PacketFence – version 1.7.5

Installation and Configuration Guide

Version 1.7.5 – October 2008

Contents

Chapter 1

About this Guide

This guide will walk you through the installation and configuration of the PacketFence solution. It covers VLAN isolation setup.

The instructions are based on version 1.7.5 of PacketFence.

The latest version of this guide is available at
http://inverse.ca/uploads/docs/PacketFence_Installation_Guide.pdf.

Chapter 2

Introduction

PacketFence is an open-source network access control (NAC).

PacketFence features:

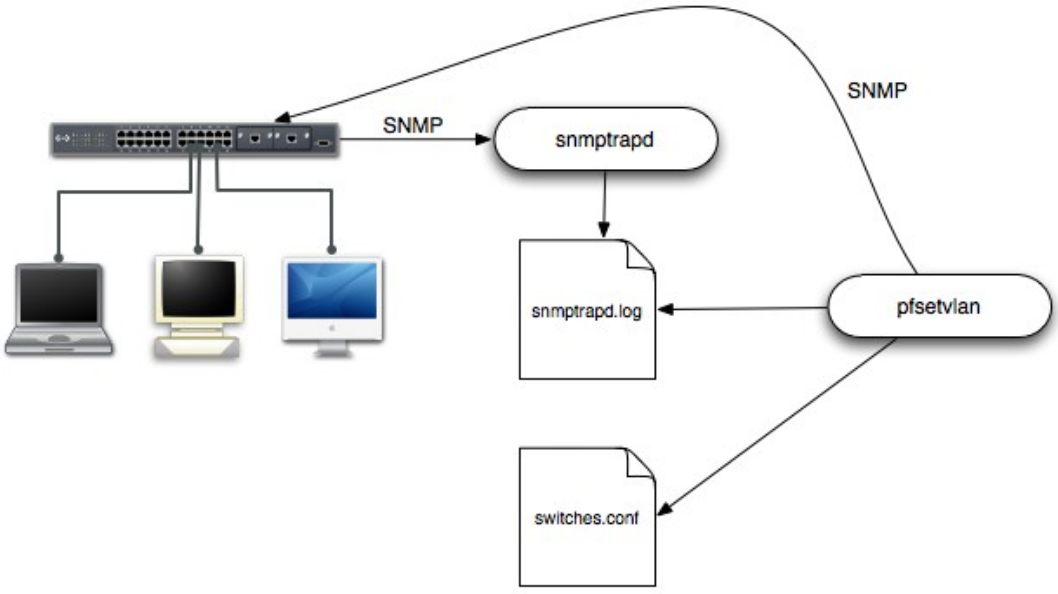
- Registration of new network devices through a captive portal. Configurable options exist for registration “windows” – absolute or relative time periods during which users may skip registration with periodic reminders. An Acceptable Use Policy can be specified such that users cannot enable network access without first accepting it. The duration of a node registration can be a relative value (eg. “four weeks from first network access”) or an absolute date (eg. “Thu Jan 20 20:00:00 EST 2009”).
- Detection of abnormal network activities (computer virus, worms, spyware, etc.) including from 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: email, log, trap.
- Proactive vulnerability scans: Nessus vulnerability scans can be performed on a scheduled or ad-hoc basis. A host can be scanned at registration, allowing administrators to isolate infected machines. 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 and user-directed mitigation/remediation: Once trapped, all HTTP, IMAP and POP sessions are terminated by the PacketFence system. Based on the nodes 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 removal instructions for the particular infection he/she has.A maximum number of re-enables, also configurable per violation, exists to permanently disable access if the user cannot solve the problem on his/her own.
- Web-based and command-line interfaces for management tasks
- Three different trapping mechanisms: ARP, DHCP and VLAN.
  - ARP allows to much more control over policy violations, but requires that PacketFence has a local interface to all networks (must sit in front of the router).
  - DHCP allows to have one PacketFence system in a remote location controlling many networks (Router will Relay DHCP requests). The down side is that you must replace your existing DHCP server with PacketFence, that static IPs can bypass isolation, and that the DHCP lease time will need to expire (50-100% of lease time) before a host can be put into isolation.
  - VLAN isolation allows you to totally isolate devices from the network by putting them in separate (non routed) VLANs. You need to create two VLANs: registration and isolation and then configure your switches to send SNMP traps to PacketFence. VLAN isolation is available in 1.7 and later.

- DHCP fingerprinting which can be used to automatically register VoIP phones, game consoles and more
- VoIP support (even in heterogeneous environments) for multiple switch vendors (Cisco, Edge-Core, HP, LinkSys, Nortel Networks and many more)
- 802.1X support through a FreeRADIUS module
- Wireless integration with FreeRADIUS which allows you to secure your wired and wireless networks (different APs and Cisco WLC) the same way

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

How does VLAN isolation work ?

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 pot the port in the MAC detection VLAN and can than 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.

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.

PacketFence policy violations are controlled using the `/usr/local/pf/conf/violation.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
```

- `[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.
- `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) 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`

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
```

- `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.

  `Violation.conf` is loaded at startup.

New features in 1.7.5

See `/usr/local/pf/CHANGES` file for a complete list.

- Add support for stacked Nortel BayStack 5520 (contribution from Matt Ashfield)
- Add support for Enterasys SecureStack C2
- Add support for Cisco Controller 4400
- Add support for 3COM SS4500

Bugs fixed in 1.7.5

See `/usr/local/pf/CHANGES` file for a complete list.

- #450: aup link in register.html leads to page change of the registration page (missing return false after window.open)
- #454: fixed OUI and fingerprint updates in installer.pl: installer.pl cannot execute pfcmd calls since pf.conf does not exist at that point in time !
- #466: Unknown modifier 'r' in /usr/local/pf/html/admin/common.php on line 340
- #487: hostname and domainname are not always correctly set in configurator.pl
- #493: It should be forbidden to delete a person when it still has registered nodes in its name
- #495: calling pfcmd violation delete generates “Use of uninitialized value in concatenation (.) or string at ./pfcmd line 1372”

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)
- 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 and configuration 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
ISC DHCP	DHCP 3
ISC BIND	BIND 9
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) ■ 3 Network cards

Operating System Requirements

Currently PacketFence 1.7.5 supports the following 32-bit operating systems:

- Red Hat Enterprise Linux 5.x Server
- Community ENTerprise Operating System (CentOS) 5.x

Make sure the required components are started automatically (except Snort that is controlled by PacketFence) at boot time and that they are running before proceeding with the PacketFence configuration. Also 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 and Fedora are known to work but this document won't cover them.

Installation

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

OS Installation

Install CentOS 5 or RedHat Enterprise Linux 5 with minimal installation and no additional packages. Then:

- Enable Firewall
- Disable SELinux

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-0.3.6-1.el5.rf.i386.rpm
```

Before you continue with the installation we recommended that you go through the section "1.1 Priorities" (http://wiki.centos.org/AdditionalResources/Repositories/RPMForge) in order to protect your base repository.

Update your database repository and your system:

```
yum update
```

Software Downloads

Download PacketFence package for CentOS5 from the PacketFence web site (http://www.packetfence.org/download/releases.html).

Software Installation

We recommend you to install PacketFence with Yum since Yum will satisfy all possible dependencies for you:

```
yum –nogpgcheck install packetfence-1.7.5-1.el5.noarch.rpm
```

If you install PacketFence without Yum, you have to install the following dependencies before:

- chkconfig, coreutils, glibc-common, grep, httpd, iproute, libpcap, libxml2, mod_ssl, mysql, net-snmp, openssl, php, php-gd, sed, tar, wget, zlib, zlib-devel
- perl (>= 5.8.0), perl-Apache-Htpasswd, perl-Config-IniFiles, perl-CGI, perl-CGI-Session, perl-Date-Parse,.perl-DBD-MySQL, perl-File-Spec, perl-File-Tail, perl-Locale-gettext, perl-LWP-UserAgent, perl-Net-Appliance-Session, perl-Log-Log4perl (>= 1.11), perl-Net-MAC, perl-Net-MAC-Vendor, perl-Net-Netmask, perl-Net-Pcap (>= 0.16), perl-Net-RawIP (0.2), perl-Net-SNMP, perl-Net-Telnet, perl-Parse-RecDescent, perl-RRDs, perl-suidperl, perl-Template, perl-Term-ReadKey, perl-Thread-Pool, perl-Time-HiRes,

Add perl-Net-RawIP in the list of packages to exclude from your package manager updates. For Yum, edit `/etc/yum.conf` and add the following line:

```
exclude=perl-Net-RawIP
```

Update line 756 of `/usr/lib/perl5/vendor_perl/5.8.8/Net/Telnet/Cisco.pm`:

```
return wantarray ? split /$/m, $_ : $_; # ORS instead?
```

Install the IPTables::IPv4 perl module using MCPAN:

```
perl -MCPAN -e 'install IPTables::IPv4
```

and update line 5 of `/usr/lib/perl5/site_perl/5.8.8/i386-linux-thread-multi/IPTables/IPv4.pm`:

```
my %IPv4;
```

Set the timezone in `/etc/php.ini`. For example:

```
date.timezone="America/Montreal
```

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 isolation. 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.

General Configuration

Execute the configurator at `/usr/local/pf/configurator.pl` to configure PacketFence according your needs.

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.

Remember that SELinux must be disabled.

Authentication (flat file, LDAP, Radius)

PacketFence can authenticate users that register devices using a flat file, an LDAP 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/ldap.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
```

Enter the password twice

LDAP

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

```
my $LDAPUserBase = "ou=People,dc=domain,dc=edu";

my $LDAPUserKey = "uid";

my $LDAPUserScope = "one";

my $LDAPBindDN = "cn=ldapuser,dc=domain,dc=edu";

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. The list entries will be pulled from the AuthName directives of each method.

VLAN isolation

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)
- VLAN 4 is the MAC detection VLAN (empty VLAN)
- VLANs 2 and 3 are spanned throughout the network
- 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
- We use Snort as NIDS. Refer to Snort web site for installation and configuration instructions
- Since Snort sees only the IP address of the devices and PacketFence's database is indexed by MAC, we span the DHCP traffic to PacketFence so it always knows the IP-MAC association. We use eth1 on PacketFence for the DHCP span (Refer to your switch configuration for SPAN setup)
- The traffic monitored by Snort is spanned on eth2
- 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	192.168.2.1	192.168.2.1
4	Mac Detection			
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=no
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=no
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 DHCP traffic.

```
DEVICE=eth1
ONBOOT=no
BOOTPROTO=none
```

- `/etc/sysconfig/network-scripts/ifcfg-eth2`. This NIC is used for the span of traffic monitored by Snort.

```
DEVICE=eth2
ONBOOT=no
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`) in th `[default]` section:

```
[default]
communityTrap = public
```

Switch Setup

In our example, we enable linkUp/linkDown + MAC Notification on 2900XL and Port Security on 2960.

- 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
```

  There are no parameters needed on each interface for linkUp/linkDown traps since these traps are enabled globally for all the ports.

- 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

See Appendix B for the complete list of supported switches and Appendix C for configuration instructions.

Logs

The log config file is `/usr/local/pf/conf/log.conf`. It contains the configuration for Log::Log4Perl and you normally don't need to modify it.

Custom Trap Handling Functions

Pfsetvlan is the daemon responsible of trap handling. When it receives a trap, pfsetvlan uses some functions defined in `/usr/local/pf/conf/pfsetvlan.pm` in order to know what to do.

For example, `custom_getCorrectVlan()` allows you to define what you consider to be the correct VLAN for a given switch port and connected MAC. In our example there is only one VLAN (VLAN 1) so the function should look like

```
sub custom_getCorrectVlan {
  my ($switch_ip, $ifIndex, $mac, $status, $vlan, $pid) = @_;
  my $logger = Log::Log4perl->get_logger();
  Log::Log4perl::MDC->put('tid', threads->self->tid());
  return 1;
}
```

If all your VLANs are spanned throughout the network, you might want to keep the default definition, which defines the VLAN saved in the node table to be the correct default VLAN for a given MAC.

If on the other hand, you have many VLANs depending on your physical location (switch, building, campus), you need to put some more effort into this function to define that a given computer must be put into VLAN A when connected into one switch and into VLAN B when connected into another switch.

Have look at the other functions and make sure they fit your needs.

Switch Definition

PacketFence needs to know which switches it manages and their type and configuration. All this information is stored in `/usr/local/pf/conf/switches.conf.`

This files contains a default section including:

- DB connection parameters
- 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 (so that the correct subclasses of pf::SNMP are instantiated)
- Switch uplink ports (trunks and non-managed ports)

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.

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

```
[default]
communityRead = public
communityWrite = private

communityTrap = public
version = 1
vlans = 1,2,3,4
normalVlan = 1
registrationVlan = 2
isolationVlan = 3
macDetectionVlan = 4
VoIPEnabled = no

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

[192.168.1.101]
ip = 192.168.1.101
type = Cisco::Catalyst_2960
mode = production
uplink = 25
```

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

Once you have modified `switches.conf` for your network, you can execute some first tests (only SNMP reads) using the supplied `/usr/local/pf/test/connect_and_read.pl` script.

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`.

See Appendix A for the complete list of all available parameters.

Here is the `pf.conf` file for our setup:

```
[general]
domain=yourdomain.org
dnsservers=192.168.2.1,192.168.3.1
dhcpservers=192.168.2.1,192.168.3.1

[network]
vlan=enabled

[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 eth0.1]
mask=255.255.255.0
type=internal,registration
gateway=192.168.2.1
ip=192.168.2.1

[interface eth0.2]
mask=255.255.255.0
type=internal,isolation
gateway=192.168.3.1
ip=192.168.3.1
```

```
[interface eth1]
mask=255.255.255.0
type=dhcplistener
gateway=192.168.1.5
ip=192.168.1.254

[interface eth2]
mask=255.255.255.0
type=monitor
gateway=192.168.1.5
ip=192.168.1.1
```

Iptables

You need to open some ports (53: DNS). Add the following lines to `/usr/local/pf/conf/iptables.pre`

```
*filter
:INPUT ACCEPT [0:0]
-A INPUT -p udp -m udp --dport 53 -i eth0.2 -j ACCEPT
-A INPUT -p udp -m udp --dport 53 -i eth0.3 -j ACCEPT
COMMIT
```

DHCP

The DHCP server will manage IP distribution in VLANs 2 and 3.

Put the following line in `/etc/sysconfig/dhcpd:`

```
DHCPDARGS="eth0.2 eth0.3"
```

Edit `/etc/dhcpd.conf` and replace its content with:

```
authoritative;
ddns-update-style none;
ignore client-updates;
subnet 192.168.2.0 netmask 255.255.255.0 {
  option routers 192.168.2.1;
  option subnet-mask 255.255.255.0;
  option domain-name "registration.example.com";
  option domain-name-servers 192.168.2.1;
  range 192.168.2.2 192.168.2.254;
  default-lease-time 300;
  max-lease-time 600;
}

subnet 192.168.3.0 netmask 255.255.255.0 {
  option routers 192.168.3.1;
```

```
  option subnet-mask 255.255.255.0;
  option domain-name "isolation.example.com";
  option domain-name-servers 192.168.3.1;
  range 192.168.3.2 192.168.3.254;
  default-lease-time 300;
  max-lease-time 600;
}
```

DNS

The DNS server will answer to all domain resolution requests in VLANs 2 and 3.

Create `/etc/named.conf` with the following content:

```
options {
  directory "/var/named";
  dump-file "/var/named/data/cache_dump.db";
  statistics-file "/var/named/data/named_stats.txt";
  listen-on { 192.168.2.1; 192.168.3.1; };
};

controls {
  inet 127.0.0.1 allow { localhost; } keys { rndckey; };
};

view "registration" {
   match-clients { 192.168.2.0/24; };
   zone "." IN {
     type master;
     file "named-registration.ca";
   };
};

view "isolation" {
   match-clients { 192.168.3.0/24; };
   zone "." IN {
     type master;
     file "named-isolation.ca";
   };
};

include "/etc/rndc.key";
```

Create `/var/named/named-registration.ca` with the following content:

```
$TTL 3600
. IN SOA pf. admin.example.com (
    2005061501  ; serial
```

```
    10800       ; refresh
    3600        ; retry
    604800      ; expire
    86400       ; default_ttl
)

       IN      NS      pf.
*.     IN      A       192.168.2.1
       IN      MX      5       pf.
1.2.168.192.in-addr.arpa.     IN      PTR       pf
```

Create `/var/named/named-isolation.ca` with the following content:

```
$TTL 3600
. IN SOA pf. admin.example.com (
    2005061501  ; serial
    10800       ; refresh
    3600        ; retry
    604800      ; expire
    86400       ; default_ttl
)

       IN      NS      pf.
*.     IN      A       192.168.3.1
       IN      MX      5       pf.
1.3.168.192.in-addr.arpa.     IN      PTR       pf
```

Violations

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
```

Starting 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.

Testing

PacketFence Web Interface

To test the PacketFence admin interface, go to the following URL :
https://pf.yourdomain.org:1443.

Log in using the “admin” user and the “qwerty” password.

VLAN Isolation

There many tests that you need to do in order to make sure everything works fine.

Make sure that VLANs 2,3 and 4 are not routed anywhere and can not communicate with the rest of the network:

- any device in VLAN 2 can communicate with PacketFence through (and only through) eth0.2
- any device in VLAN 2 can not communicate with any device in any other VLAN
- any device in VLAN 3 can communicate with PacketFence through (and only through) eth0.3
- any device in VLAN 3 can not communicate with any device in any other VLAN
- any device in VLAN 4 can not communicate with any device in any other VLAN

Make sure PacketFence receives traps from the switches:

- configure the Catalyst 2900 switch to send linkUp/linkDown traps to PacketFence
- configure the Catalyst 2960 switch to send port-security traps to PacketFence
- plug a device on each switch
- make sure `snmptrapd` writes a line in `/usr/local/pf/logs/snmptrapd.log`
- make sure each trap is correctly decoded by `pfsetvlan` in `/usr/local/pf/logs/pfsetvlan.log`

Make sure there are no error messages in `/usr/local/pf/logs/error*` nor in `/var/log/messages` while PacketFence starts

Plug an unregistered computer in a switch and make sure:

- the port is put in VLAN 2
- the computer gets an IP in VLAN 2
- any DNS request resolves to PacketFence (use nslookup (for example)
- the computer can access the registration web page

Register the computer by following the instructions in the registration web pages and make sure that when computer reboots it has access to VLAN 1.

Install Limewire on the test computer (Snort log its activity in `/var/log/snort/*`). Start using it and make sure:

- the computer is put in VLAN 3 (see `/var/log/messages` and `/usr/local/pf/logs/pfsevlan.log`)
- you can see a message in the browser explaining why the computer is isolated
- you can re-enable your network access on your own

Appendix A: pf.conf

Alerting

[alerting.admin_netbiosname]

type: text

description: NetBIOS name of administrative workstation to send alerts with "winpopup" action assigned.(default: EXAMPLE)

[alerting.emailaddr]

type: text

description: Email address to which notifications of rogue DHCP servers, violations with an action of "email", or any other PacketFence-related message goes to.(default: pf@localhost)

[alerting.fromaddr]

type: text

description: Email address from which notifications of rogue DHCP servers, violations with an action of "email", or any other PacketFence-related message are sent.

[alerting.log]

type: text

description: Log file where "log" actions are sent.(default: /usr/local/pf/logs/violation.log)

[alerting.smtpserver]

type: text

description: Server through which to send messages to the above emailaddr. (default: localhost)

[alerting.subjectprefix]

type: text

description: Subject prefix for email notifications of rogue DHCP servers, violations with an action of "email", or any other PacketFence-related message.(default: “PF Alert:”)

[alerting.wins_server]

type: text

description: WINS server to resolve NetBIOS name of administrative workstation to IP address. (default: 192.168.0.100)

Arp

[arp.cleanshutdown]

type: toggle

options: enabled|disabled (default: enabled)

description: If enabled, ARPs are sent to all trapped systems to re-point them to the correct gateway device at shutdown.

[arp.dhcp_timeout]

type: time

description: Used in detection of systems with static IP addresses. Looks for broadcast DHCPDISCOVERs and flags a node as rogue if it fails to see one before timer is exceeded. This value should be greater than 50% of your DHCP lease time. (default: 8h)

[arp.gw_timeout]

type: time

description: Used in detection of systems with statically-defined gateway ARP entries. If a system has not ARPed for the gateway within this interval, it is removed from the IP->MAC mappings and should be flagged as rogue by the next probe. (default: 1d)

[arp.heartbeat]

type: time

description: To eliminate the negative effects of switch flooding of poisoned ARPs on some (cough...cough...Netgear MR814v2) routers, we must first send a valid ARP to establish that the system is on-line. The heartbeat is the length of time between the initial "hello" and a poisoned "goodbye". (default: 30s)

[arp.interval]

type: time

description: Interval at which poisoned ARPs ("traps") are sent to infected/unregistered systems. (default: 60s)

[arp.strobe]

type: toggle

options: enabled|disabled (default: enabled)

description: If enabled, sends ARP request to all IP addresses within range immediately after startup. This allows for the internal MAC to IP mappings to be populated quickly.

[arp.stuffing]

type: toggle

options: enabled|disabled (default: disabled)

description: If enabled, forces PacketFence system to "stuff" router ARP cache with a bogus MAC for systems that are not responding. This option effectively increases the "stickiness" of traps by suppressing broadcast ARP traffic from the gateway. It is also somewhat dangerous in that it relies on systems to issue a GARP (gratuitous ARP) at boot to reclaim previously stuffed addresses.

[arp.timeout]

type: time

description: Length of time of inactivity after which an unresponsive system is aged out. Hello

ARPs are sent at timeout/2 and timeout-interval to avoid prematurely timing out a system. (default: 8h)

Database

[database.db]

type: text

description: Name of the MySQL database used by PacketFence. (default: pf)

[database.host]

type: text

description: Server the MySQL server is running on. (default: localhost)

[database.pass]

type: text

description: Password for the MySQL database used by PacketFence. (default: packet)

[database.port]

type: numeric

description: Port the MySQL server is running on. (default: 3306)

[database.user]

type: text

description: Username of the account with access to the MySQL database used by PacketFence. (default: pf)

Dhcp

[dhcp.isolation_lease]

type: text

description: Optional lease time for the isolation scope. (default: 2m)

[dhcp.isolation_scopes]

type: text

description: List of scope definitions that isolated clients are assigned to.

[dhcp.registered_lease]

type: text

description: Optional lease time for the registered scope.(default: 2h)

[dhcp.registered_scopes]

type: text

description: List of scope definitions that registered (and non-isolated) clients are assigned to.

[dhcp.unregistered_lease]

type: text

description: Optional lease time for the unregistered scope.(default: 2m)

[dhcp.unregistered_scopes]

type: text

description: List of scope definitions that unregistered clients are assigned to. This is the "default" scope for a new client.

Expire

[expire.iplog]

type: time

description: Time which you would like to keep logs on IP/MAC information A value of 0d disables expiration. (default: 180d)

[expire.locationlog]

type: time

description: Time which you would like to keep logs on location information. Please note that this table should not become too big since it could degrade pfsetvlan performance. A value of 0d disables expiration. (default: 180d)

[expire.node]

type: time

description: Time before a node is removed due to inactivity. A value of 0d disables expiration. (default: 90d)

General

[general.caching]

type: toggle

options: enabled|disabled (default: enabled)

description: Enable caching of isinternal values as well as other fun stuff. Leave this enabled or suffer the performance consequences.

[general.dhcpservers]

type: text

description: Comma-delimited list of DHCP servers. Passthroughs are created to allow DHCP transactions from even "trapped" nodes. (default: 127.0.0.1)

[general.dnsservers]

type: text

description: Comma-delimited list of DNS servers. Passthroughs are created to allow queries to these servers from even "trapped" nodes. (default: 127.0.0.1)

[general.domain]

type: text

description: Domain name of the PacketFence system (default: example.com)

[general.hostname]

type: text

description: Hostname of PacketFence system. This is concatenated with the domain in Apache rewriting rules and therefore must be resolvable by clients. (default: abc)

[general.locale]

type: text

description: Locale used for message translation (default: en_US)

[general.logo]

type: text

description: Logo displayed on web pages.

Interface

[interface.authorizedips]

type: text

description: (Optional) list of IPs/subnets to authorize on this interface. If not specified, all IPs are authorized to connect. This can be used for example to limit access to the management interface to some specific hosts.

[interface.gateway]

type: text

description: Gateway of the named interface.

[interface.ip]

type: text

description: IP address of the named interface - note that this should mirror the OS-level configuration but it does not make any OS-level changes.

[interface.mask]

type: text

description: Network mask of the named interface.

[interface.type]

type: multi

options: internal|managed|monitor|dhcplistener|isolation|registration

description: Describes "type" of named interface. Internal describes internal client networks, managed (aka external) interfaces have the administrative GUI running on them, monitor is the interface that snort listens on and dhcplistener is an interface connected to a SPAN of the DHCP traffic.

Logging

[logging.facility]

type: text

description: Syslog facility to log on.

[logging.priority]

type: text

description: Syslog priority to log at.

[logging.verbosity]

type: numeric

description: Logging verbosity level. 4 is good value for day-to-day operation. For minor troubleshooting, use 8. To see database queries, use 12. For everything, use 20. For only errors, use 1.

Network

[network.dhcpdetector]

type: toggle

options: enabled|disabled (default: enabled)

description: If enabled, PacketFence will monitor DHCP-specific items such as rogue DHCP services, DHCP-based OS fingerprinting, computername/hostname resolution, and (optionnally) option-82 location-based information. The monitored DHCP packets are DHCPDISCOVERs and DHCPREQUESTs - both are broadcasts, meaning a span port is not necessary. This feature is highly recommended if the internal network is DHCP-based.

[network.dhcpoption82logger]

type: toggle

options: enabled|disabled (default: disabled)

description: If enabled PacketFence will monitor DHCP option82 location-based information.

This feature is only available if the dhcpdetector is activated.

[network.mode]

type: toggle

options: passive|inline|dhcp (default: passive)

description: Defines the mode in which PacketFence will operate.

When deployed in-line, PacketFence acts as a router and requires internal and external interfaces to "live" on separate networks. It's also likely that a static route for the internal network will need to be added to the upstream router. The PacketFence system can act as a DHCP server or relay to one or more external servers.

When deployed in passive mode, PacketFence uses ARP manipulation inject itself into the datastream trapped nodes. ARP is a protocol that allows IP addresses to be mapped to the underlying data-link protocol (eg.Ethernet). ARP is an insecure protocol – relying on each host

to respond only when its “name” is called. The responses are stored in a cache on each end system for a short time (typically 5-20 minutes) and are used to deliver packets on the local network. For more information, please visit Wikipedia. Passive deployment has several benefits over an inline deployment including elimination of a performance bottleneck and single point of failure. Its major failing is that it's not 100% in catching all traffic - spurious packets can and will occasionaly get through. In an academic environment or environments where in-line devices are frowned upon, this failing is minor in relation to the benefits.

[network.named]

type: toggle

options: enabled|disabled (default: disabled)

description: If enabled, run a nameserver locally. Combined with a 53/udp redirection port, this can allow you redirect clients based on name resolution versus HTTP interception. There are several caveats to keep in mind, First, many clients cache DNS responses which may interrupt connectivity even after successful registration/remediation. Second, in practice we've noticed issues with the local nameserver refusing to answer queries in some cases - this may be related to netfilter connection tracking.

If you're running DHCP locally, though, it may make sense to run a nameserver locally as well rather than defining external servers to passthrough. Not that running either DHCP or DNS on a passive deployed PacketFence system establishes dependencies on it that are likely not wanted.

[network.nat]

type: toggle

options: enabled|disabled (default: disabled)

description: If enabled, NATs outgoing traffic to the external interface IP address. This setting is only useful in an in-line deployment. Enabling in a passive environment will likely cause network issues for trapped nodes. Enabling this option also forces snort to listen on the internal interface - this could have performance implications in high-throughput environments.

[network.rogueinterval]

type: numeric

description: When rogue DHCP server detection is enabled, this parameter defines how often to email administrators. With its default setting of 10, it will email administrators the details of the previous 10 DHCP offers. (default: 10)

[network.vlan]

type: toggle

options: enabled|disabled (default: disabled)

description: If enabled, VLAN based isolation is used.

PassThroughs

[passthroughs]

description: This section allows you to create passthroughs to HTML content or remote addresses/networks. Here's an example:

```
packetfence=http://www.packetfence.org
```

The above will allow 80/tcp traffic to the resolved IP address (the LHS value is arbitrary). Passthroughs can also take the form of:

```
test=192.168.100.10/23
```

which would allow full IP to all 512 destination addresses.

(default: packetfence=http://www.packetfence.org
symantec_scanner=http://security.symantec.com)

Ports

[ports.admin]

type: text

description: Port the administrative interface listens on.(default: 1443)

[ports.allowed]

type: text

description: Ports allowed through the PacketFence system regardless of registration or violation status. It is not necessary to define 53/udp is DNS servers are defined as passthroughs are automagically added.

[ports.listeners]

type: multi

options: imap|pop3

description: Enables "bogus" IMAP and POP servers. These servers serve only to deliver a message (POP3) or send an alert (IMAP) to inform the user that he/she must register before connectivity is allowed. Content of the message is found at `/usr/local/pf/conf/templates/listener.msg`

[ports.redirect]

type: text

description: Ports to intercept and redirect for trapped and unregistered systems. IMAP and POP3 listeners must be enabled via the listeners parameter if the redirection is to be of any use. Redirecting 443/tcp (SSL) will work, although users will get ugly and confusing pop-ups as the common name will no longer match. Redirecting 53/udp (DNS) seems to have issues and is also not recommended.(default: “80/tcp,110/tcp,143/tcp”)

Proxies

[proxies]

description: This section allows you to configure locally proxied content. We typically use this to proxy tools like Stinger rather than having to continually download the latest version. (default: tools/stinger.exe=http://download.nai.com/products/mcafee-avert/stinger.exe)

The Stinger utility could then be accessed at https://pfhostname/proxies/tools/stinger.exe.

Registration

[registration.aup]

type: toggle

options: enabled|disabled (default: disabled)

description: If enabled, users will be required to accept an Acceptable Use Policy before network access is allowed. Currently, registration must be enabled for the AUP to take effect. The AUP text is found at `/usr/local/pf/html/user/content/violations/aup.php`

[registration.auth]

type: multi

options: local|ldap|radius (default: local)

description: Method or Methods by which registering nodes will be authenticated. Templates for LDAP and local are available at `/usr/local/pf/conf/authentication/`. If you wish to use a different authentication mechanism, simply create a file called `/usr/local/pf/conf/authentication/<authname>.pm`, fill it with the necessary data, and set `auth=<authname>`. The default value local relies on a local access file in `/usr/local/pf/conf/user.conf`.

[registration.button_text]

type: text

description: The button text will appear on the registration page submit button. (default: Register)

[registration.completemsg]

type: toggle

options: enabled|disabled

description: If enabled, a confirmation screen is displayed after a user successfully registered. This is useful if you believe the registration process isn't sufficiently clear to users. The HTML file displayed is found at `/usr/local/pf/html/user/content/violations/reg_complete.php`

[registration.expire_deadline]

type: date

description: If expire_mode is set to "deadline", this is the date (formatted as returned by the "date" command) at which nodes revert to an unregistered state. This would typically be the end of a semester. (default: “Mon Nov 12 12:00:00 EST 2012”)

[registration.expire_mode]

type: toggle

options: window|deadline|session|disabled (default: disabled)

description: If set to "deadline", the expire_deadline option defines the date at which a node reverts to an unregistered state. If set to "window", the window is used to determine the length of time after registration that a node remains registered. If set to "session", it specifies that a

client should be unregistered as soon as its iplog entry closes (or with a bit of latency - check regitration.expire_session).

[registration.expire_session]

type: time

description: If expire_mode is set to "session", this is the amount of time after a node's iplog entry is closed that it reverts to an unregistered state. (default: 5m)

[registration.expire_window]

type: time

description: If `expire_mode=window`, this is length of time after registration that a node reverts to an unregistered state. (default: 52w)

[registration.maxnodes]

type: numeric

description: If defined, the maximum number of nodes that can be registered to a single PID. (default: 0)

[registration.queuesize]

type: numeric

description: Useful for passive deployments on very large networks, this defines the number of nodes that PacketFence will simultaneously trap for registration (trappings due to violation always occur). If set to 0, this queue is disabled. (default: 0)

registration.skip_deadline

type: date

description: If `skip_mode=deadline`, this is the date at which the "skip registration" option is disabled. Date string is formatted as the output of the "date" command is. (default: “Mon Nov 12 12:00:00 EST 2012”)

[registration.skip_mode]

type: toggle

options: window|deadline|disabled (default: disabled)

description: If set to "deadline", the deadline option defines the time at which skipping registration is no longer an option for clients. If set to "window", the window is used to determine the amount of time after first network access that a node may skip registration.

[registration.skip_reminder]

type: time

description: Interval that a user is re-prompted to register after skipping. For example, if `window=2w` and `reminder=1d`, a user will be allowed to skip for two weeks but will be re-prompted every day. (default: 1d)

[registration.skip_window]

type: time

description: The length of time that a node may skip registration. For instance, setting it to 2880 minutes would allow students to skip registration for two days, giving them time to get a student ID, password, etc. (default: 14d)

Routedsubnet

[routedsubnet.gateway]

type: text

description: Gateway of the named routed subnet.

[routedsubnet.mask]

type: text

description: Network mask of the named routed subnet.

[routedsubnet.network]

type: text

description:Network of the named routed subnet.

[routedsubnet.type]

type: multi

options: isolation|registration

description: Describes "type" of named routed subnet.

Scan

[scan.host]

type: text

description: Host the nessus server is running on. For performance reasons, we recommend running the nessus server remotely. A passthrough will be automagically created. (default: 127.0.0.1)

[scan.live_tids]

type: text

description: If a host fails a scan AND the tid is listed in live_tids the corresponding violation will be added. If the tid is not listed here the event will be logged only. This is used to test Nessus plugins before going live.

[scan.pass]

type: text

description: Password to log into nessus server with. (default: packet)

[scan.port]

type: text

description: Port nessus server is running on. (default: 1241)

[scan.registration]

type: toggle

options: enabled|disabled (default: disabled)

description: If this option is enabled, the PacketFence system will scan each host after registration is complete with all nessusids.

[scan.ssl]

type: toggle

options: enabled|disabled (default: enabled)

description: enable ssl communication with the nessus server.

[scan.user]

type: text

description: Username to log into nessus server with. (default: admin)

Scope

[scope.gateway]

type: text

description: Network gateway of scope.

[scope.network]

type: text

description: Network (in CIDR or prefix notation) of the subnet encompassing the DHCP ranges.

[scope.range]

type: text

description: Address range eligible for DHCP assignment. A comma-delimited list of networks of the form:

`a.b.c.0/24`

`a.b.c.0-255`

`a.b.c.0-a.b.c.255`

`a.b.c.d`

Services

[services.dhcpd]

type: text

description: Location of the dhcpd binary. Only necessary to change if you are not running the RPMed version. DHCP is not supported until PacketFence 1.6(default: /usr/sbin/dhcpd)

[services.httpd]

type: text

description: Location of the apache binary. Only necessary to change if you are not running the RPMed version.(default: /usr/sbin/httpd)

[services.named]

type: text

description: Location of the named binary. Only necessary to change if you are not running the RPMed version. (default: /usr/sbin/named)

[services.snmptrapd]

type: text

description: Location of the snmptrapd binary. Only necessary to change if you are not running the RPMed version. (default: /usr/sbin/snmptrapd)

[services.snort]

type: text

description: Location of the snort binary. Only necessary to change if you are not running the RPMed version. (default: /usr/sbin/snort)

Servicewatch

[servicewatch.email]

type: toggle

options: enabled|disabled (default: enabled)

description: Should '`pfcmd service pf watch`' send an email when services are not running

[servicewatch.restart]

type: toggle

options: enabled|disabled (default: disabled)

description: Should '`pfcmd service pf watch`' restart PacketFence when services are not running

Trapping

[trapping.blacklist]

type: text

description: Comma-delimited list of MAC addresses that are not allowed to pass through the PacketFence system.

[trapping.detection]

type: toggle

options: enabled|disabled (default: disabled)

description: Enables snort-based worm detection. If you don't have a span interface available, don't bother enabling it. If you do, you'll most definitely want this on.

[trapping.immediate]

type: toggle

options: enabled|disabled (default: disabled)

description: Enable this if you want to see lots of "IP conflict boxes on Windows systems! On detection of a violation, a spoofed GARP (gratuitous ARP) is sent to the offending system. This causes it to think another system is using its IP address and, under Windows 2000, causes it to disable its IP stack. When the user manages to get the system back on the wire (ipconfig /release, reboot, etc) he/she will be assigned an address from the isolation scope.

[trapping.passthrough]

type: toggle

options: iptables|proxy (default: iptables)

description: Method by which content is delivered to trapped systems. When set to "proxy", PacketFence uses Apache's reverse proxy functionality and the mod_proxy_html module to rewrite links. Note that links external servers will not be properly rewritten. When set to "iptables", PacketFence creates passthroughs to the content for only those nodes trapped with the corresponding violation. Be aware that an iptables passthrough is based on IP address and clients will be able to get to ALL content on the destination site.

[trapping.range]

type: text

description: Address ranges/CIDR blocks that PacketFence will monitor/detect/trap on. Gateway, network, and broadcast addresses are ignored. Comma-delimited entries should be of the form:

```
a.b.c.0/24
```

```
a.b.c.0-255
```

```
a.b.c.0-a.b.c.255
```

```
a.b.c.d
```

(default: 192.168.0.0/24)

[trapping.redirecturl]

type: text

description: Default URL to redirect to on registration/mitigation release. This is only used if a per-violation redirecturl is not defined. (default: http://www.packetfence.org)

[trapping.redirlocal]

type: toggle

options: enabled|disabled (default: disabled)

description: Typically best to leave this disabled unless you are having problems and understand why you need this.

[trapping.redirtimer]

type: time

description: How long to display the progress bar during trap release. Setting it to a value of 5 or higher is recommended when in passive mode. Doing so allows the client time to receive and process the redirection ARP sent by PacketFence. (default: 10s)

[trapping.registration]

type: toggle

options: enabled|disabled (default: disabled)

description: If enabled, nodes will be required to register on first network access. Further registration options are configured in the registration section.

[trapping.testing]

type: toggle

options: enabled|disabled (default: enabled)

description: Disables sending of ARPs - note that this has implications on node detection and timeouts.

[trapping.whitelist]

type: text

description: Comma-delimited list of MAC addresses that are immune to registration/trapping and are always allowed to pass. Useful for monitored switches, etc.

Vlan

[vlan.adjustswitchportvlanreasons]

type: multi

options: node_modify|manage_register|manage_deregister|manage_vclose|manage_vopen|violation_modify|violation_add|violation_delete

description: After which calls to pfcmd do we have to re-calculate and re-assign the switch port VLAN a node is connected to. (default: node_modify|manage_register|manage_deregister|manage_vclose|manage_vopen|violation_modify|violation_add|violation_delete)

[vlan.adjustswitchportvlanscript]

type: text

description: Script that adjusts the switch port VLAN. (default: /usr/local/pf/bin/flip.pl)

[vlan.closelocationlogonstop]

type: toggle

options: enabled|disabled (default: enabled)

description: Should open locationlog entries be closed when pfsetvlan is stopped

[vlan.nbtraphandlerthreads]

type: text

description: Number of trap handler threads pfsetvlan should start. (default: 20)

[vlan.nbtrapparserthreads]

type: text

description: Number of trap parser threads pfsetvlan should start. (default: 5)

Appendix B: Switches supported by PacketFence

PacketFence supports the following switches:

Vendor	Model	PacketFence Type (used in switches.conf)
3COM	NJ220	3COM::NJ220
	SuperStack 3 Switch 4500	3COM::SS4500
Cisco	Catalyst 2900XL	Cisco::Catalyst_2900XL
	Catalyst 2950	Cisco::Catalyst_2950
	Catalyst 2960	Cisco::Catalyst_2960
	Catalyst 2970	Cisco::Catalyst_2970
	Catalyst 3500XL	Cisco::Catalyst_3500XL
	Catalyst 3550	Cisco::Catalyst_3550
	Catalyst 3560	Cisco::Catalyst_3560
	Controller 4400	Cisco::Controller_4400_4_2_130
D-Link	DES 3526	Dlink::DES_3525
Dell	PowerConnect 3424	Dell::PowerConnect3424
Edge-corE	3526XA	Accton::ES3536XA
	3528M	Accton::ES3528M
Enterasys	SecureStack C2	Enterasys::SecureStack_C2
HP ProCurve	2500	HP::Procurve_2500
	2600	HP::Procurve_2600
	4100	HP::Procurve_4100
Intel	Express 460	Intel::Express_460
	Express 530	Intel::Express_530
Linksys	SRW224G4	Linksys::SRW224G4
Nortel	BPS2000	Nortel::BPS2000
	ES325	Nortel::ES325
	Baystack 470	Nortel::Baystack470

	Baystack 4550	Nortel::Baystack4550
	Baystack 5520	Nortel::Baystack5520

Appendix C: Switch Configuration

Assumptions

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

- PacketFence IP address: 192.168.1.5
- MAC Detection VLAN: 4
- VoIP, Voice VLAN: 100
- community name used by the switches to send traps to PacketFence: public

3COM

PacketFence supports D-Link switches with no VoIP using 1 trap type:

- linkUp/linkDown

Don't forget to update the startup config !

SuperStack 3 Switch 4500

- Global config settings

```
snmp-agent

snmp-agent target-host trap address udp-domain 192.168.1.5 params
securityname public

snmp-agent trap enable standard linkup linkdown
```

- On each interface:

```
port access vlan 4
```

Cisco

PacketFence supports Cisco switches with VoIP using 3 different trap types:

- •linkUp/linkDown
- •MAC Notification
- •Port Security (with static MACs)

Enable either linkUp/linkDown and MAC notification together or Port Security only (When possible, we recommend Port Security), see below for details.

Don't forget to update the startup config !

2900XL and 3500XL

Those switches do not support port-security with static MAC address so we enable linkUp/linkDown and MAC notification traps.

- Global config settings:

```
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 with no VoIP:

```
switchport mode access

switchport access vlan 4

snmp trap mac-notification added
```

- On each interface with VoIP:

```
switchport trunk encapsulation dot1q

switchport trunk native vlan 4

switchport mode trunk

switchport voice vlan 100

snmp trap mac-notification added

snmp trap mac-notification removed
```

2950 and 3550

Those switches support port-security with static MAC address but we can not secure a MAC on the data VLAN specifically so enable it if there is no VoIP, use linkUp/linkDown and MAC notification otherwise.

With port-security, if no MAC is connected on ports when activating port-security, we need to secure bogus MAC addresses on ports in order for the switch to send a trap when a new MAC appears on a port. On the other hand, if a MAC is actually connected when you enable port security, you must secure this MAC raher than the bogus one. Otherwise this MAC will lose its connectivity instantly.

- Global config settings with no VoIP

```
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 with no VoIP

```
switchport mode access

switchport access vlan 4

switchport port-security

switchport port-security violation restrict

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

where xx stands for the interface index.

- Global config settings with VoIP:

```
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 with VoIP

```
switchport voice vlan 100

switchport access vlan 4

switchport mode access

snmp trap mac-notification added

snmp trap mac-notification removed
```

2960, 2970, 3560

Those switches support port-security with static MAC address and allow us to secure a MAC on the data VLAN so we enable it whether .there is VoIP or not.

We need to secure bogus MAC addresses on ports in order for the switch to send a trap when a new MAC appears on a port.

- Global config settings

```
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 with no VoIP:

```
switchport access vlan 4

switchport port-security

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

- On each interface with VoIP:

```
switchport voice vlan 100
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

D-Link

PacketFence supports D-Link switches with no VoIP using 2 different trap types:

- linkUp/linkDown
- MAC Notification

We recommend to enable linkUp/linkDown and MAC notification together.

Don't forget to update the startup config !

DES3526

Those switches support port-security with static MAC address and allow us to secure a MAC on the data VLAN so we enable it whether .there is VoIP or not.

- Global config settings

```
to be completed...
```

- On each interface:

```
to be completed...
```

Dell

This section is under construction.

Edge-corE

PacketFence supports Edge-corE switches with no VoIP using 1 trap type:

- linkUp/linkDown

Don't forget to update the startup config !

3526XA and 3528M

- Global config settings

```
SNMP-server host 192.168.1.5 public version 2c udp-port 162
```

Enterasys

PacketFence supports Enterasys switches with no VoIP using 2 different trap types:

- linkUp/linkDown
- MAC Locking (Port Security with static MACs)

We recommend to enable enable MAC locking only.

Don't forget to update the startup config !

SecureStack C2

linkUp/Lindown traps are enabled by default so we disable them and enable MAC locking only.

- Global config settings

```
set snmp community public

set snmp targetparams v2cPF user public security-model v2c message-
processing v2c

set snmp notify entryPF tag TrapPF

set snmp targetaddr tr 192.168.1.5 param v2cPF taglist TrapPF

set maclock enable
```

- On each interface:

```
set port trap fe.1.xx disable

set maclock enable fe.1.xx

set maclock static fe.1.xx 1

set maclock firstarrival fe.1.xx 0
```

where xx stands for the interface index

HP ProCurve

PacketFence supports ProCurve switches with no VoIP using 2 different trap types:

- linkUp/linkDown
- Port Security (with static MACs)

We recommend to enable enable Port Security only.

Don't forget to update the startup config !

2500

linkUp/Lindown traps are enabled by default so we disable them and enable Port Security only.

On 2500's, we need to secure bogus MAC addresses on ports in order for the switch to send a trap when a new MAC appears on a port.

- Global config settings

```
snmp-server community "public" Unrestricted
```

```
snmp-server host 192.168.1.5 "public" Not-INFO

no snmp-server enable traps link-change 1-26
```

- On each interface:

  ```
  port-security xx learn-mode static action send-alarm mac-address
  0200000000xx
  ```

  where xx stands for the interface index

2600

linkUp/Lindown traps are enabled by default so we disable them and enable Port Security only.

On 2600's, we don't need to secure bogus MAC addresses on ports in order for the switch to send a trap when a new MAC appears on a port.

- Global config settings

  ```
  snmp-server community "public" Unrestricted

  snmp-server host 192.168.1.5 "public" Not-INFO

  no snmp-server enable traps link-change 1-26
  ```

- On each interface:

  ```
  port-security xx learn-mode configured action send-alarm
  ```

  where xx stands for the interface index

2500

linkUp/Lindown traps are enabled by default so we disable them and enable Port Security only.

On 2500's, we need to secure bogus MAC addresses on ports in order for the switch to send a trap when a new MAC appears on a port.

- Global config settings

  ```
  snmp-server community "public" Unrestricted

  snmp-server host 192.168.1.5 "public" Not-INFO

  no snmp-server enable traps link-change 1-26
  ```

- On each interface:

  ```
  port-security xx learn-mode static action send-alarm mac-address
  0200000000xx
  ```

  where xx stands for the interface index

4100

linkUp/Lindown traps are enabled by default and we have not found a way yet to disable them so do not forget to declare the trunk ports as uplinks in the switch config file.

On 4100's, we need to secure bogus MAC addresses on ports in order for the switch to send a trap when a new MAC appears on a port. The ports are indexed differently on 4100's: it is based on the number of modules you have in your 4100. Each module is indexed with a letter (A,B,C, …)

- Global config settings

  ```
  snmp-server community "public" Unrestricted

  snmp-server host 192.168.1.5 "public" Not-INFO

  no snmp-server enable traps link-change 1-26
  ```

- You should configure interfaces like this:

  ```
  port-security A1 learn-mode static action send-alarm mac-address
  020000000001

  ...

  port-security A24 learn-mode static action send-alarm mac-address
  020000000024

  port-security B1 learn-mode static action send-alarm mac-address
  020000000025

  ...

  port-security B24 learn-mode static action send-alarm mac-address
  020000000048

  port-security C1 learn-mode static action send-alarm mac-address
  020000000049

  ...
  ```

```
port-security C24 learn-mode static action send-alarm mac-address
020000000072
```

Intel

This section is under construction.

Linksys

PacketFence supports Linksys switches with no VoIP using 1 trap type:

- linkUp/linkDown

Don't forget to update the startup config !

SRW224G4

- Global config settings

```
no snmp-server trap authentication

snmp-server community CS_2000_le rw view Default

snmp-server community CS_2000_ls ro view Default

snmp-server host 192.168.1.5 public 2
```

- On each interface

```
switchport access vlan 4
```

Nortel

PacketFence supports Nortel switches with VoIP using 1 trap type:

•Mac Security

Don't forget to update the startup config !

NOTE: if you are using a 5520 in a stack, you must declare it as a Nortel::BayStack5520Stacked in /usr/local/pf/conf/switches.conf. Indeed, when stacked, this switch refers to its ifindex differently than when not stacked so there is some specific code in a different perl module.

470, 4550, 5520 and ES325

- Global config settings

```
snmp-server authentication-trap disable

snmp-server host 192.168.1.5 "public"

snmp trap link-status port 1-24 disable

no mac-security mac-address-table

interface FastEthernet ALL

mac-security port ALL disable

mac-security port 1-24 enable

default mac-security auto-learning port ALL max-addrs

exit

mac-security enable

mac-security snmp-lock disable

mac-security intrusion-detect disable

mac-security filtering enable

mac-security snmp-trap enable

mac-security auto-learning aging-time 60

mac-security learning-ports NONE

mac-security learning disable
```

BPS2000

You can only configure this switch through menus.

- Enable “MAC Address Security”:

```
MAC Address Security: Enabled

MAC Address Security SNMP-Locked: Disabled

Partition Port on Intrusion Detected: Disabled

DA Filtering on Intrusion Detected: Enabled

Generate SNMP Trap on Intrusion: Enabled

Current Learning Mode: Disabled

Learn by Ports: NONE


Port  Trunk  Security

----  -----  --------

   1         Enabled

 ...

  24         Enabled
```

Appendix D: Additional Softwares

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-Net-Nessus-Client
- perl-Net-Nessus-Message
- perl-Net-Nessus-ScanLite

Please visit http://www.nessus.org/download/ to download and install Nessus.

Snort

If you plan on using Snort as a network intrusion prevention and detection system, we encourage the usage of oinkmaster to manage your snort rules.

Please visit http://www.snort.org/dl/ to download and install Snort.

Oinkmaster

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

Chapter 11

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.

Chapter 13

GNU Free Documentation License

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