### PKI Provider ×

Organization

State for the certificate

Common Name Attribute

Organization for the certificate

Defines what attribute of the node to use as the common name during the certificate generation

Common Name Format

Defines how the common name will be formatted. %s will expand to the defined Common Name Attribute value

Revoke on unregistration

Check this box to have the certificate revoke when the node using it is unregistered. Do not use if multiple devices share the same certificate

CA cert path \*

Path of the CA certificate that will generate your certificates

Server cert path \*

The "CA cert path" and "Server cert path" fields 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 for the device itself using the MAC address as the identifier. If you select *Username*, a certificate will be generated for the user using his login name on the authentication backend (e.g. Active Directory).



### Important

This means that revoking the certificate for a username based certificate will block all the devices that this user registered. If you generate the certificates using the MAC address, revoking a certificate will block only that device.

## 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's web admin interface under *Configuration* → *Users* → *Provisioners*, see [Integrating Provisioning Agents](#) section.

## Revocation process

Certificates can be checked for revocation at authentication time using either OCSP to interrogate the PKI for every RADIUS authentication or using the certificate revocation lists (CRL).

OCSP is scalable, its main downside would be that one request per certificate authentication is sent to the PKI to verify if the certificate is still valid and that adds additional latency to authentication. Additionally, RADIUS authentication then becomes dependent on an external service which could be unreachable although that can be mitigated in the FreeRADIUS configuration.

Using a CRL implies that each time the CRL is updated, every services that uses this CRL has to download it again. For security reason we recommend a short delay on CRL expiration (to avoid using revoked certificate on the network).

By default a CRL list will be created when you revoke a certificate. The file will be under `/usr/local/packetfence-pki/ca/YourProfileName.crl`. Note that one CRL by profile will be created.

# Best Practices

## RHEL7 systemd early swapoff bug mitigation

A [known bug](#) is still present in systemd-219-30.el7\_3.7.x86\_64 shipped with CentOS. (Debian fixed it in 228-3).

The bug arises because not all swap aliases are registered, which results in an incorrect dependence tree which results in swapoff being called way too early at shutdown.

### Workaround

- Obtain the list of swap items that should be considered by systemd for it to enforce a correct ordering:

```
#grep swap /var/log/dmesg |grep "dead -> active"
```

In our example, that gave the following output:

```
[ 1.995413] systemd[1]: dev-dm\x2d1.swap changed dead -> active
[ 1.995495] systemd[1]: dev-c1-swap.swap changed dead -> active
[ 1.995550] systemd[1]: dev-disk-by\x2did-dm\x2dname\x2dcl\x2dswap.swap changed
dead -> active
[ 1.995616] systemd[1]: dev-disk-by\x2did-dm\x2duuid\x2dLVM
\x2dXOAK7DHxMdmQCrNdWWE3Pt836Q9pHYSGyr09ycCGeIYavzbamVWNKMaVUMLf1NWZ.swap changed
dead -> active
[ 1.995678] systemd[1]: dev-disk-by\x2duuid-6509e6e1\x2daf2d\x2d4d23\x2d9ebd
\x2da9aa8801e658.swap changed dead -> active
```

- Create `/etc/systemd/system/swap.target` and fill it with all swap aliases obtained from the previous command:

```
[Unit]
Description=Swap
Documentation=man:systemd.special(7)
After=dev-disk-by\x2duuid-6509e6e1\x2daf2d\x2d4d23\x2d9ebd
\x2da9aa8801e658.swap dev-dm1.swap dev-disk-by\x2did-dm\x2duuid\x2dLVM
\x2dXOAK7DHxMdmQCrNdWWE3Pt836Q9pHYSGyr09ycCGeIYavzbamVWNKMaVUMLf1NWZ.swap dev-
disk-by\x2did-dm\x2dname\x2dcl\x2dswap.swap dev-c1-swap.swap dev-dm\x2d1.swap
```

## IPTables

---

IPTables is now entirely managed by PacketFence. However, if you need to perform some custom rules, you can modify `conf/iptables.conf` to your own needs. However, the default template should work for most users.

## 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.



# Performance Optimizations

---

## 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→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.

## Additional Packages

In order to use this cache, you will need to install additional packages available from the `packetfence-extra` repository. These packages will allow PacketFence to query your Active Directory for the NT hash of the users.

```
CentOS: `` yum install python2-impacket --enablerepo=packetfence-extra ``
```

```
Debian: `` apt-get install python-impacket ``
```

## PacketFence Configuration

First of all, you will need to enable the NTLM caching globally by enabling *NTLM Redis cache* in *Configuration→System Configuration→Main Configuration→Advanced*. You then need to restart *radiusd*.

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 *LDAP filter* if necessary. Note that this is only used for the batch job.
- Adjust the *Expiration*
- Enable *NTLM cache background job* and/or *NTLM cache on connection*. In the case of this example, both will be enabled.

## Domain **inverseinc** ✕

**Settings** | NTLM cache

**NTLM cache**    
Should the NTLM cache be enabled for this domain?

**Source**    
The source to use to connect to your Active Directory server for NTLM caching.

**LDAP filter**    
An LDAP query to filter out the users that should be cached.

**Expiration**    
The amount of seconds an entry should be cached. This should be adjusted to twice the value of `maintenance.populate_ntlm_redis_cache_interval` if using

## Domain **inverseinc** ✕

LDAP filter	(&(samAccountName=*)(!(((lockoutTime=>0)(userAccountControl:1.2.840.113556.1.4.803:=2))))))
An LDAP query to filter out the users that should be cached.	
Expiration	604800 <input type="text" value="↑"/> <input type="text" value="↓"/>
The amount of seconds an entry should be cached. This should be adjusted to twice the value of <code>maintenance.populate_ntlm_redis_cache_interval</code> if using the batch mode.	
NTLM cache background job	<input checked="" type="checkbox"/>
When this is enabled, all users matching the LDAP filter will be inserted in the cache via a background job ( <code>maintenance.populate_ntlm_redis_cache_interval</code> controls the interval).	
NTLM cache on connection	<input checked="" type="checkbox"/>
When this is enabled, an async job will cache the NTLM credentials of the user every time he connects.	

Once done, click *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*→*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
```

If you chose to enable *NTLM cache background job* in one of your domains, you will need to enable the `pfmon` job that will periodically cache the credentials. This can be configured in *Configuration*→*System Configuration*→*Main Configuration*→*Maintenance*→`populate_ntlm_redis_cache`. It is advised to set the interval of this task to half the expiration of the credentials you have set in the domain configuration. This will ensure you have an optimal cache hit. Once done, restart the `pfmon` service.

## 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: <https://support.microsoft.com/en-us/kb/303972>

## 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.

## MySQL optimizations

---

### Tuning MySQL

If you're PacketFence system is acting very slow, this could be due to your MySQL 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 is 1.25 and IOWait is peaking at 20% - this is not good. If your IO wait is low but your MySQL is taking +50% CPU this is also not good. Check your MySQL install for the following variables:

```
mysql> 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
```

## MySQL Optimization Tool

We recommend that you run the [MySQL Tuner](#) on your database setup after a couple of weeks to help you identify MySQL configuration improvement. The tool is bundled with PacketFence and can be run from the command-line:

```
# /usr/local/bin/pftest mysql
```

## Keeping Tables Small

Over time, some of the tables will grow large and this will drag down performance (this is especially true on a wireless setup).

One such table is the `locationlog` table. We recommend that closed entries in this table be moved to the archive table `locationlog_archive` after some time. A closed record is one where the `end_time` field is set to a date (strictly speaking it is when `end_time` is not null and not equals to 0).

We provide a script called `database-backup-and-maintenance.sh` located in `addons/` that performs this cleanup in addition to optimize tables on Sunday and daily backups.

## 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 MySQL value tend 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.

## 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, MySQL 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.

## Using Percona XtraBackup

When dealing with a large database, the database backup and maintenance script (`/usr/local/pf/addons/database-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 Percona XtraBackup which can complete a full database backup without locking your tables.

The installation instructions below are made for CentOS 7 but adjusting them to Debian should only be a matter of installing the proper packages for your MySQL/MariaDB version.

First install the Percona repository:

```
# yum install http://www.percona.com/downloads/percona-release/redhat/0.1-3/
percona-release-0.1-3.noarch.rpm
```

Make sure to disable the newly installed repository not to interfere with future updates:

```
# sed -i -e 's/^enabled\ \=.*/enabled = 0/g' /etc/yum.repos.d/percona-release.repo
```

Next, install Percona XtraBackup by manually specifying the Percona repository:

```
# yum install percona-xtrabackup --enablerepo=percona-release-`uname -m`
```

Once this is done, grant the proper rights to the `pf` user (or the one you configured in `pf.conf`):

```
# mysql -u root -p
mysql> GRANT RELOAD, LOCK TABLES, REPLICATION CLIENT ON *.* TO 'pf'@'localhost';
mysql> FLUSH PRIVILEGES;
```

Next, run the maintenance script `/usr/local/pf/addons/database-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 Percona XtraBackup documentation for troubleshooting.





## Note

In the event that you want to stop using Percona XtraBackups for your MySQL backups, simply uninstall it and the database script will fallback to mysqldump.

# Captive Portal Optimizations

---

## Avoid captive portal overload due to non-browser HTTP requests

By default we allow every query to be redirected and reach PacketFence for the captive portal operation. In a lot of cases, this means that a lot of non-user initiated queries reach PacketFence and waste its resources for nothing since they are not from browsers. (iTunes, Windows update, MSN Messenger, Google Desktop, ...).

Since version 4.3 of PacketFence, you can define HTTP filters for Apache from the configuration of PacketFence.

Some rules have been enabled by default, like one to reject requests with no defined user agent. All rules, including some examples, are defined in the configuration file `apache_filters.conf`.

Filters are defined with at least two blocks. First are the tests. For example:

```
[get_ua_is_dalvik]
filter = user_agent
method = GET
operator = match
value = Dalvik
```

```
[get_uri_not_generate204]
filter = uri
method = GET
operator = match_not
value = /generate_204
```

The last block defines the relationship between the tests and the desired action. For example:

```
[block_dalvik:get_ua_is_dalvik&get_uri_not_generate204]
action = 501
redirect_url =
```

This filter will return an error code (501) if the user agent is Dalvik and the URI doesn't contain `/generate_204`.

## Dashboard Optimizations (statistics collection)

---

The collection and aggregation of statistics in the whisper database can be I/O intensive per moment. This means that it can be beneficial to separate them on another disk even if it is a virtual disk that will share the same underlying physical disk.

First, add a disk in your virtual machine or bare metal server and reboot (this example will use /dev/sdb as the new device).

Make sure packetfence is stopped:

```
# service packetfence stop
```

Create an ext4 partition:

```
# mkfs.ext4 /dev/sdb
```

Then move the old databases to a backup point:

```
# mv /usr/local/pf/var/graphite /usr/local/pf/var/graphite.bak
```

Mount your new disk and check that it is mounted:

```
# echo "/dev/sdb /usr/local/pf/var/graphite          ext4    defaults    1" >> /etc/fstab
# mkdir /usr/local/pf/var/graphite
# mount -a
# dh -h
```

Apply the proper user rights and restore your database from your backup

```
# chown pf.pf /usr/local/pf/var/graphite
# cp -frp /usr/local/pf/var/graphite.bak/* /usr/local/pf/var/graphite/
```

Start packetfence and make sure your stats are still there and being collected properly. Then remove the backup you made `rm -fr /usr/local/pf/var/graphite.bak/`.

## Troubleshooting

---

This section will address specific problems and known solutions.

## "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.
```

# Advanced Network Topics

---

## 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.

## 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 Vlans
- 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

## 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 that have to be tagged on the port.

## 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.

## 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.

## 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.

## 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
```

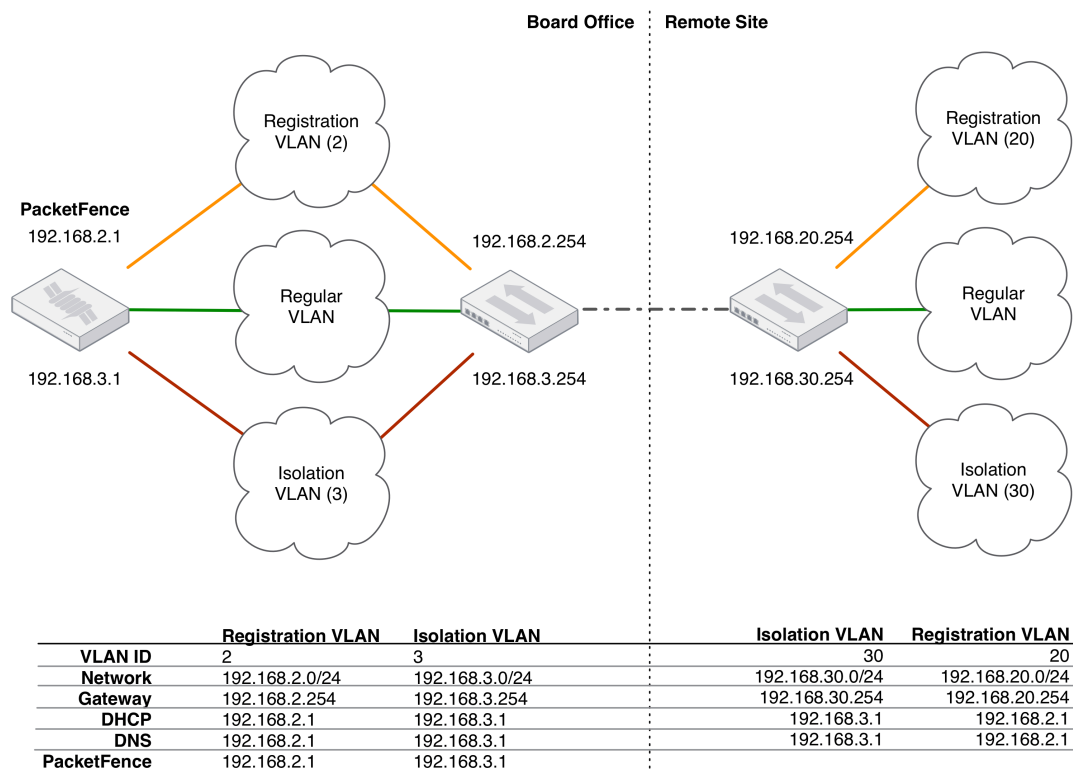
Repeat the above for all your production VLANs then restart PacketFence.

## 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 `pfdhcplistener` runs on the same interface where production DHCP runs. However, please note that this is **NOT** recommended. See [this ticket](#) to see why.

## 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
```

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.

## 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`.

## 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 exist it will be updated. In order to only define the switch group through the import, leave the description field empty.

## 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-register 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.

## 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.

## 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
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
```

## 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**.

## 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**.

## 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!

## 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.

### 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.

### 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.

## 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).

## 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 pfdhcpllistener and pfmon services. pfmon will query via SNMP all the switches to create a map (MAC ↔ switch) pfdhcpllistener 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.

# Additional Integration

---

## 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

## 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.

The project page can be found [here](#).

## 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.

# Active Directory Integration

---

## 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 → Taskschd.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*)

## 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 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 → 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*)

## 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 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 → 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`)

## 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.



### 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.

## Cisco Mobility Services Engine (MSE)

---

This section has been created to give a quick start to configure the Cisco MSE with PacketFence. In the first part we will explain how to use the Cisco MSE in order to present a captive portal based on the localization. Then in the second part we will integrate the device localization in the node view.

### Assumptions

- You have at least one server with PacketFence 6.3 or later.
- The PacketFence management IP will be 192.168.1.5.

- An account has been created on the MSE in order to use RESTful API of the MSE. (Read only for localization , write to create notifications)
- The Cisco MSE IP will be 192.168.1.6.

## Portal Based on Localization

### Enable httpd.collector service on PacketFence

This service is mandatory and will receive the mse notifications in json format. To enable this service go in **Configuration** -> **services** and tick **services.httpd\_collector** and leave the default order.

services.httpd\_collector  Should httpd.collector be started ? Keep enabled unless you know what you're doing.

services.httpd\_collector\_order  Order in which the service should be started.Lower order services start first and stop last.

### Create a Notification

This part can be done directly on the MSE web admin GUI but we include a Perl script in addons/mse-subscribe.pl that will help you to create one using the RESTful API of the MSE.

So let say that you have a username with write permissions on the MSE (mswrite/password) now go in /usr/local/pf/addons then run:

```
./mse-subscribe.pl --username=mswrite --password=password --
url=http://192.168.1.6:8083 --target-ip=192.168.1.5 --target-port=9292 --url-
path=/mse/ --zone=Campus>Building>Level>Zone --notification-name=Zone1
```

This will create the notification and print out the notifications configured on the MSE. So now each time a device will enter or leave the specific zone PacketFence will be notified.

### Configure a Portal Filter

In **Configuration Policies and Access Control** → **Connection Profiles** → **Add profile** → **Advanced filter** specify an advanced filter like that:

```
extended.mse_inout.locationMapHierarchy == "Campus>Building>Level>Zone"
```

PacketFence Status Reports Auditing Nodes Users **Configuration**

Advanced OMAPI Web Services Monitoring Clustering Metadefender Node MSE Tab Portal Modules **Portal Profiles** Admin Access Filter engines

NETWORK Interfaces and Networks Switches Floating Device Firewall SSO WRIX

USERS Roles Access Duration Sources Billing Tiers Provisioners PKI Providers

RADIUS

registering. Note that activating this disables the o

Automatically register devices  This activates automatic registration of devices for option only makes sense in the context of an 802.

Reuse dot1x credentials  This option emulates SSO when someone needs to and get the appropriate actions. As a security prec users use 802.1x credentials with a domain part (u authentication source needs to be configured for th configured for it.

Dot1x recompute role from portal  When enabled, PacketFence will not use the role in

Filters If  of the following conditions are met

With no filter specified, an

Advanced filter `extended.mse_inout.locationMapHierarchy == "Campus>Building>Level>Zone"`

And fill the other configuration needed to configure a connection profile then save it.

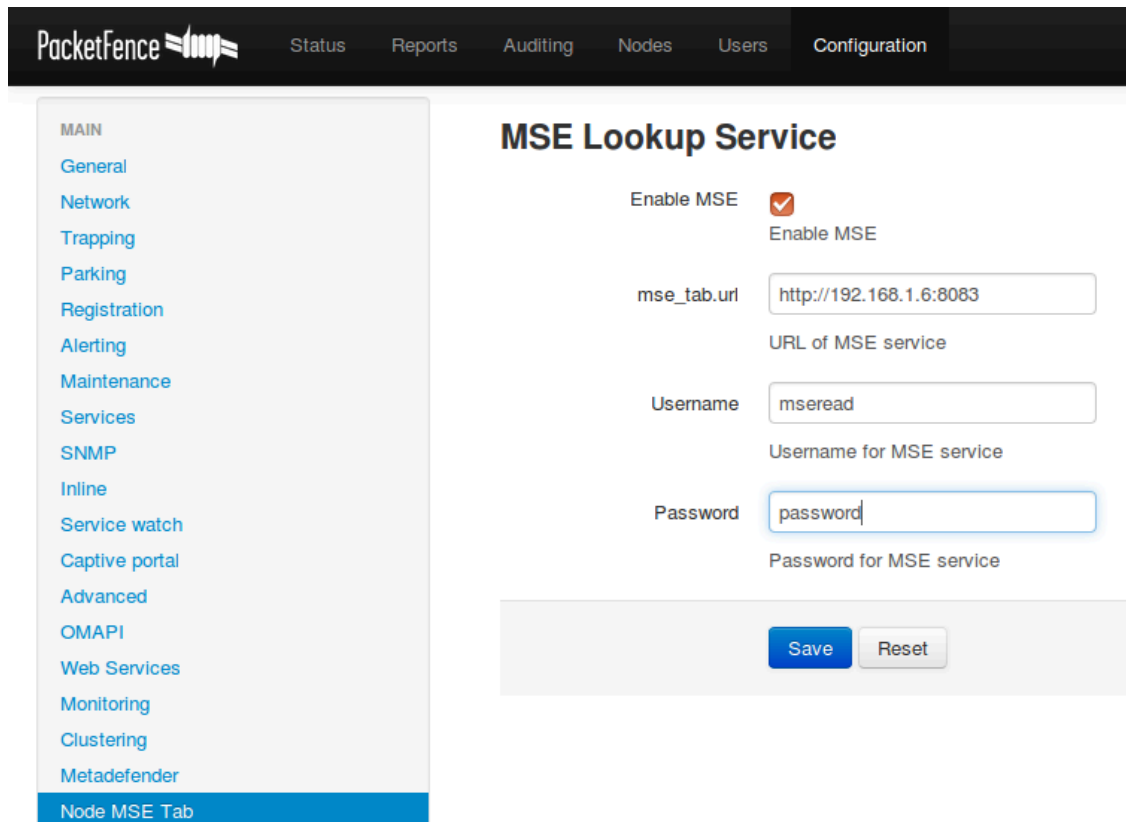
That's all, now when a device will hit the captive portal and will be in the specific zone then it will hit this connection profile.

## MSE Tab

This configuration is really simple, you just have to enable and fill the URL, the username and password in **Configuration → Integration → Cisco Mobility Service Engine**. So in our example:

```
URL: http://192.168.1.6:8083
Username: mseread
Password: password
```





The screenshot shows the PacketFence web interface. The top navigation bar includes the PacketFence logo and menu items: Status, Reports, Auditing, Nodes, Users, and Configuration. A left sidebar menu lists various configuration categories, with 'Node MSE Tab' highlighted at the bottom. The main content area is titled 'MSE Lookup Service' and contains the following configuration options:

- Enable MSE**: A checked checkbox with the label 'Enable MSE' below it.
- mse\_tab.url**: A text input field containing 'http://192.168.1.6:8083' with the label 'URL of MSE service' below it.
- Username**: A text input field containing 'mseread' with the label 'Username for MSE service' below it.
- Password**: A text input field containing 'password' with the label 'Password for MSE service' below it.

At the bottom of the configuration area, there are two buttons: 'Save' (in blue) and 'Reset' (in grey).

Now go in **Nodes** and click on a MAC address and you will see that a new tab appears, then if you click on it you will be able to retrieve the map and the localization of the device.

### MAC 64:89:9a:8d:4d:7a

Info Fingerbank IP Address Location Violations MSE WMI Rules

Get Current Location Get History Location

MAC Address	64:89:9a:8d:4d:7a
AP MAC Address	dc:a5:f4:8d:4a:b0
Band	UNKNOWN
Currently Tracked	1
Dot11 Status	ASSOCIATED
IP Address	10.36.34.22
Is Guest User	0
SSID	[REDACTED]
First Located Time	2016-10-27T14:13:58.746-0400
Last Located Time	2016-10-27T14:20:20.627-0400
Map Hierarchy	[REDACTED]

Reevaluate access Close Save

# 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 `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.

## Custom 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* tab of the administration interface.

In order to configure a report, you need to edit `/usr/local/pf/conf/report.conf` and add a section that will define your report.

The following attributes are available to define your report (the ones that have an asterisk are mandatory):

- `* description` : The user friendly description that will display for this report
- `* base_table` : The base SQL table that will be used to create the view
- `* columns` : The columns to select from the table(s) (ex: `node.mac`).
- `* date_field` : The field to use for date filtering. Will also be used as default sorting field unless `order_fields` is set in the report.
- `joins` : The tables to join to the base table and how to join them. See example below and [the following documentation](#).
- `group_field` : The field to group the entries by. No grouping is done if this field is omitted or empty.

- **order\_fields** : Comma delimited fields for the ordering of the report. 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`).
- **base\_conditions** : Comma delimited conditions that should be applied to the report. This can be used to filter the report without using the search in the administration interface to provide the proper unsearched view. Conditions should match the following format : **field:operator:value** (ex: `auth_log.source:=:sms,auth_log.status!=:completed`).
- **base\_conditions\_operator** : Whether the base conditions should be matched using an all or any logic. Accepted values are **all** and **any**.
- **person\_fields** : The fields in your report that represent a user in the PacketFence database. Field values in this field will be clickable and will allow to view/modify the values of the user in question. The fields must be listed with the name they have in the report header without any quotes and are comma delimited.
- **node\_fields** : The fields in your report that represent a node in the PacketFence database. Field values in this field will be clickable and will allow to view/modify the values of the node in question. The fields must be listed with the name they have in the report header without any quotes and are comma delimited.
- **searches** : Comma delimited searches that should be available on the report. Should match the following format **type:Display Name:field** (ex: `string:Username:auth_log.pid`).
  - **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



## Note

You should always prefix the fields with the table name and a dot (ex: `node.mac, locationlog.role, ...`) so that they are not ambiguous. Although your query may work with a single table, it will not if you decide to add joins that contain column name(s) that are the same as the base table.

## 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 violations:

```
[open_violations]
description=Open violations
# The table to search from
base_table=violation
# The columns to select
columns=violation.vid as "Violation ID", violation.mac as "MAC Address",
  class.description as "Violation description", node.computername as "Hostname",
  node.pid as "Username", node.notes as "Notes", locationlog.switch_ip as "Last
  switch IP", violation.start_date as "Opened on"
# Left join node, locationlog on the MAC address and class on the violation ID
joins=<<EOT
=>{violation.mac=node.mac} node|node
=>{violation.mac=locationlog.mac} locationlog|locationlog
=>{violation.vid=class.vid} 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 violation table is *left joined* to the class, node and locationlog tables. Using that strategy we make sure all the violations 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.

## 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.

### 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.

- Violation Manager: Provides the user the ability to manage the violations (trigger, open, close) for the nodes.

## 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 both the captive portal and the admin interfaces 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**.

# Additional Information

---

For more information, please consult the mailing archives or post your questions to it. For details, see:

- [packetfence-announce@lists.sourceforge.net](mailto:packetfence-announce@lists.sourceforge.net): Public announcements (new releases, security warnings etc.) regarding PacketFence
- [packetfence-devel@lists.sourceforge.net](mailto:packetfence-devel@lists.sourceforge.net): Discussion of PacketFence development
- [packetfence-users@lists.sourceforge.net](mailto:packetfence-users@lists.sourceforge.net): User and usage discussions

# Commercial Support and Contact Information

---

For any questions or comments, do not hesitate to contact us by writing an email to: [support@inverse.ca](mailto:support@inverse.ca).

Inverse (<http://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 <http://inverse.ca/> for details.



# GNU Free Documentation License

---

Please refer to <http://www.gnu.org/licenses/fdl-1.2.txt> for the full license.

# Appendix A. Administration Tools

---

## 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 violation classes
  configreload        | reload the configuration
  fingerbank          | Fingerbank related commands
  floatingnetworkdeviceconfig | query/modify floating network devices
configuration parameters
  help                | show help for pfcmd commands
  import              | bulk import of information into the 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
  connectionprofileconfig | query/modify connection profile configuration
parameters
  reload              | rebuild fingerprint or violations 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
  violationconfig     | query/modify violations.conf configuration
parameters

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 dump

---

If you enabled Percona XtraBackup 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:

```
# xbstream -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, install qpress (available from the percona repository) and process any qp file that were extracted:

CentOS:

```
# yum install qpress --enablerepo=percona-release-`uname -m`
# for i in $(find -name "*.qp"); do qpress -vd $i $(dirname ${i}) && rm -f $i;
done
```

Debian:

```
# apt-key adv --keyserver keys.gnupg.net --recv-keys 1C4CBDCDCD2EFD2A
# echo 'deb http://repo.percona.com/apt VERSION main' >> /etc/apt/sources.list
# echo 'deb-src http://repo.percona.com/apt VERSION main' >> /etc/apt/sources.list
# apt-get update
# apt-get install qpress
# for i in $(find -name "*.qp"); do qpress -vd $i $(dirname ${i}) && rm -f $i;
done
```

Next, apply the innodb log:

```
# innobackupex --apply-log ./
```

We will now stop MySQL, move the existing data directory and replace it by the data that was extracted:



### Note

On Debian the service will be called `mysql` and on CentOS 7 `mariadb`. Make sure you adjust the commands above to your environment.

```
# 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
```

Should the service fail to start, make sure you look into the MySQL error logs.

# Optional components

---

## Blocking malicious activities with violations

---

Policy violations allow you to restrict client system access based on violations of certain policies. For example, if you do not allow P2P type traffic on your network, and you are running the appropriate software to detect it and trigger a violation for a given client, PacketFence will give that client a "blocked" page which can be customized to your wishes.

In order to be able to block malicious activities, installation and configuration of a PacketFence compatible IDS is required. PacketFence currently support Snort, Suricata and Security Onion.

### Suricata

#### Installation

Since the Suricata IDS is not packaged with the distros (except maybe Fedora, which we do not officially support), you need to build it the "old" way.

The OISF provides a really well written how-to for that. It's available here: <https://redmine.openinfosecfoundation.org/projects/suricata/wiki/CentOS5>



#### Note

To benefit the OPSWAT Metadefender Cloud integration, Suricata needs to be built with libnss / libnspr support. Make sure to use JSON output. More information on how to achieve this can be found there: <https://redmine.openinfosecfoundation.org/projects/suricata/wiki/MD5>

#### Configuration

You will need to follow the configuration below related to the Suricata configuration (Syslog).

#### OPSWAT Metadefender Cloud

It is possible to trigger violations based on threat level of downloaded files using the Metadefender Cloud integration in conjunction with the Suricata MD5 extraction feature. Without entering in the details, here are the basic steps to make it work.

First, an OPSWAT portal account is required to make use of the API. Such account can be obtained through the OPSWAT portal: <https://portal.opswat.com>.

Other requirement is a Suricata working installation built with libnss / libnspr support as described in the upper "Installation" section.

Along with the OPSWAT API key for Metadefender Cloud (they call it *License Key*) and the working Suricata installation, some configuration (PacketFence based AND Suricata based) is also required.

Assuming that all the steps for Suricata MD5 extraction have been followed, here's what to do next.

On PacketFence, under Configuration→Integration→OPSWAT Metadefender, enter your *Licence Key*.

On the Suricata server (syslog-ng is preferred due to easier and more powerful configuration. If not installed, it might be an idea):

Configure `/etc/syslog-ng/syslog-ng.conf` by adding the following to enable sending MD5 file store log entries to PacketFence:

```
### PacketFence / OPSWAT Metadefender Cloud integration
# This line specifies where the files-json.log file is located
# -> Make sure to configure the right path along with the right filename
source s_suricata_files { file("/MY_SURICATA_LOG_FILES_PATH/files-json.log"
    program_override("suricata_files") flags(no-parse)); };
# This line tells syslog-ng to send the data read to the PacketFence management
    interface IP address using UDP 514
# -> Make sure to configure the right PacketFence management interface IP address
destination d_packetfence_md5 { udp("PACKETFENCE_MGMT_IP" port(514)); };
# This line indicates syslog-ng to use the s_suricata_files source and send it to
    the d_packetfence_md5 destination
log { source(s_suricata_files); destination(d_packetfence_md5); };
```

A restart of the syslog-ng daemon is 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/suricata_files.conf` so it contains the following which will redirect Suricata MD5 file store log entries and stop further processing of current matched message:

```
if $programname == 'suricata_files' then /usr/local/pf/var/suricata_files
& ~
```

Make sure the receiving alert pipe (FIFO) exists

```
mkfifo /usr/local/pf/var/suricata_files
```

Restart the rsyslog daemon

```
service rsyslog restart
```

At this point, Suricata should be able to extract MD5 checksum of downloaded files and send the related log entry to PacketFence.

A configuration of a new *syslog parser* as well as some violations are the only remaining steps to make full usage of the OPSWAT Metadefender Cloud integration.

Configuration of a new *syslog parser* (Configuration→Integration→Syslog Parsers) should use the followings:

```
Type: suricata_md5
Alert pipe: the previously created alert pipe (FIFO) which is, in this case, /usr/local/pf/var/suricata_files
```

Configuration of a new violation can use the following trigger types:

```
Type: metadefender
Triggers ID: The scan result returned by Metadefender Cloud online
```

```
Type: suricata_md5
Trigger ID: The MD5 hash returned by Suricata
```

## Violations

In order to make PacketFence react to the Snort alerts, you need to explicitly tell the software to do so. Otherwise, the alerts will be discarded. This is quite simple to accomplish. In fact, you need to create a violation and add the Snort alert SID in the trigger section of a Violation.

PacketFence violations are configured in *Configuration→Compliance→Violations*

The example below will guide you to create a violation that will isolate device that have generated Peer-to-peer traffic and that are using Mac OSX or have a malware and are using Microsoft Windows

### Violation Definition

First you need to configure the violation definition



**New Violation**

Definition Triggers Remediation Advanced

Enabled  ON

Identifier   
Use a number above 1500000 if you want to be able to delete this violation later.

Description

Actions

Priority   
Range 1-10, with 1 the highest priority and 10 the lowest. Higher priority violations will be addressed first if a host has more than one.

Whiteisted Roles   
Nodes with the selected roles won't be affected by a violation of this type.

Close Save

Where :

- **Enabled** is whether or not the violation is currently activated (should be **ON**).
- **Identifier** is the violation ID. Any integer except 1200000-120099 which is reserved for required administration violations.
- **Description** is the user friendly description of the violation.
- **Actions** is the list of actions to be executed when this violation is raised.
  - **Unregister node** will unregister the node.
  - **Send email to owner** will email the violation details to the owner of the device. Will only work if the person has it's email field populated.
  - **Send email to admin** will email the violation details to the address specified in `[alerting].emailaddr`, using `[alerting].smtpserver`. Multiple `emailaddr` can be separated by comma.
  - **Reevaluate access** will place the device in the destination VLAN configured in the violation. It opens a violation and leaves it open. If it is not there, the violation is opened and then automatically closed.
  - **Log message** will log the violation in the log file `violation.log`.
  - **External command** will execute a command on the operating system when this violation is raised.
  - **Close a violation** will close an existing violation for this device when this one is raised.
  - **Set role** will modify the role of the device.

- **Enforce provisioning** will trigger a check of compliance on the provisioners defined for the device.
- **Priority** defines the order onto which violations should be handled should there be more than one for a device.
- **Whitelisted Roles** is the list of roles that are not affected by this violation.

## Triggers

Next, in the **Triggers** tab, you need to define the triggers that will raise the violation. In the case of this example it will be these two cases : \* A device that has generated Peer-to-peer traffic and that is using MAC OSX. \* A device that has been detected as being a rogue DHCP.

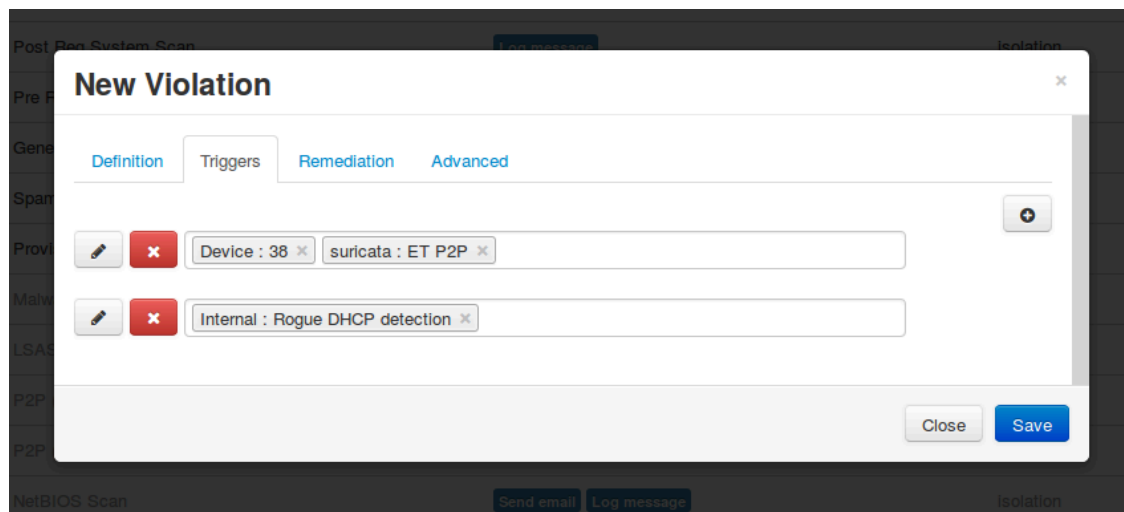
Click the + sign at the top right in order to create a new trigger, then in the dropdown select **Suricata**.

A menu will appear into which you can select **ET P2P** which will match all P2P alerts from Suricata.

Once you added this trigger, select **Device** from the dropdown and enter **38** which is the device identifier for Mac OSX.

Next hit the < button, then the + to add another trigger.

Select the type **Internal**, then in the menu that appears below it, select **Rogue DHCP detection** and click **Add**.



## Remediation

Next, in the **Remediation** tab, you can configure the behavior when a client gets isolated.

**New Violation**

Definition Triggers Remediation **Advanced**

**Auto Enable**   
 Specifies if a host can self remediate the violation (enable network button) or if they can not and must call the help desk.

**Max Enables**   
 Number of times a host will be able to try and self remediate before they are locked out and have to call the help desk. This is useful for users who just *click through* violation pages.

**Grace**  seconds minutes hours days weeks months years  
 Amount of time before the violation can reoccur. This is useful to allow hosts time (in the example 2 minutes) to download tools to fix their issue, or shutoff their peer-to-peer application.

**Dynamic Window**   
 Only works for accounting violations. The violation will be opened according to the time you set in the accounting violation (ie. You have an accounting violation for 10GB/month. If you bust the bandwidth after 3 days, the violation will open and the release date will be set for the last day of the current month).

**Window**  seconds minutes hours days weeks months years  
 Amount of time before a violation will be closed automatically. Instead of allowing people to reactivate the network, you may want to open a violation for a defined amount of time instead.

**Delay By**  seconds minutes hours days weeks months years  
 Delay before triggering the violation.

**Template**   
 HTML template the host will be redirected to while in violation. You can create new templates from the *Portal Profiles* configuration section.

**Button Text**   
 Text displayed on the violation form to hosts.

Close Save

Where :

- **Auto Enable** is whether or not the user can release the violation himself after acknowledging the message on the captive portal.
- **Max Enables** is the amount of time a user can use the **Auto Enable** functionality. After this amount of times, he will not be able to release the violation and it will have to be manually release by an administrator using the PacketFence administration interface.
- **Grace** is Amount of time before the violation can reoccur. This is useful to allow hosts time (in the example 2 minutes) to download tools to fix their issue, or shutoff their peer-to-peer application.
- **Dynamic Window** will only work for accounting violations. The violation will be opened according to the time you set in the accounting violation (ie. You have an accounting violation for 10GB/month. If you bust the bandwidth after 3 days, the violation will open and the release date will be set for the last day of the current month).
- **Window** is the amount of time before a violation will be closed automatically. Instead of allowing people to reactivate the network, you may want to open a violation for a defined amount of time instead.

- **Delay by** is the delay before triggering the violation.
- **Template** is the HTML template the host will be redirected to while in violation. You can create new templates from the Connection Profiles configuration section.
- **Button text** is the text of the button that is used when the user is releasing the violation directly from the captive portal.

## Advanced

In the **Advanced** tab you configure the destination VLAN of the device when it has the **Reevaluate access** action and its redirection URL when the user is released.

# Guests Management

---

PacketFence supports the ability to manage guests by establishing expire dates and assign different roles which will permit different accesses to the network resources.

Guests can self-register themselves using an activation code sent to their mobile phone or they can use their email address and receive an activation link to activate their network access.

PacketFence has the option to have guests sponsored their access by local staff. Once a guest requests a sponsored access an email is sent to the sponsor and the sponsor must click on a link and authenticate in order to enable his access.

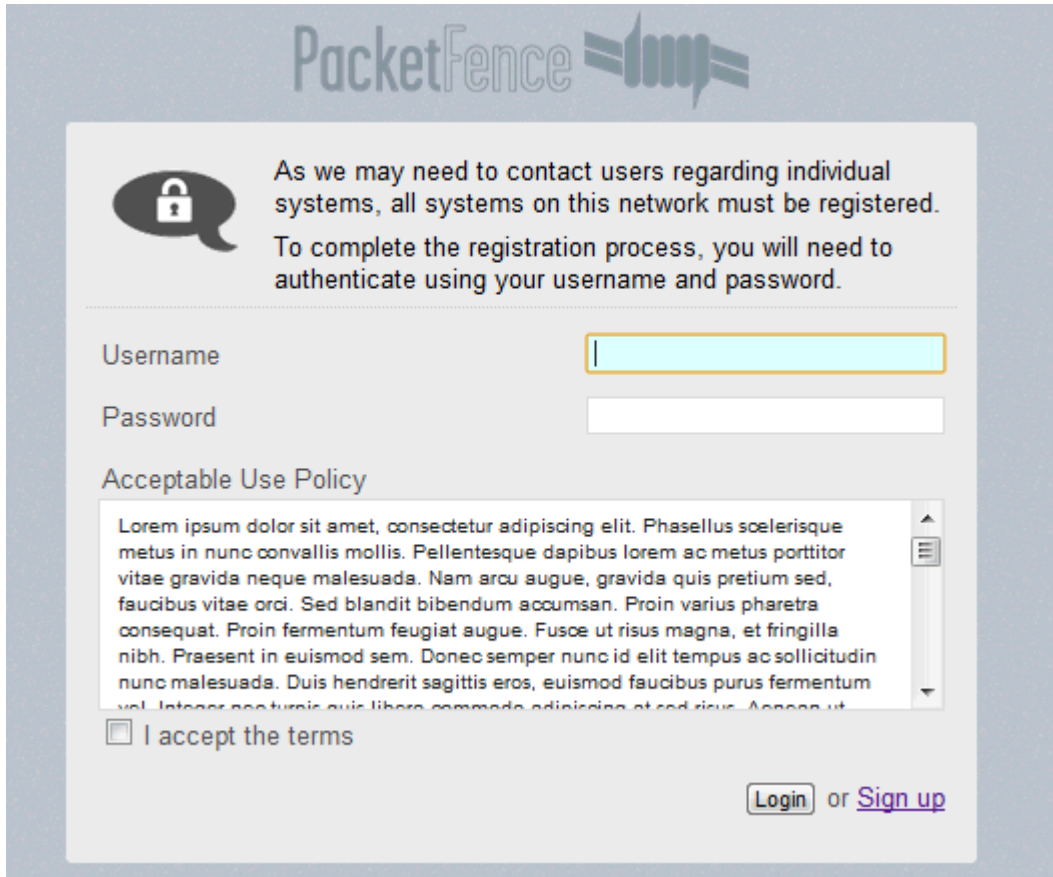
Moreover, PacketFence also has the option for guests to request their access in advance. Confirmation by email and by a sponsor are the two pre-registration techniques supported at this point.

The admin GUI allow PacketFence administrators or guests managers to create single accounts, multiple accounts using a prefix (ie.: guest1, guest2, guest3...) or import data from a CSV to create accounts. Access duration and expected arrival date are also customizable.

## Usage

### Guest self-registration

Self-registration is enabled by default. It is part of the connection profile and can be accessed on the registration page by clicking the **Sign up** link.



PacketFence

As we may need to contact users regarding individual systems, all systems on this network must be registered. To complete the registration process, you will need to authenticate using your username and password.

Username

Password

Acceptable Use Policy

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus scelerisque metus in nunc convallis mollis. Pellentesque dapibus lorem ac metus porttitor vitae gravida neque malesuada. Nam arcu augue, gravida quis pretium sed, faucibus vitae orci. Sed blandit bibendum accumsan. Proin varius pharetra consequat. Proin fermentum feugiat augue. Fusce ut risus magna, et fringilla nibh. Praesent in euismod sem. Donec semper nunc id elit tempus ac sollicitudin nunc malesuada. Duis hendrerit sagittis eros, euismod faucibus purus fermentum vel. Integer posuere quis libero, commodo adipiscing et sed risus. Aenean ut

I accept the terms

or [Sign up](#)

## Managed guests

Part of the web administration interface, the guests management interface is enabled by default. It is accessible through the **Users** → **Create** menu.

## 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>.



### 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.

## Configuration

### Access Duration

Default values are located in `/usr/local/pf/conf/pf.conf.defaults` and documentation for every settings is available in `/usr/local/pf/conf/documentations.conf`.

```
[guests_admin_registration]
access_duration_choices=1h,3h,12h,1D,2D,3D,5D
default_access_duration=12h
```

The format of the duration is as follow:

```
<DURATION><DATETIME_UNIT>[<PERIOD_BASE><OPERATOR><DURATION><DATE_UNIT>]
```

Let's explain the meaning of each parameter:

- **DURATION**: a number corresponding to the period duration.
- **DATETIME\_UNIT**: a character corresponding to the units of the date or time duration; either s (seconds), m (minutes), h (hours), D (days), W (weeks), M (months), or Y (years).
- **PERIOD\_BASE**: either F (fixed) or R (relative). A relative period is computed from the beginning of the period unit. Weeks start on Monday.
- **OPERATOR**: either + or -. The duration following the operator is added or subtracted from the base duration.
- **DATE\_UNIT**: a character corresponding to the units of the extended duration. Limited to date units (D (days), W (weeks), M (months), or Y (years)).

These parameters should be configured from the **Configuration → Advanced Access Configuration → Access Duration** section of the Web admin interface.

From the Users page of the PacketFence Web admin interface, it is possible to set the access duration of users, change their password and more.

### Guest pre-registration

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.

Finally, it is advised that you read the whole guest self-registration section since pre-registration is simply a twist of the self-registration process.



#### Caution

A valid MTA configured in PacketFence is needed to correctly relay emails related to the guest module. If `localhost` is used as `smtpserver`, make sure that a MTA is installed and configured on the server.