PacketFence – version 1.7.5

Installation Guide

Version 1.7.5 – December 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 online at http://inverse.ca/uploads/docs/PacketFence_Installation_Guide.pdf.

Chapter 2

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 Download

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/local.pm` and change the following parameter :

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

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

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

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.

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 the `[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

Please consult the Administration Guide for the complete list of supported switches 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 the Administration Guide 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

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 8

GNU Free Documentation License

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