PacketFence – version 2.1.0

Administration Guide

Copyright © 2008-2011 Inverse inc. (http://inverse.ca)

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".

Version 2.1.0 – March 2011

Contents

Chapter 1 **About this Guide** **6**

Other sources of information 6

Chapter 2 **Introduction** **7**

Features 7

Network Integration 9

Components 10

Chapter 3 **System Requirements** **11**

Assumptions 11

Minimum Hardware Requirements 12

Operating System Requirements 13

Chapter 4 **Installation** **14**

OS Installation 14

Software Download 15

Software Installation 15

Chapter 5 **Configuration** **16**

First Step 16

Web-based Administration Interface 16

Global configuration file (pf.conf) 17

Apache Configuration 17

SELinux 17

Authentication (flat file, LDAP/AD, RADIUS) 18

Network Device Definition (switches.conf) 19

Default VLAN assignment 21

DHCP and DNS Server Configuration (networks.conf) 23

Production DHCP access 26

Routed Networks 29

FreeRADIUS Configuration 31
Starting PacketFence Services 39
Log files 39

Chapter 6 Configuration by example 41
Assumptions 41
Network Interfaces 43
Switch Setup 44
switches.conf 45
pf.conf 45
networks.conf 47
DHCP and DNS Server Configuration 47
FreeRADIUS 48

Chapter 7 Optional components 50
Blocking malicious activities with violations 50
Conformity Scan (Nessus) 54
Oinkmaster 56
Floating Network Devices 57

Chapter 8 Operating System Best Practices 59
Iptables 59
Log Rotations 59
High availability 60

Chapter 9 Performance optimization 69
MySQL optimizations 69
Captive portal optimizations 73

Chapter 10 Frequently Asked Questions 74
Services 74
Web interface 74
Advanced features 76
Switches 79
Troubleshooting 81

Chapter 11 Technical introduction to VLAN mode 83
Introduction 83

More on SNMP traps VLAN isolation 85

Chapter 12 Appendix A: Administration Tools 87

pfcmd 87

pfcmd_vlan 88

Web Admin GUI 91

Chapter 13 Appendix B: Legacy FreeRADIUS 1.x configuration 92

Chapter 14 Additional Information 96

Chapter 15 Commercial Support and Contact Information 97

Chapter 16 GNU Free Documentation License 98

About this Guide

This guide will walk you through the installation and the day to day administration of the PacketFence solution. It covers mainly VLAN assignment mode.

The instructions are based on version 2.1.0 of PacketFence.

The latest version of this guide is available at http://www.packetfence.org/documentation/

Other sources of information

Network Devices Configuration Guide – Covers switch, controllers and access points configuration.

Developers Guide – Covers captive portal customization, VLAN management customization and instructions for supporting new hardware.

For the list of noteworthy changes since the last release see the `NEWS` file.

For a list of compatibility related changes and notes about upgrading see the `UPGRADE` file.

For more details and developer visible changes see the `ChangeLog` file.

These files are included in the package and release tarballs.

Chapter 2

Introduction

PacketFence is a fully supported, trusted, Free and Open Source network access control (NAC) system. Boosting an impressive feature set including a captive-portal for registration and remediation, centralized wired and wireless management, 802.1X support, layer-2 isolation of problematic devices, integration with the Snort IDS and the Nessus vulnerability scanner; PacketFence can be used to effectively secure networks - from small to very large heterogeneous networks.

Features

- Out of band

  PacketFence's operation is completely out of band which allows the solution to scale geographically and to be more resilient to failures.

- Voice over IP (VoIP) support.

  Also called IP Telephony (IPT), VoIP is fully supported (even in heterogeneous environments) for multiple switch vendors (Cisco, Edge-Core, HP, LinkSys, Nortel Networks and many more).

- 802.1X

  802.1X wireless and wired is supported through a FreeRADIUS module.

- Wireless integration

  PacketFence integrates perfectly with wireless networks through a FreeRADIUS module. This allows you to secure your wired and wireless networks the same way using the same user database and using the same captive portal, providing a consistent user experience. Mixing Access Points (AP) vendors and Wireless Controllers is supported.

- Registration

  PacketFence supports an optional registration mechanism similar to "captive portal" solutions. Contrary to most captive portal solutions, PacketFence remembers users who

previously registered and will automatically give them access without another authentication. Of course, this is configurable. An Acceptable Use Policy can be specified such that users cannot enable network access without first accepting it.

- Detection of abnormal network activities

  Abnormal network activities (computer virus, worms, spyware, traffic denied by establishment policy, etc.) can be detected using local and remote Snort sensors. Beyond simple detection, PacketFence layers its own alerting and suppression mechanism on each alert type. A set of configurable actions for each violation is available to administrators.

- Proactive vulnerability scans

  Nessus vulnerability scans can be performed upon registration, scheduled or on an ad-hoc basis. PacketFence correlates the Nessus vulnerability ID's of each scan to the violation configuration, returning content specific web pages about which vulnerability the host may have.

- Isolation of problematic devices

  PacketFence supports several isolation techniques, including VLAN isolation with VoIP support (even in heterogeneous environments) for multiple switch vendors.

- Remediation through a captive portal

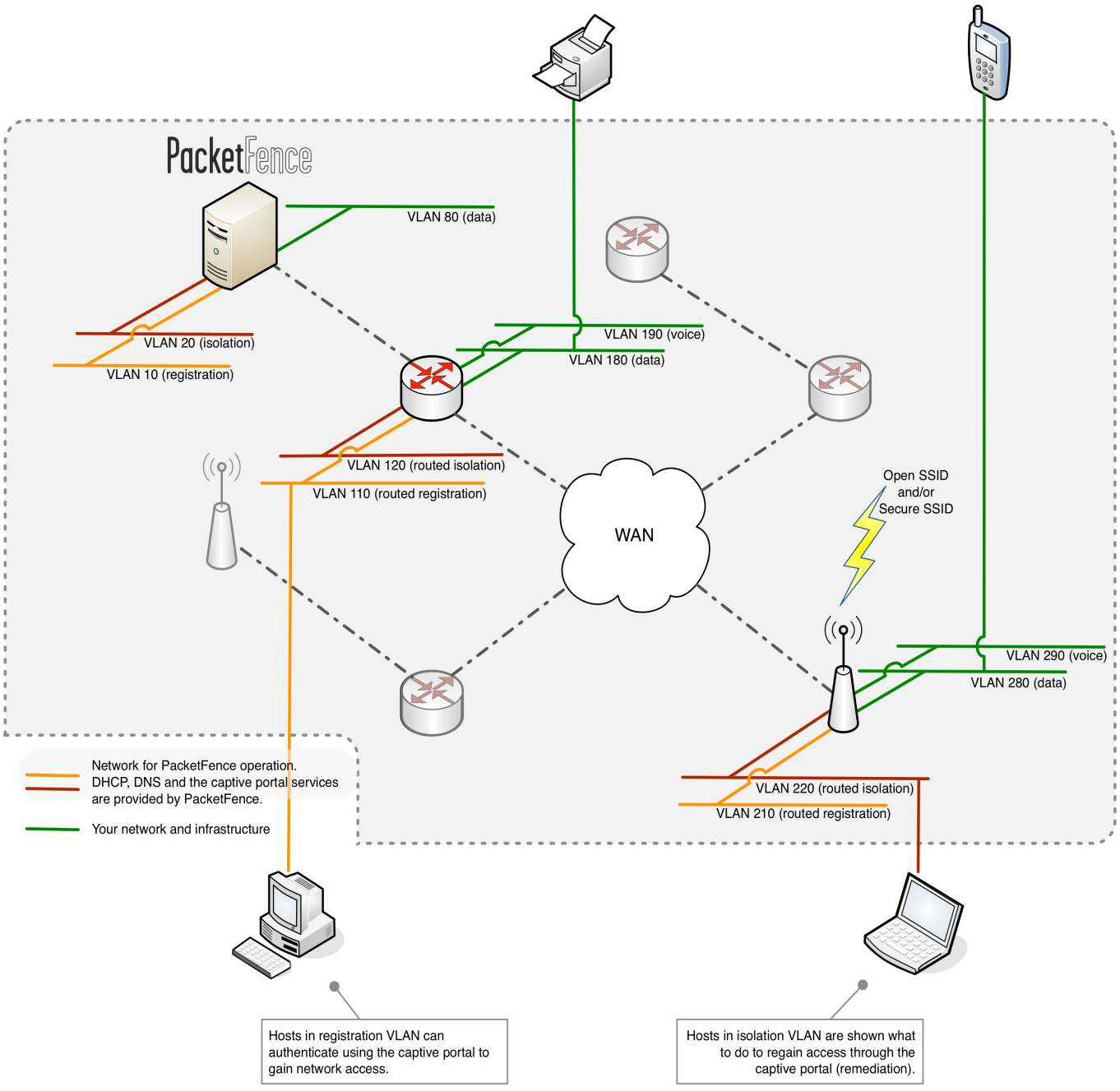
  Once trapped, all network traffic is terminated by the PacketFence system. Based on the node's current status (unregistered, open violation, etc), the user is redirected to the appropriate URL. In the case of a violation, the user will be presented with instructions for the particular situation he/she is in reducing costly help desk intervention.

- Command-line and Web-based management

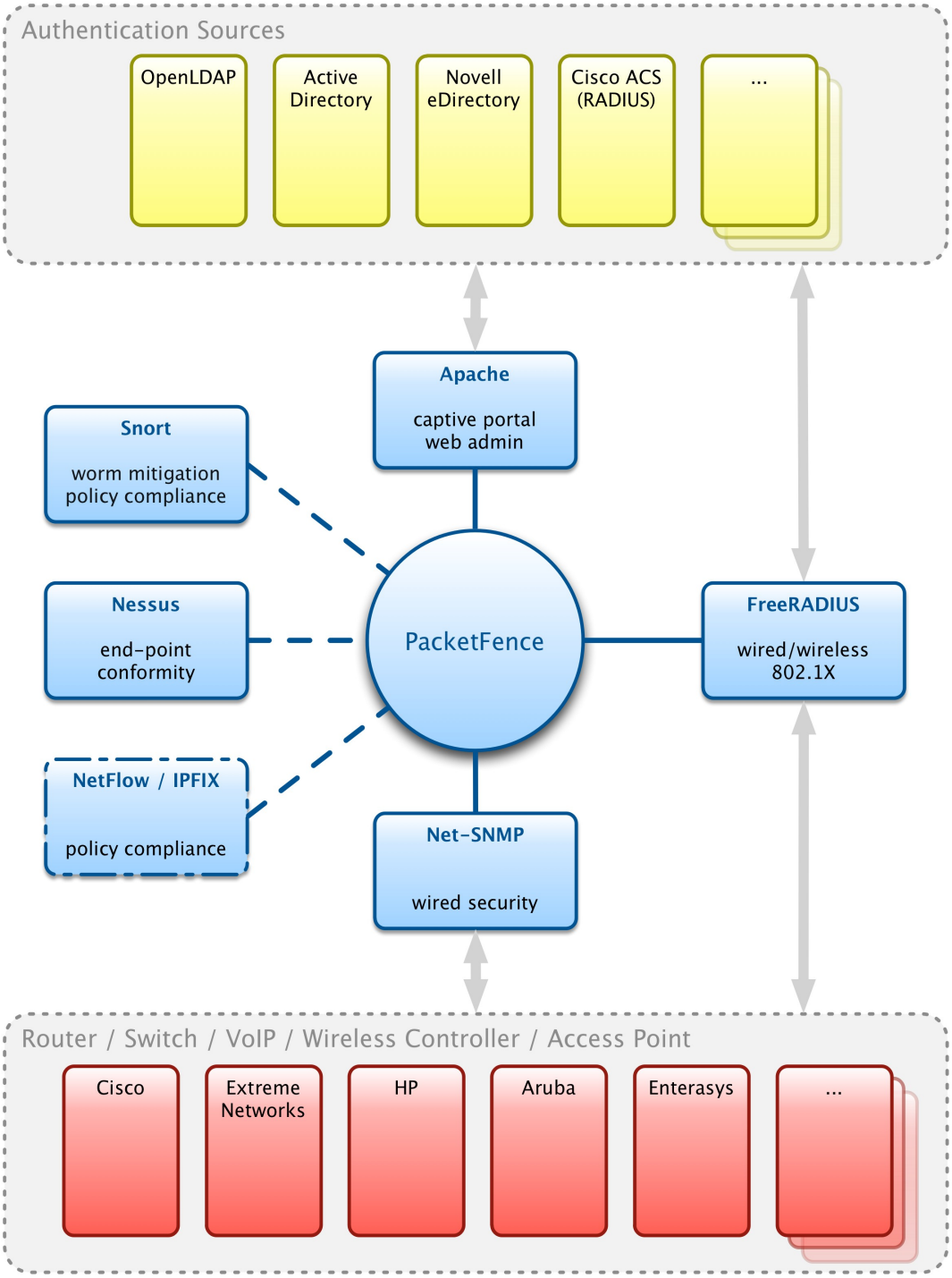
  Web-based and command-line interfaces for all management tasks.

PacketFence is developed by a community of developers located mainly in North America. More information can be found on http://www.packetfence.org

Network Integration



Components


Authentication Sources
OpenLDAP
Active Directory
Novell eDirectory
Cisco ACS (RADIUS)
...
Apache
captive portal
web admin
Snort
worm mitigation
policy compliance
Nessus
end-point
conformity
NetFlow / IPFIX
policy compliance
PacketFence
FreeRADIUS
wired/wireless
802.1X
Net-SNMP
wired security
Router / Switch / VoIP / Wireless Controller / Access Point
Cisco
Extreme Networks
HP
Aruba
Enterasys
...

System Requirements

Assumptions

PacketFence reuses many components in an infrastructure. Thus, it requires the following ones:

- Database server (MySQL)
- Web server (Apache)

Depending on your setup you may have to install additional components like:

- DHCP server (ISC DHCP)
- DNS server (BIND)
- RADIUS server (FreeRADIUS)
- NIDS (Snort)

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 component and GNU/Linux is required to install PacketFence. If you miss some of those required components, please refer to the appropriate documentation and proceed with the installation of these requirements before continuing with this guide.

The following table provides recommendations for the required components, together with version numbers :

MySQL server	MySQL 4.1 or 5.1
Web server	Apache 2.2
DHCP server	DHCP 3
DNS server	BIND 9
RADIUS server	FreeRADIUS 2
Snort	Snort 2.8

More recent versions of the software mentioned above can also be used.

Minimum Hardware Requirements

The following table provides hardware recommendations for the server and desktops :

Server	■ Intel or AMD CPU 3 GHz ■ 2048 MB of RAM ■ 20 GB of disk space (RAID 1) ■ 1 Network card ■ + 1 for high-availability ■ + 1 for intrusion detection

Operating System Requirements

PacketFence supports the following operating systems on the i386 or x86_64 architectures:

- Red Hat Enterprise Linux 5.x Server

  With the EPEL (Extra Packages for Enterprise Linux) repository enabled. It is **disabled** by default. Here are instructions to enable that repository: http://fedoraproject.org/wiki/EPEL/FAQ#howtouse

- Community ENTerprise Operating System (CentOS) 5.x

  With the `extras` yum repository enabled. It is enabled by default.

Make sure that you can install additional packages from your standard distribution. For example, if you are using Red Hat Enterprise Linux 5, you have to be subscribed to the Red Hat Network before continuing with the PacketFence software installation.

Other distributions such as Debian, Fedora and Gentoo are known to work but this document won't cover them.

Services start-up

PacketFence takes care of handling the operation of the following services:

- Web server
- DHCP server
- DNS server
- Snort (Network IDS)

Make sure that all the other services are automatically started by your operating system!

Installation

This section will guide you through the installation of PacketFence together with its dependencies.

OS Installation

Install your distribution with minimal installation and no additional packages. Then:

- Enable Firewall
- Disable SELinux

Make sure your system is up to date and your yum database is updated:

```
yum update
```

Some PacketFence dependencies are available through the DAG repository (http://dag.wieers.com/) so you need to configure YUM to use it.

First import the DAG RPM GPG key:

```
rpm -import http://dag.wieers.com/rpm/packages/RPM-GPG-KEY.dag.txt
```

Then install the latest version of the RPMForge package (http://dag.wieers.com/rpm/packages/rpmforge-release/):

```
rpm -i rpmforge-release-0.5.2-2.el5.rf.i386.rpm
```

Disable the repository by default. In the /etc/yum.repos.d/rpmforge.repo, set enabled to 0 under the rpmforge section:

```
enabled=0
```

Software Download

Starting with 1.8.5, PacketFence is now provided as an RPM repository for Red Hat Enterprise Linux 5 / CentOS 5 instead of a single RPM file.

This repository contains all required dependencies to install PacketFence. This provides numerous advantages:

- very easy installation
- everything is packaged as RPM (no more CPAN hassle)
- easy upgrade

Software Installation

In order to use the repository, just create a file named `/etc/yum.repos.d/PacketFence.repo` with the following content:

```
[PacketFence]
name=PacketFence Repository
baseurl=http://inverse.ca/downloads/PacketFence/RHEL5/$basearch
gpgcheck=0
enabled=0
```

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

```
yum groupinstall --enablerepo=PacketFence,rpmforge “Packetfence-complete”
```

Or, if you prefer, to install only the core PacketFence without all the external services, you can use :

```
yum install --enablerepo=PacketFence,rpmforge packetfence
```

Execute the installer at `/usr/local/pf/installer.pl` and follow the instructions.

Once completed, PacketFence will be fully installed on your server. You are now ready to configure it.

Configuration

In this section, you'll learn how to configure PacketFence with VLAN mode. PacketFence will use MySQL, Apache, ISC DHCP, ISC DNS. As previously mentioned, we assume that those components run on the same server on which PacketFence is being installed.

First Step

In order to properly begin the configuration of PacketFence, we strongly recommend to execute the configuration script located at `/usr/local/pf/configurator.pl`. This script will guide you through the process of creating a working PacketFence configuration file that is suitable to your needs.

The script will give you different avenues for configuration templates, either for ARP mode or VLAN mode. Depending on what you want to achieve, we strongly recommend that you start with the "VLAN isolation mode with Registration and Detection (snort)" template (option t template 8). Hence, the script will ask some more information about your network infrastructure, like the DNS servers, and the DHCP servers address, etc.

Keep in mind that the resulting PacketFence configuration will be located in `/usr/local/pf/conf/pf.conf`, and it can always be adjusted by hand afterward.

Web-based Administration Interface

PacketFence provides a web-based administration interface for easy configuration and operational management.

In order to access the interface you need to create an administrator and a web services account.

You need to encrypt the new password in the `admin.conf` file with `htpasswd`:

```
htpasswd /usr/local/pf/conf/admin.conf admin
```

Then enter the new password twice.

Then again for webservice:

```
htpasswd /usr/local/pf/conf/admin.conf webservice
```

Then enter the new password twice. Use a very strong password. You will never have to enter it

more than once.

Once PacketFence is started, administration interface is available at: https://<hostname>:1443/

Global configuration file (pf.conf)

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.

Apache Configuration

The PacketFence configuration for Apache is located in `/usr/local/pf/conf/templates/httpd.conf`.

Upon PacketFence installation, a default configuration file is created which is suitable for most configurations. SSL is enabled by default to secure access.

If you used the installer.pl script, you should have self-signed SSL certificates in `/usr/local/pf/conf/ssl` (server.key and server.crt). Those certificates can be replaced anytime by your 3rd-party or existing wildcard 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).

SELinux

Even if this feature may be wanted by some organizations, PacketFence will not run properly if SELinux is set to enforced. You will need to explicitly disable it in the `/etc/selinux/config` file.

Authentication (flat file, LDAP/AD, RADIUS)

PacketFence can authenticate users that register devices via the captive-portal using a flat file, an LDAP (or Active Directory) server or a RADIUS server.

Flat file

By default, PacketFence looks into `/usr/local/pf/conf/user.conf` to find users allowed to register devices. If you want to use a different file, edit `/usr/local/pf/conf/authentication/local.pm` and change the following parameter :

```
my $passwdFile = '/usr/local/pf/conf/user.conf';
```

You need to encrypt the password of each user with htpasswd like this :

```
htpasswd /usr/local/pf/conf/user.conf newuser
```

LDAP / Active Directory (AD)

Edit `/usr/local/pf/conf/authentication/ldap.pm` and make the necessary changes to the following parameters :

```
my $LDAPUserBase = "ou=People,dc=domain,dc=org";
my $LDAPUserKey = "uid";
my $LDAPUserScope = "one";
my $LDAPBindDN = "cn=ldapuser,dc=domain,dc=org";
my $LDAPBindPassword = "password";
my $LDAPServer = "127.0.0.1";
```

RADIUS

Edit `/usr/local/pf/conf/authentication/radius.pm` and make the necessary changes to the following parameters:

```
my $RadiusServer = 'localhost';
my $RadiusSecret = 'testing123';
```

Selecting an Authentication Method

To configure authentication set the `[registration].auth` option in `/usr/local/pf/conf/pf.conf`:

```
auth=local,ldap,radius
```

If more than one method are specified, PF will display a pull-down list to allow users to select the preferred authentication method.

Network Device Definition (switches.conf)

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 in the Web Administration panel under Configuration -> Switches.

This files contains a default section including:

- List of VLANs managed by PacketFence
- Default SNMP read/write communities for the switches
- Default working mode (see note about working mode below)

and a switch section for each switch (managed by PacketFence) including:

- Switch IP
- Switch vendor/type
- Switch uplink ports (trunks and non-managed ports)
- per-switch re-definition of the vlans (if required)

Working modes

There are three different working modes:

- 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.

CAUTION: **If [trapping.testing] (see pf.conf) is enabled, PacketFence will consider that all the switches are in test mode whatever the working mode is!**

SNMP v1, v2c and v3

PacketFence uses SNMP to communicate with most switches. Starting with 1.8, PacketFence now supports SNMP v3. You can use SNMP v3 for communication in both directions: from the switch to PacketFence and from PacketFence to the switch.

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

PackeFence needs sometimes to establish an interactive command-line session with a switch. This can be done using Telnet. Starting with 1.8, you can now use SSH. In order to do so, edit the switch config 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 -> Switches.

Web Services Interface

PackeFence 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
```

Note: as of PacketFence 1.9.1 few switches require Web Services configuration in order to work. It can also be done through the Web Administration Interface under Configuration -> Switches.

Default VLAN assignment

The default VLAN assignment technique used in PacketFence is a per-switch one. The correct default VLAN for a given MAC is the `normalVlan` variable of the switch where the MAC is connected or the `[default] normalVlan` if the switch doesn't specify a `normalVlan`.

This allows you to do easy per-building VLAN segmentation.

If you need more flexibility (per SSID, per node category, etc.) take a look at the "I need more flexible VLAN assignment" section under Advanced features.

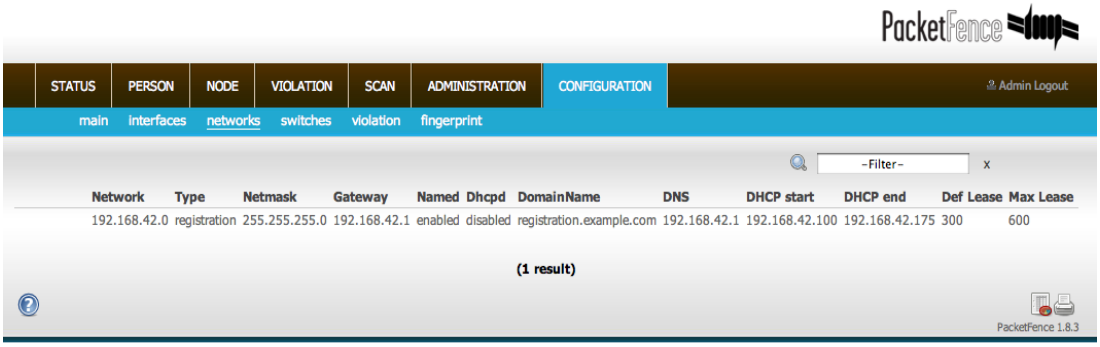
DHCP and DNS Server Configuration (networks.conf)

PacketFence automatically generates the DHCP and DNS configuration files for Registration and Isolation VLANs. This is done when executing the configurator script (see the General Configuration section).

If you want to perform the configuration manually, make sure that `pf.conf`'s vlan section has the following values set:

```
[vlan]
dhcpd=enabled
named=enabled
```

The Registration and Isolation networks information is accessible through the GUI in Administration -> Networks:



- network: Network subnet
- netmask: Network mask
- gateway: PacketFence IP address in this network
- pf_gateway: used only with routed networks; IP address of the router in this network (This is used to create static routes to the routed networks). See the Routed Networks section)
- domain-name: DNS name
- dns: PacketFence IP address in this network
- dhcp_start: starting IP address of the DHCP scope
- dhcp_end: ending IP address of the DHCP scope
- dhcp_default_lease_time: default DHCP lease time
- dhcp_max_lease_time: maximum DHCP lease time
- type: registration or isolation
- named: Is PacketFence the DNS for this network ? (Enabled/Disabled) set it to

enabled

- dhcpd: Is PacketFence the DHCP server for this network ? (Enabled/Disabled) set it to enabled

When starting PacketFence generates the DHCP and DNS configuration files by reading the information provided in `networks.conf`:

The DHCP configuration file is generated to `conf/dhcpd.conf` using `conf/templates/dhcpd_vlan.conf` as a template.

The DNS configuration files are generated this way:

- `conf/named.conf` generated from `conf/templates/named_vlan.conf`
- `conf/named/named-registration.ca` generated from `conf/templates/named-registration.ca`
- `conf/named/named-isolation.ca` generated from `conf/templates/named-isolation.ca`

Until feature #1105 is implemented, you will need to manually configure your PacketFence's Registration and Isolation network IP addresses in `conf/templates/named-registration.ca` and `conf/templates/named-isolation.ca`. Check for the underlined portions of the files below.

conf/templates/named-registration.ca

```
; This file is generated from a template at %%template%%
; Any changes made to this file will be lost on restart

$TTL 3600
. IN SOA %%hostname%%. %%incharge%% (
    2009020901  ; serial
    10800       ; refresh
    3600        ; retry
    604800      ; expire
    86400       ; default_ttl
)

       IN      NS      %%hostname%%.
*.     IN      A       192.168.2.1
       IN      MX      5       %%hostname%%.

1.2.168.192.in-addr.arpa.     IN      PTR       %%hostname%%
```

conf/templates/named-isolation.ca

```
; This file is generated from a template at %%template%%
; Any changes made to this file will be lost on restart

$TTL 3600
. IN SOA %%hostname%%. %%incharge%% (
    2009020901  ; serial
    10800       ; refresh
    3600        ; retry
```

```
    604800      ; expire
    86400       ; default_ttl
)

       IN      NS      %%hostname%%.
*.     IN      A       192.168.3.1
       IN      MX      5       %%hostname%%.

1.3.168.192.in-addr.arpa.     IN      PTR       %%hostname%%
```

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 and isolation VLANs since PacketFence acts as the DHCP server in these networks.

IP Helpers (recommended)

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. 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 `pfdhcplistener` daemon.

Make sure that no DHCP Server are running on the interface where you are sending the requests otherwise PacketFence might try to reply to the DHCP requests which would be a bad thing.

DHCP Span

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 for the span (remote or local) and adding in PacketFence the proper interface statement at the operating system level and in `pf.conf`.

`/etc/sysconfig/network-scripts/ifcfg-eth1`:

```
DEVICE=eth2
ONBOOT=yes
```

```
BOOTPROTO=none
```

Add to `pf.conf`: (IP are not important they are there only so that PacketFence will start)

```
[interface eth2]
mask=255.255.255.0
type=dhcplistener
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 span 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 by stating that it is an internal VLAN.

```
[interface eth0.1010]
mask=255.255.255.0
type=internal
gateway=10.0.101.1
ip=10.0.101.4
```

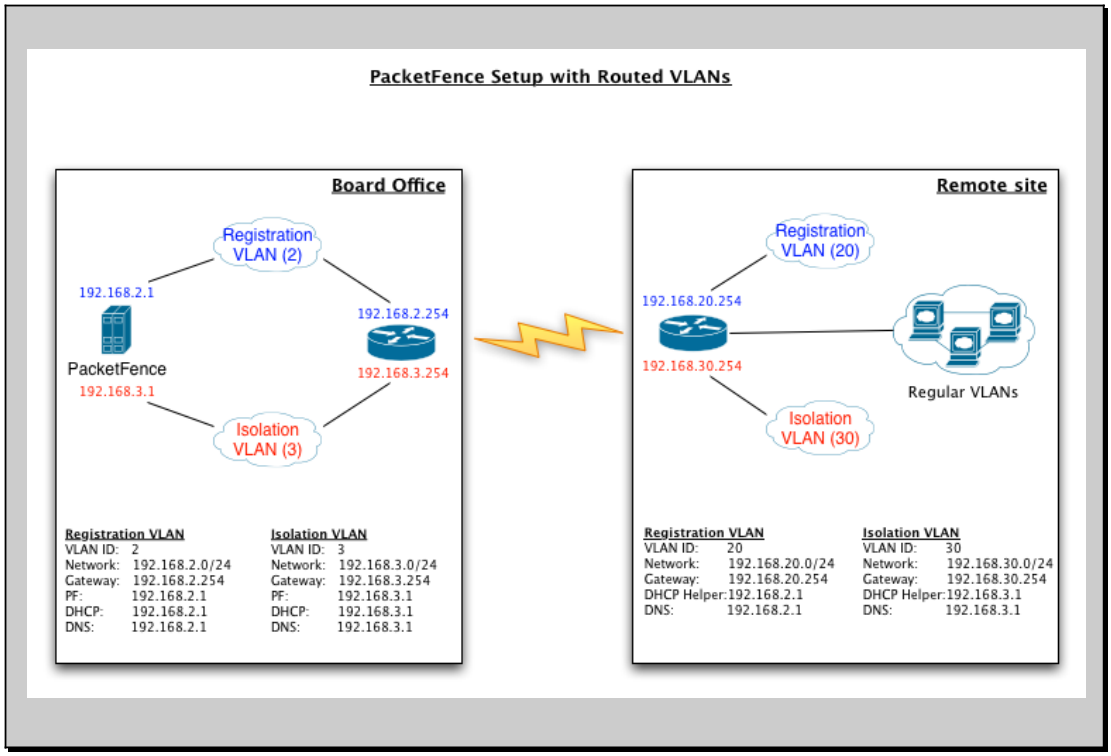
Repeat the above for all your production VLANs then restart PacketFence.

Host production DHCP on PacketFence

It's an option. Just modify `conf/templates/dhcpd_vlan.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.

Routed Networks

If your isolation and registration networks are not spanned 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 provide 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. Then make sure you followed the instructions in the DHCP and DNS Server Configuration (networks.conf) for your locally accessible network.

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`).

If we consider the network architecture illustrated in the above schema, `conf/networks.conf` will look like this:

```
[192.168.2.0]
netmask=255.255.255.0
gateway=192.168.2.1
pf_gateway=
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=registration
named=enabled
dhcpd=enabled

[192.168.3.0]
netmask=255.255.255.0
gateway=192.168.3.1
pf_gateway=
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=isolation
named=enabled
dhcpd=enabled

[192.168.20.0]
netmask=255.255.255.0
gateway=192.168.20.254
pf_gateway=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=registration
named=enabled
dhcpd=enabled

[192.168.30.0]
netmask=255.255.255.0
gateway=192.168.30.254
pf_gateway=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=isolation
named=enabled
dhcpd=enabled
```

FreeRADIUS 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 (Wireless 802.1X) , MAC authentication and Wired 802.1X all requires a RADIUS server to authenticate the users and the devices, and then to push the proper VLAN to the network equipment. We strongly recommend that you install freeRADIUS even if you plan not to use the feature now.

Install the following packages:

- freeradius2
- freeradius2-perl
- freeradius2-utils

/etc/raddb/clients.conf

Replace <...> with values useful to you. You need one client entry per network device.

```
client <useful_device_name> {
     ipaddr       = <network_device_ip_address>
     secret       = <radius secret>
}
```

/etc/raddb/modules/perl

Add the following lines to the modules{} section:

```
perl {
        module = ${confdir}/packetfence.pm
}
```

/etc/raddb/sites-enabled/default

Make sure the authorize{}, authenticate{} and post-auth{} sections look like this:

```
authorize {
     preprocess
     eap {
```

```
            ok = return
      }
      files
      expiration
      logintime
      perl
}

authenticate {
      Auth-Type MS-CHAP {
            mschap
      }
      eap
}

post-auth {
      perl
}
```

/etc/raddb/sites-enabled/inner-tunnel

Make sure the authorize{}, authenticate{} and post-auth{} sections look like this:

```
authorize {
      preprocess
      eap {
            ok = return
      }
      files
      expiration
      logintime
      perl
}

authenticate {
      Auth-Type MS-CHAP {
            mschap
      }
      eap
}

post-auth {
      perl
}
```

/etc/raddb/users

Add the following lines where we define that non EAP-messages should, by default, lead to an authentication acceptation.

```
DEFAULT EAP-Message !* "", Auth-Type := Accept
```

Comment or delete all other statements.

/etc/raddb/packetfence.pm

Make sure to set the required configuration parameters on top of the file. Set the password to the account previously created under the Web-based Administration Interface section.

```
# FreeRADIUS to PacketFence communications (SOAP Server settings)
WS_USER     => 'webservice',
WS_PASS     => 'password',
```

Optional: Wired or Wireless 802.1X configuration

Generate cryptographic material for the EAP tunnel (802.1X) to work. Run as root:

```
cd /etc/raddb/certs
make
```

/etc/raddb/eap.conf

Make sure this file looks like:

```
eap {
     default_eap_type = peap
     timer_expire     = 60
     ignore_unknown_eap_types = no
     cisco_accounting_username_bug = no
     max_sessions = 2048

     md5 {
     }
     tls {
            certdir = ${confdir}/certs
            cadir = ${confdir}/certs
            private_key_file = /usr/local/pf/conf/ssl/server.key
            certificate_file = /usr/local/pf/conf/ssl/server.crt
            dh_file = ${certdir}/dh
            random_file = ${certdir}/random
            cipher_list = "DEFAULT"
            make_cert_command = "${certdir}/bootstrap"
            cache {
                  enable = no
                   lifetime = 24 # hours
                   max_entries = 255
            }
     }
     ttls {
            default_eap_type = md5
            copy_request_to_tunnel = yes
            use_tunneled_reply = yes
            virtual_server = "inner-tunnel"
     }
     peap {
            default_eap_type = mschapv2
            copy_request_to_tunnel = yes
            use_tunneled_reply = yes
            virtual_server = "inner-tunnel"
     }
     mschapv2 {
     }
}
```

Option 1: Authentication against Active Directory (AD)

Replace `/etc/raddb/modules/mschap` with the following configuration:

```
mschap {
        use_mppe = yes
        require_encryption = yes
        require_strong = yes
        with_ntdomain_hack = yes
        ntlm_auth = "/usr/bin/ntlm_auth --request-nt-key --username=%{%
{Stripped-User-Name}:-%{mschap:User-Name:-None}} --challenge=%
{mschap:Challenge:-00} –nt-response=%{mschap:NT-Response:-00}"
}
```

Samba / Kerberos / Winbind

Install SAMBA. You can either use the sources or use the package for your OS. For CentOS, you can use :

```
wget ftp://ftp.sernet.de/pub/samba/3.5/centos/5/x86_64/samba3-3.5.6-
43.el5.x86_64.rpm
wget ftp://ftp.sernet.de/pub/samba/3.5/centos/5/x86_64/samba3-client-
3.5.6-43.el5.x86_64.rpm
wget ftp://ftp.sernet.de/pub/samba/3.5/centos/5/x86_64/samba3-utils-
3.5.6-43.el5.x86_64.rpm
wget ftp://ftp.sernet.de/pub/samba/3.5/centos/5/x86_64/samba3-winbind-
3.5.6-43.el5.x86_64.rpm
wget ftp://ftp.sernet.de/pub/samba/3.5/centos/5/x86_64/libwbclient0-
3.5.6-43.el5.x86_64.rpm

yum install ./samba*.rpm --nogpgcheck
```

Note: If you have Windows 7 PCs in your network, you need to use SAMBA version 3.5.0 or greater)

When done with the samba install, you need to modify `/etc/krb5.conf`. Here is an example for the DOMAIN.NET domain :

```
[logging]
 default = FILE:/var/log/krb5libs.log
 kdc = FILE:/var/log/krb5kdc.log
 admin_server = FILE:/var/log/kadmind.log

[libdefaults]
```

```
 default_realm = DOMAIN.NET
 dns_lookup_realm = false
 dns_lookup_kdc = false
 ticket_lifetime = 24h
 forwardable = yes

[realms]
 DOMAIN.NET = {
  kdc = adserver.domain.net:88
  admin_server = adserver.domain.net:749
  default_domain = domain.net
 }

[domain_realm]
 .domain.net = DOMAIN.NET
 domain.net = DOMAIN.NET

[appdefaults]
 pam = {
   debug = false
   ticket_lifetime = 36000
   renew_lifetime = 36000
   forwardable = true
   krb4_convert = false
 }
```

Next, edit /etc/samba/smb.conf. Again, here is an example for our DOMAIN.NET

```
[global]
workgroup = DOMAIN
server string = pf_server_name
interfaces = 192.168.1.2/24
security = ADS
passdb backend = tdbsam
realm = DOMAIN.NET
encrypt passwords = yes
winbind use default domain = yes
client NTLMv2 auth = yes
preferred master = no
load printers = no
cups options = raw
```

After that, you need to start samba, and join the machine to the domain

```
service smb start
chkconfig --level 345 smb on
net ads join -U administrator
```

Finally, start winbind, and test the setup using ntlm_auth

```
service winbind start
chkconfig --level 345 winbind on
chgrp radiusd /var/lib/samba/winbindd_privileged/
ntlm_auth –username myDomainUser
```

Option 2: Local Authentication

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

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

Option 3: Authentication against OpenLDAP

```
To be contributed...
```

Tests

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

```
# radtest dd9999 Abcd1234 localhost 12 testing123

Sending Access-Request of id 74 to 127.0.0.1 port 1812
     User-Name = "dd9999"
     User-Password = "Abcd1234"
     NAS-IP-Address = 255.255.255.255
     NAS-Port = 12
rad_recv: Access-Accept packet from host 127.0.0.1:1812, id=74, length=20
```

Debug

First, check the syslog, this is where the PacketFence module logs. Syslog messages are usually stored in `/var/log/messages`.

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

```
# radiusd -X
```

Startup

Don't forget that if you rely on FreeRADIUS for your particular setup then you need to add the `radiusd` service to the list of services automatically started by the operating system on boot.

Starting PacketFence Services

Once PacketFence is fully installed and configured, start the services using the following command :

```
service packetfence start
```

You may verify using the `chkconfig` command that the PacketFence service is automatically started at boot time.

Log files

Here are the most important PacketFence log files:

- `/usr/local/pf/logs/packetfence.log` – PacketFence Core Log
- `/usr/local/pf/logs/access_log` – Apache – Captive Portal Access Log
- `/usr/local/pf/logs/error_log` – Apache – Captive Portal Error Log
- `/usr/local/pf/logs/admin_access_log` – Apache – Web Admin/Services Access Log
- `/usr/local/pf/logs/admin_error_log` – Apache – Web Admin/Services Error Log
- `/usr/local/pf/logs/admin_debug_log` – Apache – Web Admin Debug 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 log 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.

Configuration by example

Here is an end-to-end sample configuration of PacketFence in VLAN mode.

Assumptions

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

- There are two different types of manageable switches in our network: Cisco Catalyst 2900XL and Cisco Catalyst 2960
- VLAN 1 is the “regular” VLAN
- VLAN 2 is the registration VLAN (unregistered devices will be put in this VLAN)
- VLAN 3 is the isolation VLAN (isolated devices will be put in this VLAN)
- VLANs 2 and 3 are spanned throughout the network
- VLAN 4 is the MAC detection VLAN (empty VLAN)
- VLAN 4 must be defined on all the switches that do not support port-security (in our example Catalyst 2900XL do not support port-security with static MAC address). No need to put it in the trunk port.
- We want to isolate computers using Limewire (peer-to-peer software)
- We use Snort as NIDS
- The traffic monitored by Snort is spanned on eth1
- The DHCP server on the PacketFence box that will take care of IP address distribution in VLANs 2 and 3
- The DNS server on the PacketFence box that will take care of domain resolution in VLANs 2 and 3
- The network setup looks like this:

VLAN ID	VLAN Name	Subnet	Gateway	PacketFence Address
1	Normal	192.168.1.0/24	192.168.1.1	192.168.1.5
2	Registration	192.168.2.0/24	192.168.2.1	192.168.2.1
3	Isolation	192.168.3.0/24	192.168.3.1	192.168.3.1
4	Mac Detection			

© 2008-2011 Inverse inc.

100	Voice

Network Interfaces

Here are the NICs startup scripts on PacketFence:

- `/etc/sysconfig/network-scripts/ifcfg-eth0`

```
DEVICE=eth0
BROADCAST=192.168.1.255
IPADDR=192.168.1.5
NETMASK=255.255.255.0
NETWORK=192.168.1.0
ONBOOT=yes
TYPE=Ethernet
```

- `/etc/sysconfig/network-scripts/ifcfg-eth0.2`

```
DEVICE=eth0.2
ONBOOT=yes
BOOTPROTO=static
IPADDR=192.168.2.1
NETMASK=255.255.255.0
VLAN=yes
```

- `/etc/sysconfig/network-scripts/ifcfg-eth0.3`

```
DEVICE=eth0.3
ONBOOT=yes
BOOTPROTO=static
IPADDR=192.168.3.1
NETMASK=255.255.255.0
VLAN=yes
```

- `/etc/sysconfig/network-scripts/ifcfg-eth1`. This NIC is used for the span of traffic monitored by Snort.

```
DEVICE=eth1
ONBOOT=yes
BOOTPROTO=none
```

Trap receiver

PacketFence uses snmptrapd as the trap receiver. It stores the community name used by the switch to send traps in the switch config file (`/usr/local/pf/conf/switches.conf`):

```
[default]
SNMPCommunityTrap = public
```

Switch Setup

In our example, we enable linkUp/linkDown on a Cisco 2900LX and Port Security on a Cisco Catalyst 2960. Please consult the Network Devices Configuration Guide for the complete list of supported switches and configuration instructions.

linkUp/linkDown + MAC Notification

global setup

```
snmp-server enable traps snmp linkdown linkup
snmp-server enable traps mac-notification
snmp-server host 192.168.1.5 trap version 2c public snmp mac-notification
mac-address-table notification interval 0
mac-address-table notification
mac-address-table aging-time 3600
```

On each interface

```
switchport mode access
switchport access vlan 4
snmp trap mac-notification added
```

Port Security

global setup

```
snmp-server enable traps port-security
snmp-server enable traps port-security trap-rate 1
snmp-server host 192.168.1.5 version 2c public port-security
```

On each interface, you need to initialize the port security by authorizing a fake MAC address with the following commands

```
switchport access vlan 4
switchport port-security
switchport port-security maximum 2
switchport port-security maximum 1 vlan access
```

```
switchport port-security violation restrict
switchport port-security mac-address 0200.0000.00xx
```

where xx stands for the interface index

Don't forget to update the startup-config.

switches.conf

See Network Device Definition for more information about the content of this file.

Here are the parameters (other than the defaults ones) for our example

```
[default]
SNMPCommunityRead = public
SNMPCommunityWrite = private
SNMPommunityTrap = public
SNMPVersion = 1
vlans = 1,2,3,4,10
normalVlan = 1
registrationVlan = 2
isolationVlan = 3
macDetectionVlan = 4
VoIPEnabled = no

[192.168.1.100]
type = Cisco::Catalyst_2900XL
mode = production
uplink = 24

[192.168.1.101]
type = Cisco::Catalyst_2960
mode = production
uplink = 25
normalVlan = 10
radiusSecret=useStrongerSecret
```

If you want to have a different read/write communities name for each switch, declare it in each switch section.

pf.conf

Here is the `/usr/local/pf/conf/pf.conf` file for our setup. For more information about `pf.conf` see Global configuration file (pf.conf) section.

```
[general]
```

```
domain=yourdomain.org
dnsservers=192.168.2.1,192.168.3.1
dhcpservers=192.168.2.1,192.168.3.1

[network]
mode=vlan

[trapping]
registration=enabled
detection=enabled
testing=disabled
range=192.168.2.0/24,192.168.3.0/24

[registration]
auth=ldap

[interface eth0]
mask=255.255.255.0
type=internal,managed
gateway=192.168.1.1
ip=192.168.1.5

[interface eth0.2]
mask=255.255.255.0
type=internal
gateway=192.168.2.1
ip=192.168.2.1

[interface eth0.3]
mask=255.255.255.0
type=internal
gateway=192.168.3.1
ip=192.168.3.1

[interface eth1]
mask=255.255.255.0
type=monitor
gateway=192.168.1.5
ip=192.168.1.1
```

networks.conf

Here is the `/usr/local/pf/conf/networks.conf` file for our setup. For more information about `networks.conf` see DHCP and DNS Server configuration.

```
[192.168.2.0]
netmask=255.255.255.0
gateway=192.168.2.1
pf_gateway=192.168.2.254
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=registration
named=enabled
dhcpd=enabled

[192.168.3.0]
netmask=255.255.255.0
gateway=192.168.3.1
pf_gateway=192.168.3.254
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=isolation
named=enabled
dhcpd=enabled
```

DHCP and DNS Server Configuration

conf/templates/named-registration.ca

```
; This file is generated from a template at %%template%%
; Any changes made to this file will be lost on restart

$TTL 3600
. IN SOA %%hostname%%. %%incharge%% (
    2009020901  ; serial
    10800       ; refresh
    3600        ; retry
    604800      ; expire
    86400       ; default_ttl
)

       IN      NS      %%hostname%%.
*.     IN      A       192.168.2.1
```

```
       IN      MX      5       %%hostname%%.

1.2.168.192.in-addr.arpa.     IN      PTR       %%hostname%%
```

conf/templates/named-isolation.ca

```
; This file is generated from a template at %%template%%
; Any changes made to this file will be lost on restart

$TTL 3600
. IN SOA %%hostname%%. %%incharge%% (
    2009020901  ; serial
    10800       ; refresh
    3600        ; retry
    604800      ; expire
    86400       ; default_ttl
)

       IN      NS      %%hostname%%.
*.     IN      A       192.168.3.1
       IN      MX      5       %%hostname%%.

1.3.168.192.in-addr.arpa.     IN      PTR       %%hostname%%
```

FreeRADIUS

/etc/raddb/clients.conf

The client entry for the Cisco 2960.

```
client lab-cisco2960 {
     ipaddr       = 192.168.0.101
     secret       = useStrongerSecret
}
```

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, you need to install and configure the SNORT IDS to talk with PacketFence.

Snort

Installation

The installation procedure is quite simple for SNORT. We maintain a working version on the PacketFence repository. To install it, simply run the following command:

```
yum install snort
```

Configuration

PacketFence provides a basic snort.conf template that you may need to edit depending of the Snort version. The file is located in `/usr/local/pf/conf/templates`. It is rarely necessary to change anything in that file to make Snort work and trap alerts. DO NOT edit the snort.conf located in `/usr/local/pf/conf`, all the modification will be destroyed on each PacketFence restart.

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 policy violations are controlled using the `/usr/local/pf/conf/violations.conf` configuration file. The violation format is as follows:

```
[1234]
desc=Your Violation Description
priority=8
url=/content/index.php?template=<template>
redirect_url=/proxies/tools/stinger.exe
disable=N
trigger=Detect::2200032,Scan::11808
actions=email,log,trap
vlan=isolationVlan
whitelisted_categories=
```

- `[1234]`: violation ID. Any integer except 1200000-120099 which is reserved for required administration violations.
- `desc`: single line description of violation
- `priority`: range 1-10, with 1 the higest priority and 10 the lowest. Higher priority violations will be addressed first if a host has more than one.
- `url`: HTML URL the host will be redirected to while in violation. This is usually a local URL of the form `/content/index.php?template=...` where `...` is the name of the remediation template to show to the user. Full URLs like http://myportal.com/violation1234/ are also supported if `passthrough=proxy` is set under `[trapping]`. In that case, the Captive Portal will do reverse proxying to the specified URL. Great care should be taken when using this feature because any resource outside the specified path will fail to load.
- `redirect_url`: The user is redirected to this URL after he re-enabled his network access on the remediation page.
- `disable`: if disable is set to 'Y', this violation is disabled and no additional violations of this type will be added.
- `trigger`: method to reference external detection methods such as Detect (SNORT), Scan (Nessus), OS (DHCP Fingerprint Detection), USERAGENT (Browser signature), VENDORMAC (MAC address class), etc. Trigger is formatted as follows type::ID. in this example 2000032 is the snort id and 11808 is the Nessus plugin number. The Snort ID does NOT have to match the violation ID.
- `actions`: this is the list of actions that will be executed on a violation addition. The actions can be:
  - `log`: log a message to the file specified in `[alerting].log`

- email: email the address specified in [alerting].emailaddr, using [alerting].smtpserver. Multiple emailaddr can be sperated by comma.
- trap: isolate the host and place them in violation. It opens a violation and leaves it open. If trap is not there, a violation is opened and then automatically closed
- winpopup: send a windows popup message. You need to configure [alerting].winserver, [alerting].netbiosname in pf.conf when using this option
- external: execute an external command, specified in [paths].externalapi

❏ vlan: Destination VLAN where PacketFence should put the client when a violation of this type is open. The VLAN value can be:

- isolationVlan: Isolation VLAN as specified in switches.conf. This is the recommended value for most violation types.
- registrationVlan: Registration VLAN as specified in switches.conf.
- normalVlan: Normal VLAN as specified in switches.conf. Note: It is preferable not to trap than to trap and put in normal VLAN. Make sure you understand what you are doing.

❏ whitelisted_categories: Nodes in a category listed in whitelisted_categories won't be affected by a violation of this type. Format is a comma separated list of category names.

Also included in violation.conf is the defaults section. The defaults section will set a default value for every violation in the configuration. If a configuration value is not specified in the specific ID, the default will be used:

```
[defaults]
priority=4
max_enable=3
actions=email,log
auto_enable=Y
disable=Y
grace=120
button_text=Enable Network
snort_rules=local.rules,bleeding-attack_response.rules,bleeding-
exploit.rules,bleeding-p2p.rules,bleeding-scan.rules,bleeding-virus.rules
vlan=isolationVlan
whitelisted_categories=
```

❏ max_enable: 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.

❏ 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.

- `grace`: number of minutes 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.
- `button_text`: text displayed on the violation form to hosts.
- `snort_rules`: the Snort rules file is the administrators responsibility. Please change this to point to your violation rules file(s). If you do not specify a full path, the default is `/usr/local/pf/conf/snort`. If you need to include more than one file, just separate each filename with a comma.

`violations.conf` is loaded at startup.

Example violation

In our example we want to isolate people using Limewire. Here we assume Snort is installed and configured to send alerts to PacketFence. Now we need to configure PacketFence isolation.

Enable Limewire violation in `/usr/local/pf/conf/violations.conf` and configure it to execute an external script

```
[2001808]
desc=P2P (Limewire)
priority=8
url=/content/index.php?template=p2p
actions=log,trap
disable=N
max_enable=1
trigger=Detect::2001808
```

Conformity Scan (Nessus)

If you plan on using Nessus to scan client systems, you need to install the following packages:

- openssl-devel
- perl-IO-Socket-SSL
- perl-Parse-Nessus-NBE-1.1-1

Installation

Please visit http://www.nessus.org/download/ to download 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.

Configuration

In order for a given Nessus Scan to generate a violation inside PacketFence, you have to configure two sections:

- pf.conf
  Put the Nessus plugin ID in the scan.live_tids variable, and adjust the settings in the scan section like the following :

```
[scan]
ssl=enabled
pass=userPassword
user=nessusUsername
port=1241
host=127.0.0.1
registration=enabled
nessusclient_file=basic-policy.nessus
nessusclient_policy=basic-policy
live_tids=1100001,22964,10882
```

- violations.conf
  You need to create a new violation section and have to specify

```
trigger=Scan::<violationId>
```

Where `violationId` is the Id of the Nessus 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 Nessus plugin is higher than a low severity vulnerability

NessusClient Integration

New since 1.8.3 is the ability to directly use the nessus command line client and dot nessus files. The NessusClient File format is documented at http://www.nessus.org/documentation/dot_nessus_file_format.pdf and can easily be generated using the official Nessus Client.

You'll have to save your dot nessus file in the `/usr/local/pf/conf/nessus/` directory and specify its filename using the `scan.nessusclient_file` configuration setting. You'll also have to specify your policy name using the `scan.nessusclient_policy` setting. After that, you can execute your scan using

```
pfcmd schedule now <IP>
```

NOTE: If you provide credentials in the .nessus file, you need to enable the “Store passwords as plain text” option in your Nessus Client.

Scan on registration

To perform a system scan before giving access to a host on the network you need to enable the `scan.registration` parameter in `pf.conf`.

It is also recommended to adjust `scan.duration` to reflect how long the scan takes. A progress bar of this duration will be shown to the user while he is waiting. By default, we set this variable to 60s.

Adjust setuid/setgid on pfcmd

Since there are some limitations by using pfcmd in setuid mode to launch the Nessus scan, you will need to remove the setuid/setgid sticky bytes from the executable

```
chown ug-s /usr/local/pf/bin/pfcmd
```

Oinkmaster

Oinkmaster is a perl script that enables the possibility to update the different snort rules very easily. It is simple to use, and install. This section will show you how to implement Oinkmaster to work with PacketFence and Snort.

Please visit http://oinkmaster.sourceforge.net/download.shtml to download oinkmaster. A sample oinkmaster configuration file is provided at `/usr/local/pf/addons/snort/oinkmaster.conf`

Configuration

Here are the steps to make Oinkmaster work. We will assume that you already downloaded the newest oinkmaster archive :

- Untar the freshly downloaded Oinkmaster
- Copy the required perl scripts into /usr/local/pf/oinkmaster. You need to copy over contrib and oinkmaster.pl
- Copy the oinkmaster.conf provided by PacketFence (see the section above) in /usr/local/pf/conf
- Modify the configuration to suit your own needs. Currently, the configuration file is set to fetch the bleeding rules.

Rules update

In order to get periodic updates for PacketFence Snort rules, we simply need to create a crontab entry with the right information. The example below shows a crontab entry to fetch the updates daily at 23:00 PM :

```
0 23 * * * (cd /usr/local/pf; perl oinkmaster/oinkmaster.pl -C conf/oinkmaster.conf -o conf/snort/)
```

Floating Network Devices

Starting with version 1.9, PacketFence now supports floating network devices. A Floating network device is a device for which PacketFence has a different behaviour compared to a regular device. This functionality was originally added to support mobile Access Points.

Right now PacketFence only supports floating network devices on Cisco switches configured with port-security.

For a regular device, PacketFence puts 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 should 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 should reconfigure the port like before it was plugged.

Here is how it works:

- 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

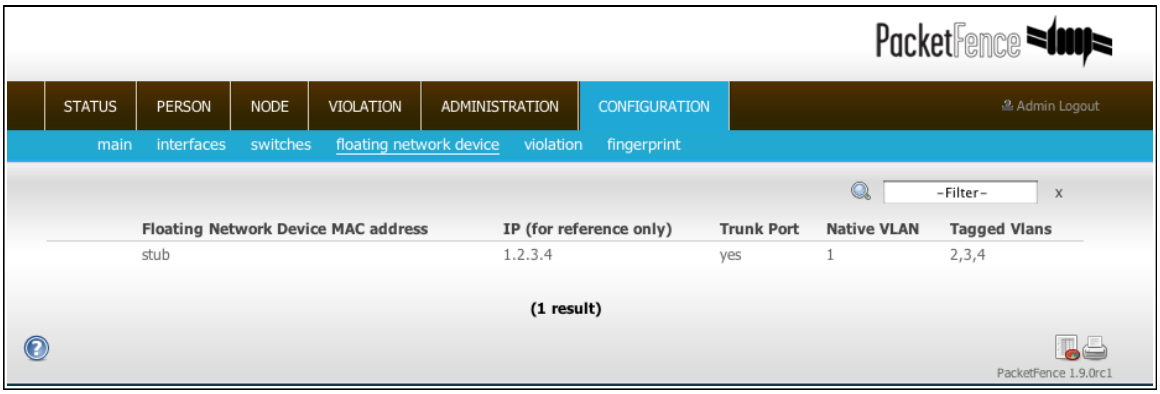
When PF receives a linkdown trap on a port in which a floating network device was plugged, it changes the port configuration so that:

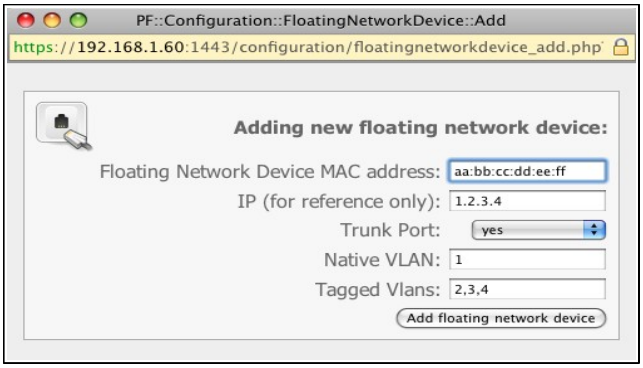
- it enables port-security
- it disables linkdown traps

Identification

As we mentioned ealier, each floating network device has to be identified. Ther are two ways to do it:

- by editing conf/floating_network_device.conf
- through the Web GUI, in the Configuration -> Floating Network Device tab.





Here are the settings that are available:

- MAC address
- IP address (in case of a static IP)
- trunkPort: yes/no. Should the port be configured as a muti-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 Vlans that have to be tagged on the port.

Operating System Best Practices

Iptables

You need to open the following ports in Iptables:

- 53 (DNS), 80 (HTTP), 443 (HTTPS) on eth0.2 and eth0.3
- 162 (SNMP) and 1443 (Web Admin Interface) on eth0

Here are the /etc/sysconfig/iptables lines for our setup:

```
-A RH-Firewall-1-INPUT -p udp -m udp --dport 53 -s 192.168.2.0/24 -d
192.168.2.1 -i eth0.2 -j ACCEPT
-A RH-Firewall-1-INPUT -p udp -m udp --dport 53 -s 192.168.3.0/24 -d
192.168.3.1 -i eth0.3 -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m tcp --dport 80 -s 192.168.2.0/24 -d
192.168.2.1 -i eth0.2 -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m tcp --dport 80 -s 192.168.3.0/24 -d
192.168.3.1 -i eth0.3 -j ACCEPT
-A RH-Firewall-1-INPUT -p udp -m udp --dport 162 -d 192.168.1.5 -i eth0
-j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m tcp --dport 443 -s 192.168.2.0/24 -d
192.168.2.1 -i eth0.2 -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m tcp --dport 443 -s 192.168.3.0/24 -d
192.168.3.1 -i eth0.3 -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m tcp --dport 1443 -d 192.168.1.5 -i eth0
-j ACCEPT
```

Log Rotations

PacketFence can generate a lot of log entries in huge production environments. This is why we recommend to use either logrotate or log4perl to periodically rotate your logs.

Logrotate

This is the easiest way to rotate your logs. In fact, a working logrotate script is provided with the PacketFence package. This script is located in /usr/local/pf/addons, and it's configured to do a weekly log rotation. Just add it to your existing logrotate cronjobs.

Log4perl

This log4perl way is a little more complex to achieve, but it is still quite simple. There are 3 packages you need to get from RPMForge :

- perl-Log-Dispatcher
- perl-Log-Dispatcher-FileRotate
- perl-Date-Manip

Once you downloaded those packages, you need to modify the logging configuration file (conf/log.conf) with something like the following example. Note that log4perl is almost the same as log4j, so you should be able to find a lot of documentation online.

```
log4perl.appender.LOGFILE=Log::Dispatch::FileRotate
log4perl.appender.LOGFILE.filename=/usr/local/pf/logs/packetfence.log
log4perl.appender.LOGFILE.mode=append
log4perl.appender.LOGFILE.autoflush=1
log4perl.appender.LOGFILE.size=51200000
log4perl.appender.LOGFILE.max=5
log4perl.appender.LOGFILE.layout=PatternLayout
log4perl.appender.LOGFILE.layout.ConversionPattern=%d{MMM dd HH:mm:ss}
%X{proc}(%X{tid}) %p: %m (%M)%n
```

High availability

A high availability setup (active/passive) for PacketFence can be created using two PacketFence servers and the following open source utilities:

- Linux-HA (www.linux-ha.org): a daemon that provides cluster infrastructure to its clients. Heartbeat would be responsible for starting the PacketFence services, eventually
- DRBD (www.drbd.org): A network based raid-1.

Since PacketFence stores most of its information in a MySQL database, the two PacketFence

redundant servers need to share this database in a way or another.

There are different options to share the database between the two PacketFence servers:

- A local MySQL database server on each PacketFence box configured to store its databases on a remote partition (a LUN on a SAN for example)
  - You have to make sure that only one database server is running at each time (don't double-mount the partition)
- A local MySQL database server on each PacketFence box and replication of the database partition using DRBD
- A remote MySQL database server with its own high availability setup

In this document, we describe the second option that involves DRBD.

We assume that:

- you are using RedHat Enterprise 5 or CentOS 5.
- pf1 is the first PacketFence server
- pf2 is the second PacketFence server
- PacketFence is properly configured on each server
- the DRBD partition is 30G long
- we use HeartBeat v1

Creation of the DRBD partition

During the OS installation, reduce the size of the main partition and create a new one (that will be used for the replicated MySQL database) of 30G. In order to do so, on VolGroup00:

- reduce the size of LogVol00 of 30G
- create a new partition (ext3) called mysql: 30G. You'll be asked to specify where this new partition will mounted: enter /data (or anything else that is used by Linux).
- after the first server reboot, edit /etc/fstab and delete the line for /data.

DRBD and Linux-HA Installation

Use the following line to install the required packages :

```
yum install drbd83 kmod-drbd83 heartbeat heartbeat-pils heartbeat-stonith
```

DRBD Configuration and setup

Initializing and configuring DRBD is not straight forward !

We strongly recommend that you read the online documentation available on DRBD website so you have a better idea of how it works...

Here we assume the name of the partition is mysql.

Load the DRBD kernel module:

```
modprobe drbd
```

Edit /etc/drbd.conf and put the following content:

```
global {
    usage-count yes;
}

common {
    protocol C;
}

resource mysql {
  syncer {
    rate 100M;
    al-extents 257;
  }

  startup {
    degr-wfc-timeout 120;    # 2 minutes.
  }

  disk {
    on-io-error   detach;
  }
  device        /dev/drbd0;
  disk          /dev/VolGroup00/mysql;
  meta-disk     internal;

  on pf1_server_name {
    address     x.x.x.x:7788;
  }

  on pf2_server_name {
    address     y.y.y.y:7788;
  }
}
```

where:

- *mysql* is the name of the partition you created when installing the OS

- *pf1_server_name* and *pf2_server_name* by the real server names.
- *x.x.x.x* and *y.y.y.y* by the IP addresses dedicated to DRBD on each server (use a dedicated NIC for this, not the main one with all the IPs)

Try to initialize DRBD by creating the metadata for the mysql partition with the following command:

```
drbdadm create-md mysql
```

You could get this kind of message:

```
md_offset 31474053120
al_offset 31474020352
bm_offset 31473057792

Found ext3 filesystem which uses 30736384 kB
current configuration leaves usable 30735408 kB

Device size would be truncated, which
would corrupt data and result in
'access beyond end of device' errors.
You need to either
   * use external meta data (recommended)
   * shrink that filesystem first
   * zero out the device (destroy the filesystem)
Operation refused.

Command 'drbdmeta 0 v08 /dev/VolGroup00/mysql internal create-md'
terminated with exit code 40
drbdadm create-md mysql: exited with code 40
```

If so, it means that you need to manually resize the partition like this:

```
root@pf1 ~]# resize2fs -p -f /dev/VolGroup00/mysql 30735408K
resize2fs 1.39 (29-May-2006)
Resizing the filesystem on /dev/VolGroup00/mysql to 7683852 (4k) blocks.
The filesystem on /dev/VolGroup00/mysql is now 7683852 blocks long.
```

Then initialize the partition:

```
[root@pf1 ~]# drbdadm create-md mysql
md_offset 31474053120
al_offset 31474020352
bm_offset 31473057792

Found ext3 filesystem which uses 30735408 kB
current configuration leaves usable 30735408 kB

Even though it looks like this would place the new meta data into
unused space, you still need to confirm, as this is only a guess.

Do you want to proceed?
[need to type 'yes' to confirm] yes

Writing meta data...
initializing activity log
NOT initialized bitmap
New drbd meta data block successfully created.
```

Start DRBD on both servers:

```
/etc/init.d/drbd start
```

Make sure you see something like this in /proc/drbd:

```
...

 0: cs:Connected ro:Secondary/Secondary ds:Inconsistent/Inconsistent C
r----
    ns:0 nr:0 dw:0 dr:0 al:0 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:b
oos:30702640
```

Synchronize the servers by forcing one to become the primary. So on pf1 do:

```
drbdadm -- --overwrite-data-of-peer primary mysql
```

After issuing this command, the initial full synchronization will start. You will be able to monitor its progress via /proc/drbd. It may take some time depending on the size of the device. Wait until complete.

Make sure DRBD is started at boot time:

```
chkconfig --level 2345 drbd on
```

Restart both servers.

When done, look in /proc/drbd and make sure you see:

```
...
 0: cs:Connected ro:Secondary/Secondary ds:UpToDate/UpToDate C r---
    ns:0 nr:0 dw:0 dr:0 al:0 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:b oos:0
```

MySQL Configuration

By default MySQL puts its data in /var/lib/mysql. In order to replicate data between the two servers, we mount the DRBD partition under /var/lib/mysql.

When first starting MySQL, the partition must be mounted.

In order to do so:

On the master server (the server you are working on), tell DRBD to become the primary node with:

```
drbdadm primary mysql
```

NOTE: mysql being the name of the DRBD partition.

In /proc/drbd you should see something like:

```
...
 0: cs:Connected ro:Primary/Secondary ds:UpToDate/UpToDate C r----
    ns:145068 nr:4448 dw:149516 dr:10490 al:31 bm:14 lo:0 pe:0 ua:0 ap:0
ep:1 wo:d oos:0
```

Mount the partition with:

```
mount /dev/drbd0 /var/lib/mysql
```

Start MySQL

```
service mysqld start
```

Execute the secure installation script in order to set the root password, remove the test databases and anonymous user created by default:

```
/usr/bin/mysql_secure_installation
```

Make sure MySQL does NOT start at boot time:

```
chkconfig --level 2345 mysqld off
```

Heartbeat configuration

Create /etc/ha.d/ha.cf with the following content:

```
bcast eth0
bcast eth1
keepalive 2
warntime 30
deadtime 60
auto_failback off
initdead 120
node pf1.example.org
node pf2.example.org
use_logd yes
```

Here we assume that the redundant connections for the Heartbeat between the 2 servers are on eth0 and eth1

Create /etc/ha.d/haresources with the following content:

```
pf1.example.com Ipaddr2::x.x.x.x IfUp::eth0.y IfUp::eth0.z
drbddisk::mysql Filesystem::/dev/drbd0::/var/lib/mysql::ext3 mysqld
packetfence
```

- *x.x.x.x* is PF admin virtual address
- *eth0.y* is the name of the NIC configuration file (/etc/sysconfig/network-scripts/ifcfg_eth0.y) dedicated to IP address in vlan y (registration for example).
- *eth0.z* is the name of the NIC configuration file (/etc/sysconfig/network-scripts/ifcfg_eth0.z) dedicated to IP address in vlan z (isolation for example).

Create the /etc/ha.d/resource.d/IfUp script that will mount IP addresses in Registration, Isolation, (eth0.y, etho0.z) with the following content:

```
case "$2" in
        start)
                echo -n "Mounting $1"
                /sbin/ifup $1
                echo "."
```

```
                ;;
        stop)
                echo -n "Unmounting $1"
                /sbin/ifdown $1
                echo "."
                ;;
        *)
                echo "Usage: $0 {start|stop}"
                exit 1
                ;;
esac
```

and make it executable:

```
chmod 755 /etc/ha.d/resource.d/IfUp
```

Create /etc/ha.d/authkeys with the following content:

```
auth 1
1 sha1 10b245aa92161294df5126abc5b3b71d
```

and change its rights like this

```
chmod 600 /etc/ha.d/authkeys
```

Create /etc/logd.cf with the following content:

```
debugfile /var/log/ha-debug
logfile /var/log/ha-log
logfacility daemon
```

NOTE: Make sure port 694 is opened (through iptables) on both servers

Start Heartbeat:

```
service heartbeat start
```

Look at Heartbeat log file /var/log/ha-log to make sure that everything is fine.

Enable HB automatic start

```
chkconfig --level 345 heartbeat on
```

RADIUS HA configuration

If you configured FreeRADIUS with your wireless setup and you configured redundancy, you could configure FreeRADIUS to answer requests exclusively coming on the virtual IP. In order to do so, you need to modify the RADIUS configuration and add RADIUS to the managed resources.

RADIUS Configuration

Modify the listen statements in the radiusd.conf file per the following. Change the [VIP_IPV4_ADDRSS] with your virtual IP address :

```
listen {
     type = auth
     ipaddr = [VIP_IPV4_ADDRESS]
     port = 0
}
listen {
     ipaddr = [VIP_IPV4_ADDRESS]
     port = 0
     type = acct
}
```

Heartbeat Configuration

Add RADIUS to the managed resources :

```
pf1.example.com Ipaddr2::x.x.x.x IfUp::eth0.y IfUp::eth0.z
drbddisk::mysql Filesystem::/dev/drbd0::/var/lib/mysql::ext3 mysqld
packetfence radiusd
```

Performance optimization

MySQL optimizations

Tuning MySQL itself

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_awe_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.

Shutdown PacketFence and MySQL

```
# /etc/init.d/packetfence stop
Shutting down PacketFence...
[...]
# /etc/init.d/mysql stop
Stopping MySQL:                                            [  OK  ]
```

Edit /etc/my.cnf (or your local my.cnf)

```
[mysqld]
# Set buffer pool size to 50-80% of your computer's memory
innodb_buffer_pool_size=800M
innodb_additional_mem_pool_size=20M
innodb_flush_log_at_trx_commit=2
# allow more connections
max_connections=700
# set cache size
key_buffer_size=900M
table_cache=300
query_cache_size=256M
# enable log
log=ON
log_slow_queries = ON
log-slow-queries=/var/log/mysql/slow-queries.log
long_query_time = 10
log-long-format
```

Start up MySQL and PacketFence

```
# /etc/init.d/mysqld start
```

```
Starting MySQL:                                            [  OK  ]
# /etc/init.d/packetfence start
Starting PacketFence...
[...]
```

Wait 10 minutes for PacketFence to initial the network map and 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 tool on your database setup after a couple of weeks to help you identify MySQL configuration improvement.

http://blog.mysqltuner.com/download/

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_history` after some time. A closed record is one where the `end_time` field is set to a date (strickly 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 tend 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 700. 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 'mysqladmin 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/blocked-host.html for details.

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, ...).

So far, we blacklisted clients known to be misbehaving. However, a completely different approach can be taken: whitelist only known browsers.

This has the nasty side-effect of being unfriendly with (blocking) less popular browsers and devices so this is disabled by default.

If you want to enable this feature, edit `/usr/local/pf/conf/templates/httpd.conf` and uncomment the following lines in each virtual host section:

```
RewriteCond %{HTTP_USER_AGENT} !^Mozilla
RewriteCond %{HTTP_USER_AGENT} !^Opera
RewriteCond %{HTTP_USER_AGENT} !^BlackBerry
RewriteRule ^.*$ - [L,forbidden]
```

This will allow the following browsers to reach the captive portal:

- BlackBerry
- Firefox
- Google Chrome
- Internet Explorer
- Opera
- Safari

Frequently Asked Questions

Services

How to restart PacketFence ?

```
service packetfence restart
```

How can I check that PacketFence is active ?

```
service packetfence status
```

How can I update PacketFence, I cannot see the updates for RPMForge or PacketFence packages?

By default, we disable those repositories to avoid potential problems. If you are ready to update your PacketFence, use the –enablerepo flag with yum :

```
yum update --enablerepo=PacketFence,rpmforge
```

Web interface

Where is the admin interface ?

The Web Administration GUI is available using https. Its default port is 1443.

How to change the admin password ?

You need to encrypt the new password in the admin.conf file with htpasswd:

```
htpasswd /usr/local/pf/conf/admin.conf admin
```

Then enter the new password twice.

How to add a new person ?

Go in the Web Admin GUI to Person -> Add

Fill the 2 following fields:

- Identifier
- Notes

Click on the Add button to save it

How can I provide different rights to users of the admin Web interface?

Starting with PacketFence 1.9.1 this is now possible. Have a look at the `conf/admin.perm` file. A copy is provided below for reference:

```
; PacketFence Granular Access Control
;
; Users to roles definitions
; Syntax:
;       <username>=<role>
;
; username MUST be an exact match to username in conf/admin.conf

[users]
; if a user is not in this file he will be assigned default_role
default_role=level1
admin=admin
obilodeau=level2

; Roles to permission-level definitions
; Syntax:
;       <role>={1|2|3|4|5}
[roles]
admin=5
level1=2
level2=3

; Permission-level required for page
; Syntax:
;       <pagename>={1|2|3|4|5}
```

```
[permissions]
status=1
person=2
node=2
violation=3
configuration=4
administration=4
scan=3
```

Basically, the permission control is using a users / role / permission level. You define the permission required to access a certain page (default values provided), you define the permission level of a role and you assign a role to users.

So, in the above example you will notice that obilodeau has a level2 access which allows him to access violation pages but not view or edit configuration or administration information. Also, by default users will be given a level1 access (see `default_role` entry under `[users]`).

Note: Users need to logout and re-login in order for permission changes to take effect.

How can users authenticate using LDAP or Active Directory (AD) instead of using a local flat file?

Have a look into the `conf/admin_ldap.conf` file. By providing the right values your users should be able to log into the Web administration interface using their LDAP or AD credentials. The two followings settings are optional $ldap_group_member_key and $ldap_group_dn (use them in case of group membership)

Advanced features

I need more flexible VLAN assignment

PacketFence is built with flexibility in mind allowing users to modify the configurable behavior in every way possible. To ease this process, extension points were defined such that any function present in `pf::vlan::custom` will take precedence over the default behavior in `pf::vlan`.

If you have many VLANs depending on node attributes (i.e.: categories), SSIDs or MAC attributes, you need to put some effort into redefining the `get_normal_vlan()` function.

Have a look at `/usr/local/pf/lib/pf/vlan/custom.pm` which already has many examples commented out. Here are a few of them. You only have to remove the comments to have them take effect.

Return `customVlan5` (as defined in `switches.conf`) to the nodes in the admin category:

```
sub get_normal_vlan {
    my ($this, $switch, $ifIndex, $mac, $node_info, $connection_type,
$user_name, $ssid) = @_;
    my $logger = Log::Log4perl->get_logger();

    # custom example: admin category
    # return customVlan5 to nodes in the admin category
    if (defined($node_info->{'category'}) && lc($node_info->{'category'})
eq "admin") {
        return $switch->getVlanByName('customVlan5');
    }
    return $switch->getVlanByName('normalVlan');
}
```

Return `guestVlan` to anyone who logged in as user guest:

```
sub get_normal_vlan {
    my ($this, $switch, $ifIndex, $mac, $node_info, $connection_type,
$user_name, $ssid) = @_;
    my $logger = Log::Log4perl->get_logger();

    # custom example: simple guest user
    # return guestVlan for pid=guest
    if (defined($node_info->{pid}) && $node_info->{pid} =~ /^guest$/i) {
        return $switch->getVlanByName('guestVlan');
    }
    return $switch->getVlanByName('normalVlan');
}
```

If a node has a Bypass VLAN set, use this value instead of `normalVlan`:

```
sub get_normal_vlan {
    my ($this, $switch, $ifIndex, $mac, $node_info, $connection_type,
$user_name, $ssid) = @_;
    my $logger = Log::Log4perl->get_logger();

    # custom example: enforce a node's bypass VLAN
    # If node record has a bypass_vlan prefer it over normalVlan
    if (defined($node_info->{'bypass_vlan'}) && $node_info-
>{'bypass_vlan'} ne '') {
        return $node_info->{'bypass_vlan'};
    }
    return $switch->getVlanByName('normalVlan');
}
```

Return the `customVlan1` for users connected to the PacketFenceRocks SSID:

```
sub get_normal_vlan {
    my ($this, $switch, $ifIndex, $mac, $node_info, $connection_type,
$user_name, $ssid) = @_;
    my $logger = Log::Log4perl->get_logger();

    # custom example: VLAN by SSID
    # return customVlan1 if SSID is 'PacketFenceRocks'
    if (defined($ssid) && $ssid eq 'PacketFenceRocks') {
        return $switch->getVlanByName('customVlan1');
    }
    return $switch->getVlanByName('normalVlan');
}
```

Instead of using the flexible VLAN name where actual VLAN numbers are looked up in `switches.conf` you can specify VLAN tags directly if you want to. For example a `return 5;` would put the user in VLAN 5. We favor using VLAN names because you don't need to change the code to change a VLAN.

How do I let users trapped in registration or isolation reach certain Websites (Passthrough)?

What you are trying to accomplish is called passthrough. Passthrough configuration is defined in `/usr/local/pf/conf/pf.conf` under `[passthroughs]`.

The format is `name=URL` where name is only for reference and URL must be a valid URL of the form: `protocol://domain/` or `protocol://domain/resource`. Supported protocols are http and https.

Also, under `[trapping]` you need `passthrough=proxy` for the feature to be enabled.

PacketFence needs to be restarted for the changes to take effect.

For now it is not editable through the Web Administration but there is an opened issue to implement that feature. See ticket #1103.

Switches

How do I add a new switch ?

You have to define the new switch in /usr/local/pf/conf/switches.conf

```
[10.0.0.xxx]
ip = 10.0.0.xxx
type = Cisco::Catalyst_2960
mode = production
uplink = 10101,10102
```

- ip: switch IP address
- mode
    - testing: pfsetvlan writes in the log files what it would normally do, but it doesn't react on the traps, it does not change any VLAN.
    - registration: pfsetvlan automatically register all MAC addresses seen on the switch ports. As in testing mode, no VLAN changes are done.
    - production: this is the regular working mode; pfsetvlan sends the SNMP writes to change the VLAN on the switch ports.

CAUTION: **If [trapping.testing] (see pf.conf) is enabled, PacketFence will consider that all the switches are in test whatever the working mode is !**

- uplink: list of ports that are NOT managed by PacketFence. All uplink ports must be defined here!
    - Dynamic (default): PF queries the switch using CDP (Cisco only !!) to get the uplinks
    - 1,2,10001,10002,... : uplink list if a hub is plugged in a port, this port must be declared as uplink !!!

- type: switch type. As defined in Supported Switches, Access points and Controllers. Could be for example:
    - Cisco::Catalyst_2960
    - Extreme::Summit_X250e

I want to manage only some ports on a switch but not all the ports. How can I do that ?

You need to use the uplink parameter. All the ports declared as uplinks will be ignored and not managed by PacketFence. This parameter is defined in the `[default]` section of `/usr/local/pf/conf/switches.conf`. You can define a different uplink list for each switch. You have to use the ifIndex of a port (not its port number) !

Troubleshooting

PacketFence does not start

Look at the error message either on the command line or in the logs file to see what is the cause of the problem.

Where can I see PacketFence logs and errors ?

In `/usr/local/pf/logs`:

- `admin_error_log`
- `error_log`
- `packetfence.log`

When I plug a computer in a switch, it does not get a network access

- Make sure PF gets traps from that switch
- Make sure PF reacts when it receives traps from that switch: look into `/usr/local/pf/logs/packetfence.log`

How to make sure PacketFence gets traps from a switch ?

- Make sure network connectivity is fine between the switch and PF
- Make sure port 162 (UDP) is open on the PF server
- Make sure you define PF as a trap receiver in the switch
- Make sure the switch uses the right snmp settings (community, snmp version,...) to send traps to PF. Checkout the SNMP settings in the switch

You could use `tcpdump` to see if PF receives traps: ''`tcpdump -i ethx port 162`''

Where should I look to see what PacketFence does when it gets a trap ?

Look into `/usr/local/pf/logs/packetfence.log`: PacketFence logs there everything it does for every trap.

PacketFence receives traps but does not do anything

There could be many reasons mentioned in `/usr/local/pf/logs/packetfence.log`:

- Is the switch is in production mode ?

- Is the port defined as a uplink

- Is the actual Vlan of the port managed by PF ?

- Is the new Vlan (i.e. the Vlan in which PF should put the port for the new device) managed by PF ?

- Is the new Vlan defined in the switch ?

How do I configure a client to do Wired 802.1X?

To be contributed...

How do I configure a client to do Wireless 802.1X?

To be contributed...

Technical introduction to VLAN mode

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 it's 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

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.

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.

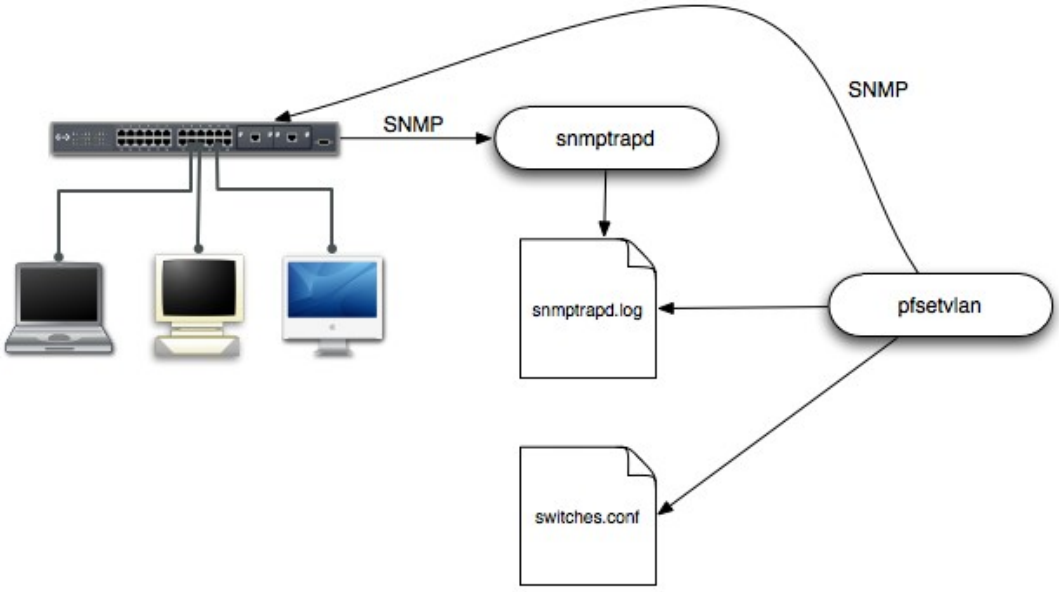
Wireless: 802.1X + MAC authentication

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).

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 writes them into a flat file: `/usr/local/pf/logs/snmptrapd.log`. The multithreaded pfsetvlan daemon reads these traps from the flat file and responds 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::SNMP class). Depending on your switches capabilities, pfsetvlan 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.

linkUp/linkDown traps

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 an empty 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 pfsetvlan will send periodical SNMP queries to the switch until the switch learns the MAC of the device. When the MAC address is known, pfsetvlan 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 pfsetvlan. 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

If your switches support MAC notification traps (MAC learnt, MAC removed), we suggest that you activate them in addition to the linkUp/linkDown traps. This way, pfsetvlan 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 thread. When the switch learns the MAC address of the device it sends a MAC learnt 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.

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:

```
# /usr/local/pf/bin/pfcmd
Usage: pfcmd <command> [options]

class                    | view violation classes
config                   | query, set, or get help on pf.conf
configuration paramaters
configfiles              | push or pull configfiles into/from database
fingerprint              | view DHCP Fingerprints
graph                    | trending graphs
history                  | IP/MAC history
ifoctetshistorymac       | accounting history
ifoctetshistoryswitch    | accounting history
ifoctetshistoryuser      | accounting history
interfaceconfig          | query/modify interface configuration parameters
ipmachistory             | IP/MAC history
locationhistorymac       | Switch/Port history
locationhistoryswitch    | Switch/Port history
lookup                   | node or pid lookup against local data store
manage                   | manage node entries
networkconfig            | query/modify network configuration parameters
node                     | node manipulation
nodecategory             | nodecategory manipulation
person                   | person manipulation
reload                   | rebuild fingerprint or violations tables
without restart
report                   | current usage reports
schedule                 | Nessus scan scheduling
```

```
service                 | start/stop/restart and get PF daemon status
switchconfig            | query/modify switches.conf configuration
parameters
switchlocation          | view switchport description and location
traplog                 | update traplog RRD files and graphs or obtain
switch IPs
trigger                 | view and throw triggers
ui                      | used by web UI to create menu hierarchies and
dashboard
update                  | download canonical fingerprint or OUI data
version                 | get installed PF version and database MD5s
violation               | violation manipulation
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|||||
```

pfcmd_vlan

pfcmd_vlan is the command line interface to most VLAN isolation related functionalitites.

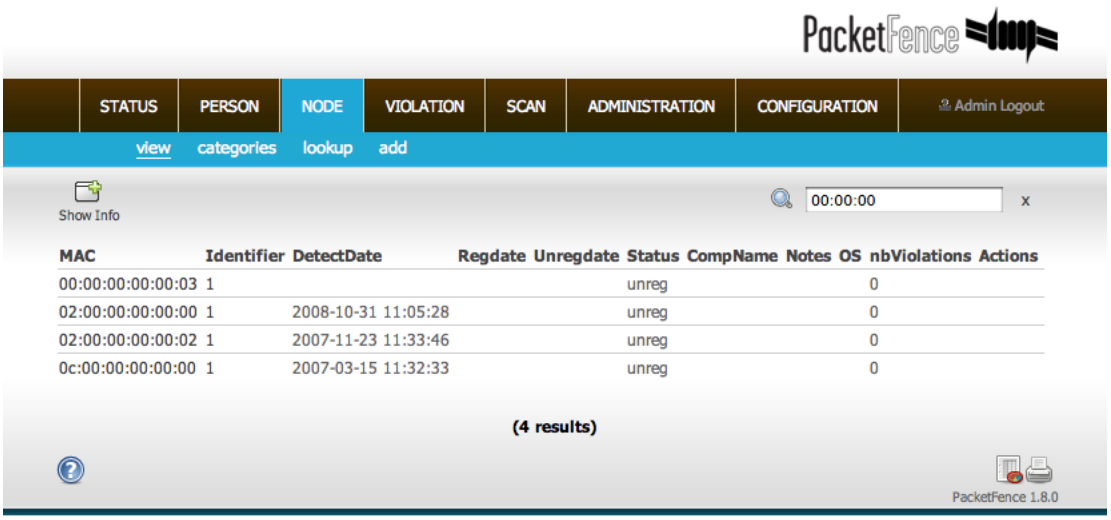
Again, when executed without any arguments, a help screen is shown.

```
# /usr/local/pf/bin/pfcmd_vlan
Usage:
    pfcmd_vlan command [options]

Command:
  -deauthenticate      de-authenticate a dot11 client
  -getAlias            show the description of the specified switch port
  -getAllMACs          show all MACS on all switch ports
  -getHubs             show switch ports with several MACs
  -getIfOperStatus     show the operational status of the specified
```

```
switch port
  -getIfType           show the ifType on the specified switch port
  -getLocation         show at which switch port the MAC is found
  -getMAC              show all MACs on the specified switch port
  -getType             show switch type
  -getUpLinks          show the upLinks of the specified switch
  -getVersion          show switch OS version
  -getVlan             show the VLAN on the specified switch port
  -getVlanType         show the type of the specified port
  -help                brief help message
  -isolate             set the switch port to the isolation VLAN
  -man                 full documentation
  -reAssignVlan        re-assign a switch port VLAN
  -resetVlanAllPort    reset VLAN on all non-UpLink ports of the
specified switch
  -resetVlanNetwork    reset VLAN on all non-UpLink ports of all managed
switches
  -setAlias            set the description of the specified switch port
  -setDefaultVlan      set the switch port to the default VLAN
  -setIfAdminStatus    set the admin status of the specified switch port
  -setVlan             set VLAN on the specified switch port
  -setVlanAllPort      set VLAN on all non-UpLink ports of the specified
switch

Options:
  -alias               switch port description
  -ifAdminStatus       ifAdminStatus
  -ifIndex             switch port ifIndex
  -mac                 MAC address
  -showMACVendor       show the MAC vendor
  -showPF              show additional information available in PF
  -switch              switch description
  -verbose             log verbosity level
                         0 : fatal messages
                         1 : warn messages
                         2 : info messages
                       > 2 : full debug
  -vlan                VLAN
```

Web Admin GUI

The Web Admin GUI, accessible using https on port 1443, shows the same information available using pfcmd.



Appendix B: Legacy FreeRADIUS 1.x configuration

Since PacketFence 1.9.0 we recommend the use of FreeRADIUS 2.x over 1.x.

This documentation is provided here for historical reference.

FreeRADIUS 1.x Configuration

Make sure to install the following packages:

- freeradius

/etc/raddb/clients.conf

Add the following lines:

```
client 192.168.0.3 {
  secret = secretKey
  shortname = AP1242
}
```

/etc/raddb/radiusd.conf

Add the following lines to the modules{} section:

```
perl {
     module = ${confdir}/rlm_perl_packetfence.pl
}
```

Make sure the authorize{} section looks like this:

```
authorize {
     preprocess
     eap
```

```
     files
     perl
}
```

Make sure the post-auth{} section looks like this:

```
post-auth {
     perl
}
```

Make sure the mschap{} section looks like this:

```
mschap {
     authtype = MS-CHAP
     use_mppe = yes
     require_encryption = yes
     require_strong = yes
     with_ntdomain_hack = yes
     ntlm_auth = "/usr/bin/ntlm_auth --request-nt-key --username=%
{mschap:User-Name:-None} --challenge=%{mschap:Challenge:-00} --nt-
response=%{mschap:NT-Response:-00}"
}
```

/etc/raddb/eap.conf

Make sure this file looks like:

```
eap {
     default_eap_type = peap
     timer_expire     = 60
     ignore_unknown_eap_types = no
     cisco_accounting_username_bug = no

     md5 {
     }

     leap {
     }

     gtc {
             auth_type = PAP
```

```
    }

    tls {
           private_key_file = /usr/local/pf/conf/ssl/keyfile.key
           certificate_file = /usr/local/pf/conf/ssl/certfile.crt
           CA_file = /usr/local/pf/conf/ssl/CAfile.crt
           dh_file = /dev/null
           random_file = /dev/urandom
    }

    peap {
           default_eap_type = mschapv2
    }

    mschapv2 {
    }
}
```

/etc/raddb/users

Add the following lines where we define that non EAP-messages should, by default, lead to an authentication acceptance

```
DEFAULT EAP-Message !* "", Auth-Type := Accept
```

/etc/raddb/rlm_perl_packetfence.pl

This perl script uses the Calling-Station-Id RADIUS request attribute, containing the MAC of the wireless station, to determine its registration and violation status. Based on this information, it sets the Tunnel-Medium-Type, Tunnel-Type and Tunnel-Private-Group-ID RADIUS reply attributes. The AP, upon reception of these three attributes, then confines the wireless station into the specified VLAN.

Make sure to set the required configuration parameters on top of the file. Mainly, the VLAN tags used in your environment and PacketFence's database credentials.

```
# Database connection settings
DB_HOSTNAME => 'localhost',
DB_NAME     => 'pf',
DB_USER     => 'pf',
DB_PASS     => 'pf',
# VLAN configuration
VLAN_GUEST        => 5,
VLAN_REGISTRATION => 2,
```

```
VLAN_ISOLATION    => 3,
VLAN_NORMAL       => 1
```

Tests

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

```
# radtest dd9999 Abcd1234 localhost 12 testing123

Sending Access-Request of id 74 to 127.0.0.1 port 1812
     User-Name = "dd9999"
     User-Password = "Abcd1234"
     NAS-IP-Address = 255.255.255.255
     NAS-Port = 12
rad_recv: Access-Accept packet from host 127.0.0.1:1812, id=74, length=20
```

Debug

In order to start FreeRadius in debug mode, start it using the following command:

```
# radiusd -X
```

Additional Information

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

packetfence-announce@lists.sourceforge.net: Public announcements (new releases, security warnings etc.) regarding PacketFence

packetfence-devel@lists.sourceforge.net: Discussion of PacketFence development

packetfence-users@lists.sourceforge.net: User and usage discussions

Commercial Support and Contact Information

For any questions or comments, do not hesitate to contact us by writing an email to :

support@inverse.ca

Inverse (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/support.html for details.

Chapter 16

GNU Free Documentation License

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