PacketFence
– version 1.7.5

Administration Guide

Version 1.7.5 – December 2008

Contents

Chapter 1

About this Guide

This guide will walk you through the day to day administration of the PacketFence solution. It covers mainly VLAN isolation.

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_Administration_Guide.pdf.

Release Notes

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”

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

Chapter 4

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

fingerprint            | view DHCP Fingerprints

graph                  | trending graphs

history                | IP/MAC history

ifoctetshistorymac     | accounting history

ifoctetshistoryswitch  | accounting history

ifoctetshistoryuser    | accounting history

ipmachistory           | IP/MAC history

locationhistorymac     | Switch/Port history

locationhistoryswitch  | Switch/Port history

lookup                 | node or pid lookup against local data store

node                   | node manipulation
```

```
graph                  | trending graphs

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

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



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.

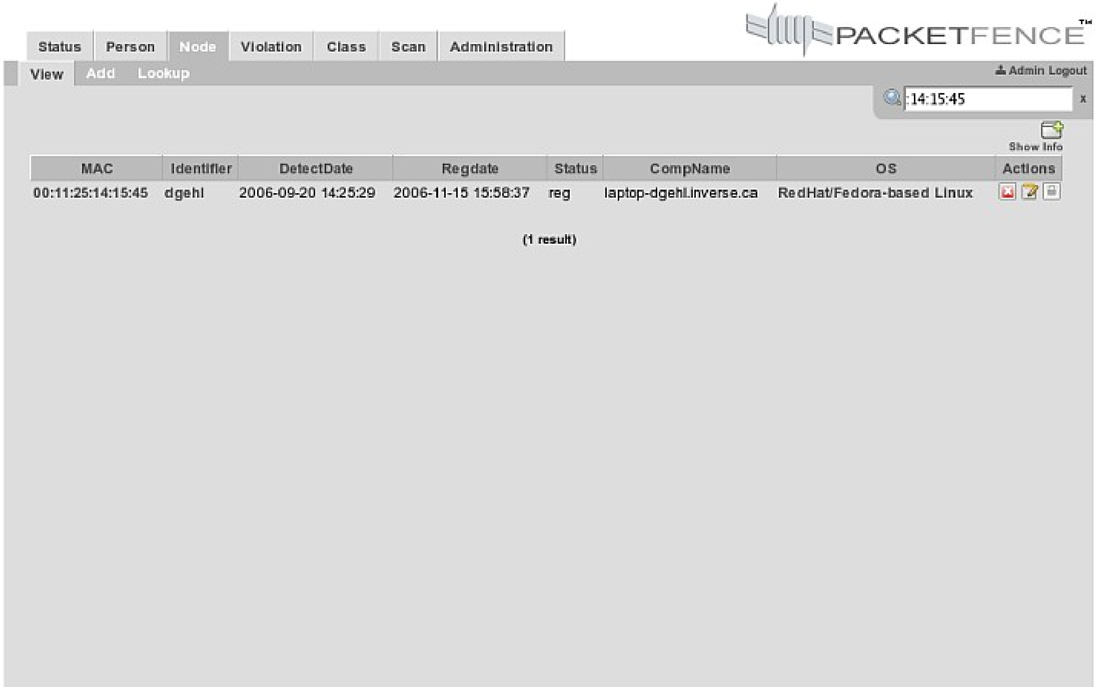
Again, when executed without any arguments, a help screen is shown.

```
# /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
```

Chapter 4

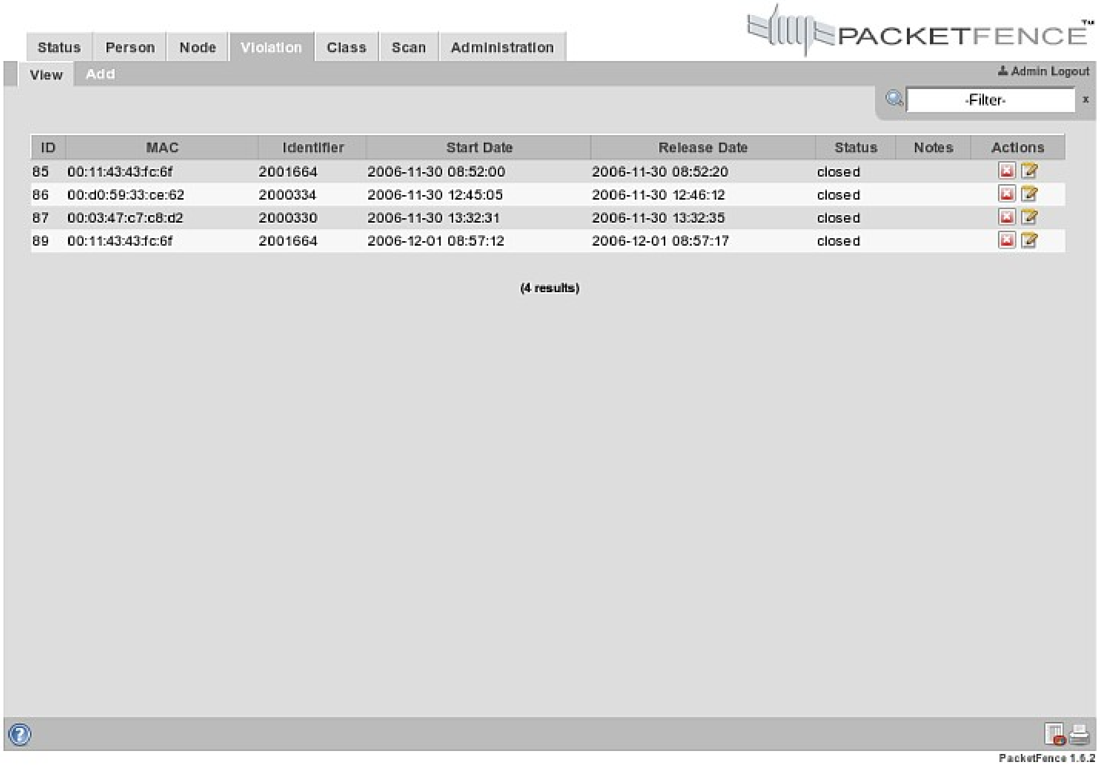
Web Admin GUI

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



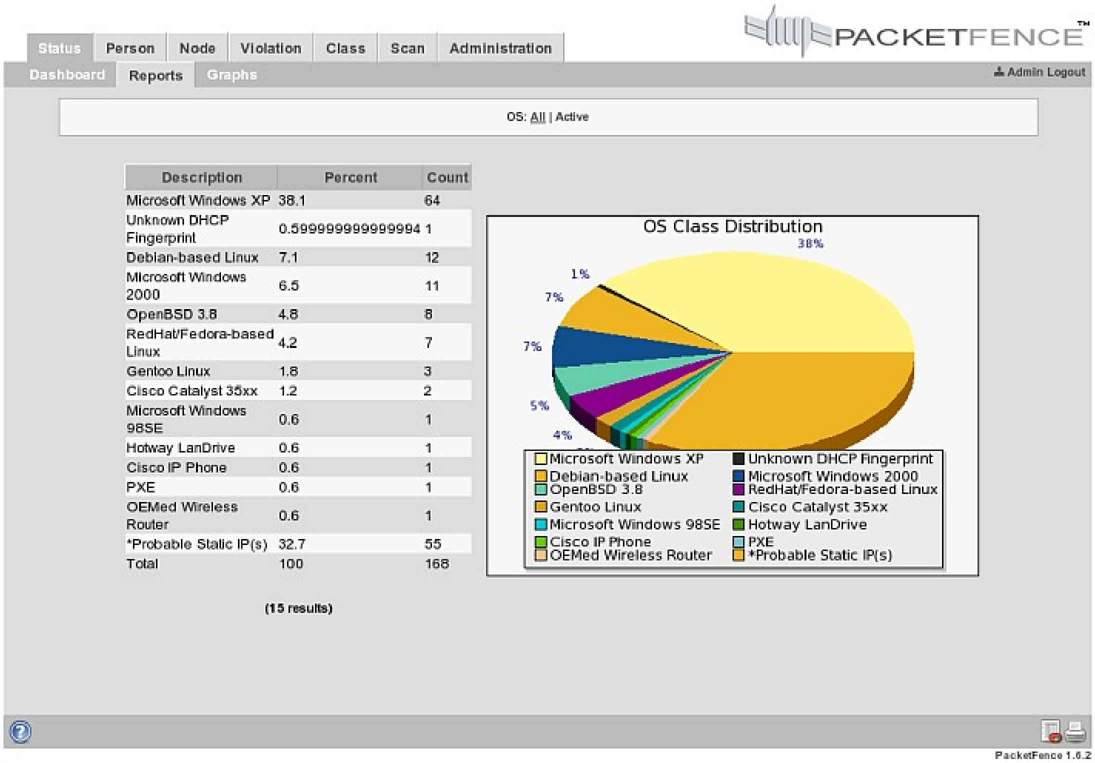
MAC	Identifier	DetectDate	Regdate	Status	CompName	OS	Actions
00:11:25:14:15:45	dgehl	2006-09-20 14:25:29	2006-11-15 15:58:37	reg	laptop-dgehl.inverse.ca	RedHat/Fedora-based Linux	

(1 result)



ID	MAC	Identifier	Start Date	Release Date	Status	Notes	Actions
85	00:11:43:43:fc:6f	2001664	2006-11-30 08:52:00	2006-11-30 08:52:20	closed		
86	00:d0:59:33:ce:62	2000334	2006-11-30 12:45:05	2006-11-30 12:46:12	closed		
87	00:03:47:c7:c8:d2	2000330	2006-11-30 13:32:31	2006-11-30 13:32:35	closed		
89	00:11:43:43:fc:6f	2001664	2006-12-01 08:57:12	2006-12-01 08:57:17	closed		

(4 results)



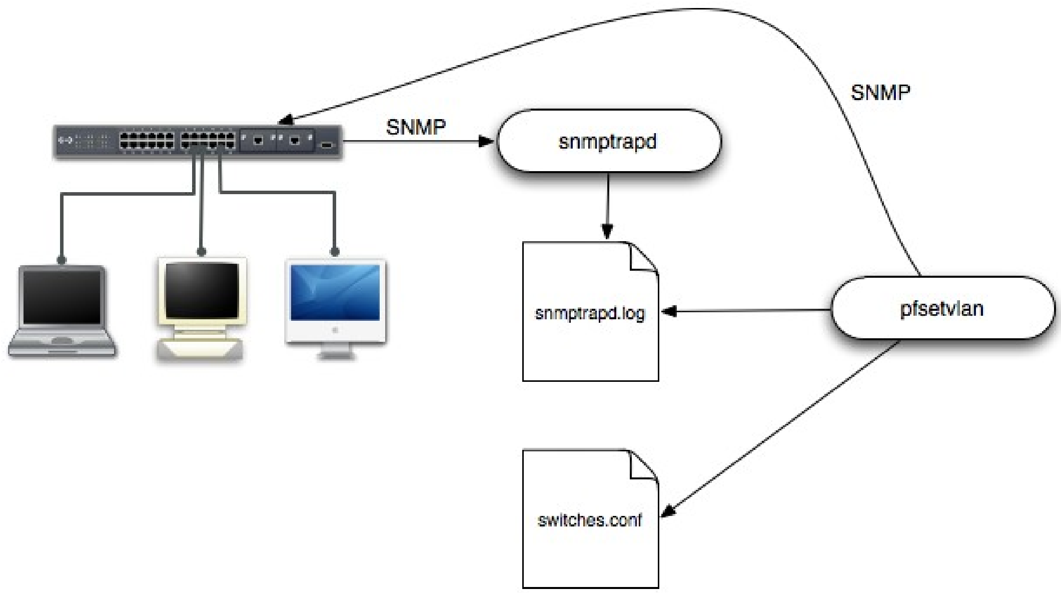
Description	Percent	Count
Microsoft Windows XP	38.1	64
Unknown DHCP Fingerprint	0.599999999999994	1
Debian-based Linux	7.1	12
Microsoft Windows 2000	6.5	11
OpenBSD 3.8	4.8	8
RedHat/Fedora-based Linux	4.2	7
Gentoo Linux	1.8	3
Cisco Catalyst 35xx	1.2	2
Microsoft Windows 98SE	0.6	1
Hotway LanDrive	0.6	1
Cisco IP Phone	0.6	1
PXE	0.6	1
OEMed Wireless Router	0.6	1
*Probable Static IP(s)	32.7	55
Total	100	168

(15 results)

VLAN isolation

Introduction

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.

Supported Switches

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

Chapter 5

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
```

Chapter 6

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.

Chapter 7

Integration with wireless networks

PacketFence integrates very well with wireless networks. As for its wired counterpart, the switch, a wireless Access Points (AP) needs to implement some specific features in order for the integration to work perfectly. In particular, the AP needs to support

- several SSIDs with several VLANs inside each SSID
- authentication against a RADIUS server
- dynamic VLAN assignment (through RADIUS attributes)
- SNMP deauthentication traps
- the deauthentication of an associated station

We can then configure two SSIDs on the AP, the first one reserved for visitors and unregistered clients. In this SSID, communications will not be encrypted and users will connect either to the registration VLAN or the visitors VLAN (depending on their registration status). The second SSID will allow encrypted communications for registered users. The VLANs here are the “normal” VLAN and the isolation VLAN (if ever there are open violations for the MAC).

In the following example, we configure a Cisco 1242 AP (IP address 192.168.0.4). Configuration of other vendor's APs is similar.

First define the normal, isolation, registration and visitor VLANs on the AP, together with the appropriate wired and wireless interfaces as shown below for the isolation VLAN

```
dot11 vlan-name isolation vlan 2

interface FastEthernet0.2

  encapsulation dot1Q 2

  no ip route-cache

  bridge-group 253

  no bridge-group 253 source-learning

  bridge-group 253 spanning-disabled

interface Dot11Radio0.2

  encapsulation dot1Q 2

  no ip route-cache
  bridge-group 253

  bridge-group 253 subscriber-loop-control

  bridge-group 253 block-unknown-source

  no bridge-group 253 source-learning

  no bridge-group 253 unicast-flooding

  bridge-group 253 spanning-disabled
```

Then create the two SSIDs

```
dot11 ssid WPA2

  vlan 2 backup normal

  authentication open eap eap_methods

  authentication key-management wpa

  accounting acct-methods

  mbssid guest-mode

dot11 ssid MACauth

  vlan 3 backup visitor

  authentication open mac-address mac_methods

  accounting acct_methods

  mbssid guest-mode
```

Configure the RADIUS server (we assume here that the FreeRADIUS server and the PacketFence server are located on the same box)

```
radius-server host 192.168.0.10 auth-port 1812 acct-port 1813 key secretKey

aaa group server radius rad_eap

  server 192.168.0.10 auth-port 1812 acct-port 1813

aaa authentication login eap_methods group rad_eap

aaa group server radius rad_mac
```

```
  server 192.168.0.10 auth-port 1812 acct-port 1813

aaa authentication login mac_methods group rad_mac
```

Enable the SNMP deauthentication traps

```
snmp-server enable traps deauthenticate

snmp-server host 192.168.0.10 public deauthenticate
```

And finally activate the SSIDs on the radio

```
interface Dot11Radio0

  encryption vlan 1 mode ciphers aes-ccm

  encryption vlan 2 mode ciphers aes-ccm

  ssid WPA2

  ssid MACauth
```

Now check with a Wi-Fi card that you can actually see the two new SSIDs. You can't connect to them yet since the RADIUS server is not up and running.

Start configuring the FreeRADIUS server by adding the following lines at the end of `/etc/raddb/clients.conf`

```
client 192.168.0.3 {

  secret = secretKey

  shortname = AP1242

}
```

In `/etc/raddb/eap.conf` set the default eap type to peap (at the beginning of the eap {} section)

```
              default_eap_type = peap
```

and setup your cryptographic keys in the tls {} section.

Then update `/etc/raddb/radiusd.conf` by first adding the following lines to the modules {} section

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

        }
```

and then add “perl” at the end of the authorize {} section. The script `/etc/raddb/rlm_perl_packetfence.pl` 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.

The last file to edit is `/etc/raddb/users` where we define that non EAP-messages should, by default, lead to an authentication acceptance

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

and then, we add our local test user with

```
testUser User-Password == "testPwd"
```

Now start FreeRADIUS in debug mode

```
radiusd -x
```

Try to connect to one of the two new SSIDs with your Wi-Fi card and you'll see the packets received by FreeRADIUS, and the generated responses.

Performance optimization

Logging

Syslog reporting causes a huge slowdown on large networks. Once you've worked out all the kinks in your deployment, be sure to reduce the logging level to the default of verbosity=4. We don't recommend reducing it any further - you will lose valuable information and the bulk of the extraneous logging is at levels greater than 4.

Snort signatures

PacketFence uses the Bleeding Snort ruleset as the basis of it's Snort-based detection. The ruleset consists of several hundred rules, most of which you may decide you do not need. We recommend using a tool such as Oinkmaster to keep your rules slim and trim. I sample oinkmaster.conf file is included with PacketFence.

MySQL tuning

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 you 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 heavly 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...

service|command

httpd|stop

pfmon|stop

# /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=200M

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=100M
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...
Checking configuration sanity...
service|command
config files|start
iptables|start
httpd|start
pfmon|start
```

Wait 10 minutes for PacketFence to initial the network map and re-check iostart 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
```

High availability

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

Database

- A local MySQL database server on each PacketFence box and replication of the database partition using drbd
    - You have to make sure that only one database server is running at each time (don't double-mount the partition)
    - You have to authorize the necessary ports for DRBD using conf/iptables.pre
- A remote MySQL database server with its own high availability setup

Service startup using Heartbeat

Heartbeat should start the MySQL and PacketFence services. Heartbeat should mount a common, shared IP address on the master node and this IP address should be configured as the DNS entry for <your PacketFence hostname.your organisation.your TLD>. It should also be used as the SNMP trap server address in your switch configurations.

Chapter 10

Frequently Asked Questions

Services

How to restart PacketFence ?

```
service packetfence restart
```

How can I check that PacketFence is active ?

```
service packetfence status
```

Web interface

Where is the admin interface ?

The Web Administration GUI is available using https. It's 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

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.
- uplink: list of ports that are NOT managed by PacketFence. All uplink ports must be defined here!
  - Dynamic (defaut): 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. Could be for example:
  - Cisco::Catalyst_2900XL
  - Cisco::Catalyst_2950
  - Cisco::Catalyst_2960
  - Cisco::Catalyst_2970
  - Cisco::Catalyst_3500XL
  - Cisco::Catalyst_3550
  - Cisco::Catalyst_3560

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 !

Chapter 11

Additional Software

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.

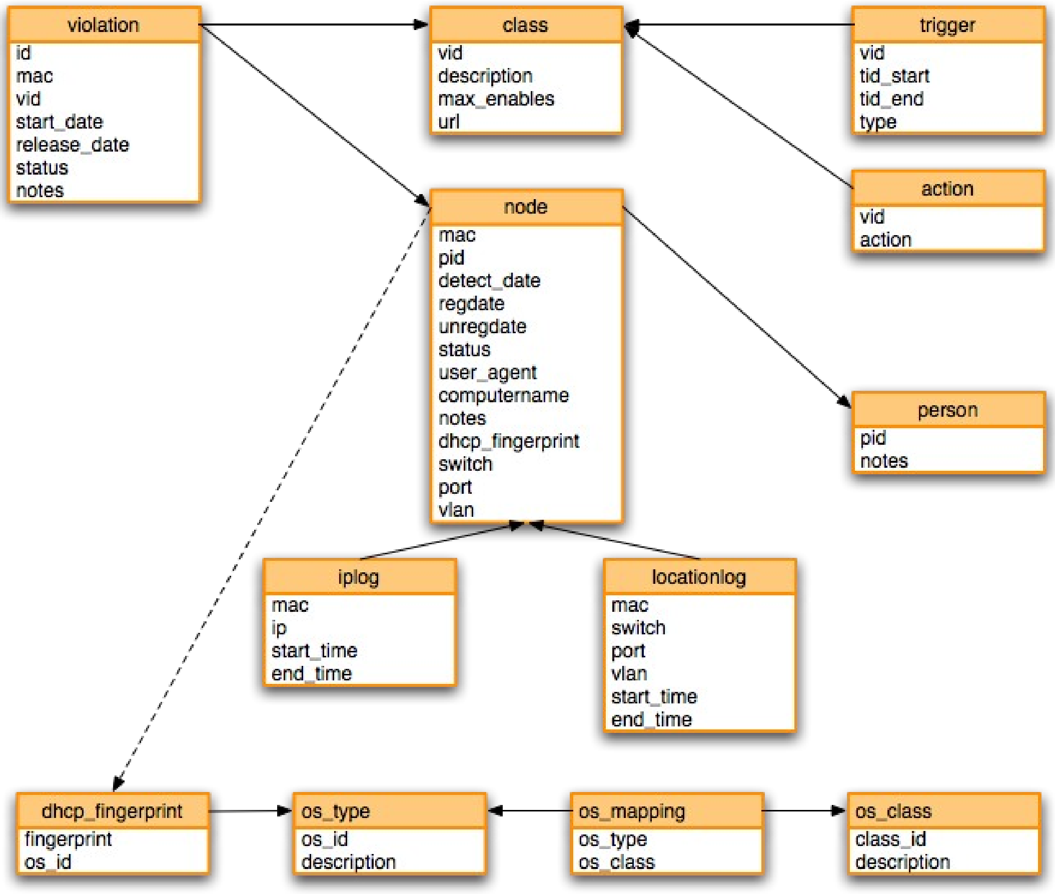
Please visit http://www.snort.org/dl/ to download and install Snort.

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 12

Appendix A: Database Schema



Chapter 13

Appendix B: Configuration parameter reference

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)

Chapter 14

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

Chapter 15

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 16

GNU Free Documentation License

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