



# PacketFence Upgrade Guide

for PacketFence version 10.1.0

---

# PacketFence Upgrade Guide

Version 10.1.0 - June 2020  
Copyright © 2020 Inverse inc.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

The fonts used in this guide are licensed under the SIL Open Font License, Version 1.1. This license is available with a FAQ at: <http://scripts.sil.org/OFL>

Copyright © Łukasz Dziejdzic, <http://www.latofonts.com>, with Reserved Font Name: "Lato".

Copyright © Raph Levien, <http://levien.com/>, with Reserved Font Name: "Inconsolata".

9279Vnj

# Table of Contents

About this Guide .....	1
Other sources of information .....	1
General Upgrade Tips .....	2
Prerequisites .....	2
Database backup .....	2
PacketFence configurations and codebase backup .....	2
Upgrade procedure .....	3
Steps .....	3
Stop all PacketFence services .....	4
Packages upgrades .....	4
Restart all PacketFence services .....	6
Upgrading from a version prior to 7.0.0 .....	7
Debian upgrade .....	7
MariaDB upgrade (CentOS + RHEL only) .....	8
Database schema update (all Linux distributions) .....	8
Systemd integration .....	9
Admin roles configuration .....	9
PacketFence configuration .....	9
Maintenance configuration .....	9
DHCP filters configuration .....	10
Roles configuration .....	10
pfdetect configuration .....	10
LinkedIn Source changes .....	11
Logging service .....	11
Redis Queue .....	11
SSL certificates .....	11
Running 7.0+ in a cluster .....	12
Upgrading from a version prior to 7.1.0 .....	15
Multiple DNS servers per domain .....	15
Add default values to new auth source parameters .....	15
Fix the Ubiquiti typo .....	15
Instagram source changes .....	15
Database schema update (all Linux distributions) .....	16
Upgrading from a version prior to 7.2.0 .....	17
Ability to «pin» a domain DC .....	17
Change to sponsor CC address .....	17
Changes to authentication sources codebase .....	17
Database schema update (all Linux distributions) .....	18
Upgrading from a version prior to 7.3.0 .....	19
Device Registration .....	19
Changes to <b>authentication.conf</b> and <b>domain.conf</b> regarding realms and source matching .....	19
MariaDB database read-only mode .....	19
Database schema update (all Linux distributions) .....	20
Upgrading from a version prior to 7.4.0 .....	21
New LinkedIn domain list .....	21
Portal redirection timer .....	21
Database schema update (all Linux distributions) .....	21
Upgrading from a version prior to 8.0.0 .....	22
Realms upgrade .....	22
Fingerbank v2 .....	22

Changes to the default switch roles .....	23
Removal of the graphite database .....	23
Changes to DNS filters .....	24
Database schema update (all Linux distributions) .....	24
Upgrading from a version prior to 8.1.0 .....	25
Changes on unreg_on_accounting_stop parameter .....	25
Database schema update (all Linux distributions) .....	25
Upgrading from a version prior to 8.2.0 .....	26
Queue Stats maintenance job removal .....	26
Upgrade pfdetect perl regex to the go RE2 regex .....	26
Upgrade realm.conf to be tenant aware .....	26
The api_user table has been deprecated .....	27
Upgrade pf user privileges .....	27
Update connection_type from WIRED_MAC_AUTH to Ethernet-NoEAP .....	27
Database schema .....	27
Upgrading from a version prior to 8.3.0 .....	29
Upgrade pf.conf to rename configuration parameters .....	29
Upgrade authentication.conf to add searchattributes parameter .....	29
Adjustment to the encoding of the configuration files and templates .....	29
Database schema .....	30
Upgrading from a version prior to 9.0.0 .....	31
Support for Debian 8 dropped .....	31
Necessity to use MariaDB .....	31
Deprecate the classic dhcp filters .....	31
Violations have been renamed to Security Events .....	32
Removed MAC detection setting .....	32
Modifications to accounting cleanup .....	32
Admin roles configuration .....	32
Database schema .....	33
Upgrading from a version prior to 9.1.0 .....	34
Now possible to disable a domain .....	34
pfperl-api port .....	34
Linkedin OAuth2 .....	34
VLAN pool configuration .....	34
Remove Useragent Triggers .....	35
Self service portal .....	35
Password of the day rotation .....	35
Database schema .....	36
Upgrading from a version prior to 9.2.0 .....	37
Merge of all RPM packages into one (RHEL / CentOS only) .....	37
New GPG key for Debian installations (Debian only) .....	38
Database schema .....	38
Upgrading from a version prior to 9.3.0 .....	39
Execute script action doesn't use sudo anymore .....	39
Database schema .....	39
Upgrading from a version prior to 10.0.0 .....	40
Kernel development package .....	40
Timezone .....	41
Tracking configuration service enabled by default .....	41
New PacketFence PKI in Golang .....	41
New MariaDB Galera recovery service .....	42
Removal of currently-at file and configurator display .....	42
Database Privileges .....	42
Filter Engine .....	42

Database schema .....	43
Upgrading from a version prior to 10.1.0 .....	44
RADIUS attributes in authentication sources .....	44
Changes in RADIUS configuration for better LDAP support .....	44
RADIUS filter templates .....	44
New EAP configuration parameter in realm.conf file .....	45
Status of rules .....	45
Support for CoA in Unifi controllers .....	45
Database schema .....	45
Upgrading from a version prior to X.Y.Z .....	46
Database schema .....	46
Archived upgrade notes .....	47
Upgrading from a version prior to 4.0.0 .....	47
Upgrading from a version prior to 4.0.1 .....	48
Upgrading from a version prior to 4.0.3 .....	48
Upgrading from a version prior to 4.0.4 .....	49
Upgrading from a version prior to 4.0.5 .....	49
Upgrading from a version prior to 4.0.6 .....	49
Upgrading from a version prior to 4.1.0 .....	49
Upgrading from a version prior to 4.2.0 .....	50
Upgrading from a version prior to 4.3.0 .....	51
Upgrading from a version prior to 4.4.0 .....	51
Upgrading from a version prior to 4.5.0 .....	52
Upgrading from a version prior to 4.6.0 .....	52
Upgrading from a version prior to 4.7.0 .....	53
Upgrading from a version prior to 5.0.0 .....	53
Upgrading from a version prior to 5.1.0 .....	55
Upgrading from a version prior to 5.2.0 .....	56
Upgrading from a version prior to 5.3.0 .....	56
Upgrading from a version prior to 5.4.0 .....	57
Upgrading from a version prior to 5.5.0 .....	59
Upgrading from a version prior to 5.6.0 .....	61
Upgrading from a version prior to 5.7.0 .....	62
Upgrading from a version prior to 6.0.0 .....	63
Upgrading from a version prior to 6.1.0 .....	66
Upgrading from a version prior to 6.2.0 .....	66
Upgrading from a version prior to 6.2.1 .....	67
Upgrading from a version prior to 6.3.0 .....	67
Upgrading from a version prior to 6.4.0 .....	69
Upgrading from a version prior to 6.5.0 .....	70
Additional Information .....	72
Commercial Support and Contact Information .....	73
GNU Free Documentation License .....	74

# About this Guide

---

This guide covers procedures to upgrade PacketFence servers.

## Other sources of information

---

[Clustering Guide](#)

Covers installation in a clustered environment.

[Developer's Guide](#)

Covers API, captive portal customization, application code customizations and instructions for supporting new equipment.

[Installation Guide](#)

Covers installation and configuration of PacketFence.

[Network Devices Configuration Guide](#)

Covers switches, WiFi controllers and access points configuration.

**NEWS.asciidoc**

Covers noteworthy features, improvements and bug fixes by release.

These files are included in the package and release tarballs.

# General Upgrade Tips

---

## Prerequisites

---

You need to have your MariaDB root password ready because several scripts will need it.

## Database backup

---

Before making any changes to your database, ensure that you have a backup. A complete database backup can be taken using this command:

```
mysqldump --opt --routines -u root -p pf | gzip > /root/packetfence_db.sql.gz
```

If your database is more than a few hundred megabytes, you may also want to consider using a tool such as Percona XtraBackup which makes for much faster restores than mysqldump.

It is also possible to simply backup the actual MySQL files themselves – as long as the database is not currently running. To do so, stop MySQL and then copy all files under `/var/lib/mysql` to a secure directory.

## PacketFence configurations and codebase backup

---

Taking a complete backup of your current installation is strongly recommended. You can take a backup of the pf directory with the following command:

```
tar -C /usr/local -czf /root/packetfence.tar.gz pf --exclude='pf/logs' --exclude='pf/var'
```

# Upgrade procedure

---

## Steps

---

A PacketFence upgrade consists of following steps:

1. stop all PacketFence services (see [Stop all PacketFence services](#) section)
2. preliminary step(s) to run before packages upgrades (see sections for your upgrade path)
3. packages upgrades (see [Package upgrades](#) section)
4. configuration migration(s) (see sections for your upgrade path)
5. database schema upgrade(s) (see sections for your upgrade path)
6. restart all PacketFence services (see [Restart all PacketFence services](#) section)

Upgrade notes for a given upgrade path are cumulative. That is to say, if you are upgrading from version 7.0 to version 9.0 you must apply in order all:

1. preliminary step(s) to run before packages upgrades
2. configuration migration(s)
3. database schema upgrade(s)

in between the two versions.



### Important

Before starting packages upgrades, make sure to go through (read) the upgrade instructions specific to the upgrade path taken since some preliminary steps may be required to be done **BEFORE** the packages upgrades.

## Clustered environment: specific procedure

Please refer to the [PacketFence Clustering Guide](#), more specifically the [Performing an upgrade on a cluster](#) section.



## Stop all PacketFence services

---

It is recommended that you stop the currently running PacketFence services before proceeding any further. To do so, run the following commands:

```
/usr/local/pf/bin/pfcmd service pf stop
systemctl stop packetfence-config
```

## Rebooting after services have been stopped

If you reboot a standalone server or one server from a cluster after services have been stopped, make sure you set the systemd target to `multi-user.target` before rebooting:

```
systemctl set-default multi-user.target
```

This will make sure your services don't start up after the reboot.

Set it back to previous target after it boots up:

Cluster.

```
systemctl set-default packetfence-cluster.target
```

Standalone.

```
systemctl set-default packetfence.target
```

## Packages upgrades

---



### Important

Again, before starting packages upgrades, make sure to go through (read) the upgrade instructions specific to the upgrade path taken since some preliminary steps may be required to be done **BEFORE** the packages upgrades.

## RHEL / CentOS based systems

Run the following command to update PacketFence:

```
yum update --enablerepo=packetfence
```

PacketFence, Fingerbank and all their dependencies should now be upgraded. However, there may be extra steps required depending on the version you are upgrading from.

## Upgrade to RHEL / CentOS 7.7 before 9.1.0

If you run a version prior to 9.1.0 and want to **only** upgrade your RHEL / CentOS system to 7.7, you will need to perform the following preliminary steps before you perform your OS update through `yum`.

Disable `systemd-logind` service which is currently causing issues with the `systemctl isolate` command.

```
/usr/bin/systemctl stop systemd-logind
/usr/bin/systemctl --now mask systemd-logind
/usr/bin/systemctl daemon-reload
/bin/bash -c "/usr/bin/systemctl status user-0.slice | /usr/bin/grep -E -o
'#[0-9]+' | /usr/bin/sed 's/#//g' | /usr/bin/xargs -I{} /bin/bash -c '/usr/bin/
kill -0 {} > /dev/null 2>/dev/null && /usr/bin/echo {} > /sys/fs/cgroup/systemd/
tasks'"
```

These commands prevent disconnection issues while `yum` is performing the update. The last command ensures your current process is removed out of the `user-0.slice`.

Get latest version of `ipset` package from packetfence repository.

```
yum update ipset ipset-libs --enablerepo=packetfence
```

## Debian based systems

Run the following commands to update PacketFence:

```
apt update
apt install packetfence
```

Since PacketFence heavily relies on Fingerbank, it is recommended to make sure you are running the latest version:

```
apt install fingerbank
```

PacketFence and Fingerbank should now be upgraded. However, there may be extra steps required depending on the version you are upgrading from.

## New versions of configuration files

You should take care to review any changes to configuration files and merge them if required.

## RHEL / CentOS based systems

To find out which configuration files have changed run following command:

```
find /usr/local/pf -name \*.rpmnew
```

The list of files returned are the new versions shipped with PacketFence. Compare them to your existing version and see if there are changes that should be merged into your existing configuration. Then, once you are done make sure to delete these files so that there is no confusion the next time you upgrade PacketFence.

## Debian based systems

Debian installation system should have interactively asked for existing modified files. If you answer **No** to keep existing files in place, you can find out which configuration files have changed by running following command:

```
find /usr/local/pf -name "*.dpkg-dist"
```

## Restart all PacketFence services

---

Once all [steps of upgrade procedure](#) are completed, restart `packetfence-config` and `packetfence` services:

```
/usr/local/pf/bin/pfcmd fixpermissions  
/usr/local/pf/bin/pfcmd pfconfig clear_backend  
systemctl restart packetfence-config  
/usr/local/pf/bin/pfcmd configreload hard  
/usr/local/pf/bin/pfcmd service pf restart
```

# Upgrading from a version prior to 7.0.0

---



## Note

You cannot upgrade from CentOS 6 or Debian Wheezy to PacketFence 7.0 and above

## Debian upgrade

---

The requirement for MariaDB 10.1 means that a simple "apt upgrade" will not be enough. You will need to help apt through the upgrade by manually removing some packages and installing some others. The need to ensure you have backups cannot be overstated.

Make sure the apt database is up to date

```
apt update
```

Remove the MySQL 5.5 packages (do not purge them, as that would delete the database)

```
dpkg -r --force-all mysql-client-5.5 mysql-common mysql-server mysql-server-5.5  
mysql-server-core-5.5 libmysqlclient18
```

Install the newer Mariadb-10.1 packages

```
apt install libmariadbclient18 libmysqlclient18 mariadb-common mariadb-server-10.1  
galera-3 gawk mariadb-client-10.1 mariadb-server-core-10.1 rsync socat libmpfr4  
mariadb-client-core-10.1 mysql-common
```

Finally, upgrade the rest of the packages

```
apt full-upgrade
```

Note that "full-upgrade" may also affect other packages you might have installed on the system if you had other software than PacketFence on it.

## MariaDB upgrade (CentOS + RHEL only)

---

Upgrading to PacketFence 7+ will install a more recent version of MariaDB than the one that is shipped with CentOS.

In order to upgrade the MariaDB metadata files and tables, first stop any started process.

```
systemctl stop mariadb
systemctl stop packetfence-mariadb
```

Then start a `mysqld_safe` process manually (this will start a background process)

```
mkdir -p /var/run/mariadb
chown mysql: /var/run/mariadb
mysqld_safe --basedir=/usr &
```

Then, execute the upgrade script and enter the root password when prompted

```
mysql_upgrade -u root -p
```

When done, kill the `mysqld_safe` process we started before the update, reattach to it and wait for it to exit

```
kill %1 && fg
```

Note that it might take up to a few minutes for the process to exit depending on the size of your database.

Once done, restart the MariaDB service (managed by PacketFence)

```
systemctl start packetfence-mariadb
```

## Database schema update (all Linux distributions)

---

Changes have been made to the database schema. You will need to update it accordingly. An SQL upgrade script has been provided to upgrade the database from the 6.5 schema to 7.0.

To upgrade the database schema, run the following command:

```
mysql -u root -p pf -v < /usr/local/pf/db/upgrade-6.5.0-7.0.0.sql
```

## Systemd integration

---

All PacketFence services are managed individually via systemd unit files instead of one unit file (`packetfence.service`). When you updated the PacketFence package, it already set the system target to `packetfence.target`.

If you are hosting the MySQL/MariaDB service on your PacketFence servers (it is by default), you should now manage the service via `packetfence-mariadb.service` instead of `mariadb.service`. The changes in the server startup will be done automatically by the packaging.

## Admin roles configuration

---

Given the portal profiles have now been renamed to connection profiles, you need to make sure any admin role that allowed portal profile Create/Read/Update/Delete operations is updated to be valid for connection profiles.

```
cd /usr/local/pf
sed -i "s/PORTAL_PROFILE/CONNECTION_PROFILE/g" conf/adminroles.conf
```

## PacketFence configuration

---

Multiple parameters inside `pf.conf` have been renamed for better clarity. Execute the following in order to migrate the parameters.

```
/usr/local/pf/addons/upgrade/to-7.0-pf-conf-changes.pl
```

## Maintenance configuration

---

Maintenance related configuration for `pfmon` has been moved to a dedicated configuration file (`/usr/local/pf/conf/pfmon.conf`).

In order to migrate your settings from `pf.conf` to `pfmon.conf`, run the following script:

```
/usr/local/pf/addons/upgrade/to-7.0-pf.conf-to-pfmon.conf.pl
```

## DHCP filters configuration

---

Minor changes were made to the DHCP filters configuration (`/usr/local/pf/conf/dhcp_filters.conf`).

First, the `computer_name` attribute was renamed to `computername` to be consistent with the rest of the application. Then, the `DhcpFingerbank` scope was changed to `Fingerbank`

In order to rename those in an automated way:

```
cd /usr/local/pf
sed -i "s/computer_name/computername/g" conf/dhcp_filters.conf
sed -i "s/DhcpFingerbank/Fingerbank/g" conf/dhcp_filters.conf
```

## Roles configuration

---

The source of truth for roles is now in a configuration file (`/usr/local/pf/conf/roles.conf`) instead of being in the database. In order to pull the existing roles from your database into the configuration file, execute the following command:

```
/usr/local/pf/addons/upgrade/to-7.0-roles-conf.pl
```



### Note

The roles still exist in the database like before (`node_category` table), but their source of truth is now in the configuration file. Should you remove a role manually from `roles.conf`, it will **not** be removed from the database unless you manually go delete it from the database.

## pfdetect configuration

---

New parameters have been introduced in `conf/pfdetect.conf`. Run the following script to migrate your configuration.

```
/usr/local/pf/addons/upgrade/to-7.0-pfdetect-conf.pl
```

## LinkedIn Source changes

---

If you are using the LinkedIn OAuth2 source, a change has been made on their API, thus you will need to do the following:

```
cd /usr/local/pf
sed -i "s/uas\|oauth2/oauth\|v2/g" conf/authentication.conf
```

## Logging service

---

Since all logging now goes through rsyslog, if you had edited the logging configuration (e.g. to forward logs to a centralized syslog server) make sure that the new logging rules in `/etc/rsyslog.d/packetfence.conf` do not conflict with your changes.

Take a look at `/usr/local/pf/conf/log.conf` and `/usr/local/pf/conf/log.conf.d/*` for the detailed configuration of the PacketFence services.

## Redis Queue

---

Clear the redis queue to avoid old stale jobs from being processes.

```
systemctl start packetfence-redis_queue
redis-cli -p 6380 FLUSHALL
systemctl stop packetfence-redis_queue
```

## SSL certificates

---

Given that haproxy is now the termination point for the captive portal, any SSL configuration you have in `/usr/local/pf/conf/httpd.conf.d/ssl-certificates.conf` needs to be ported so that it works with haproxy.

Easiest solution is to bundle your server cert, your intermediates (if any) along with the key in the default file used by the PacketFence haproxy process (`/usr/local/pf/conf/ssl/server.pem`)

In order to do so:



```
# cd /usr/local/pf/
# cat /path/to/your/server.crt /path/to/your/intermediates.crt /path/to/your/
server.key > /usr/local/pf/conf/ssl/server.pem
```

## Running 7.0+ in a cluster

---

A complete re-visit of the database clustering stack was done in version 7.0. If you run your PacketFence installation in a cluster, make sure you read the following section.

### Active/Active clusters with Active/Passive DB (default before 7.0)

We highly suggest you migrate your existing clustered installation using Corosync/Pacemaker to the new cluster stack of PacketFence that uses MariaDB Galera cluster. The easiest way to perform this is to build new servers and port your configuration (by copying the configuration files) and your database (using mysqldump). There are ways to migrate the 2 existing nodes to a 3 nodes cluster but this is not covered in this guide.

#### Corosync adjustment

Note that you can safely keep your existing 2-node cluster with Corosync/Pacemaker in place and things will work like before. You will simply have to adjust your Corosync configuration so that MariaDB points to the packetfence-mariadb file instead of the mariadb unit.

```
primitive MariaDB systemd:packetfence-mariadb \
    op start timeout=60s interval=0 \
    op stop timeout=60s interval=0 \
    op monitor interval=20s timeout=30s
```

#### Disabling Galera cluster

You must then disable the MariaDB Galera cluster as a replication mechanism as you will still be using DRBD. In order to do so, add the following in `/usr/local/pf/conf/pf.conf`

```
[active_active]
galera_replication=disabled
```

#### IP address bind

You must also instruct packetfence-mariadb to bind to the management IP address of the server manually.

In order to do so, replace the following section in `/usr/local/pf/conf/mariadb/mariadb.conf.tt`:

```
[% IF server_ip.length %]
bind-address=[% server_ip %]
[% ELSE %]
skip-networking
bind-address=
[% END %]
```

with: bind-address=1.2.3.4

Where 1.2.3.4 is the management IP address of the server.

## Disable packetfence-mariadb on boot

Like in previous versions where mariadb shouldn't have been started on boot, now you must ensure its replacement (packetfence-mariadb) doesn't start on boot.

```
systemctl disable packetfence-mariadb
```

### Enabling the packetfence-cluster target

Next, you must set the default target to packetfence-cluster:

```
systemctl set-default packetfence-cluster.target
```

## Active/Active clusters with external DB

No changes to your clustering stack is required when using an external database.

## Active/Passive clusters



### Caution

You shouldn't be running active/passive clusters anymore. If you do, you're pretty much on your own for community support. Inverse provides professional services to help you maintain these clusters. If you intend to keep an active/passive cluster, we suggest you have deep knowledge of Corosync/Pacemaker and strong Linux skills.

First, no changes are required to your database stack as MariaDB supports being deployed in Active/Passive.

You will need to adjust the Corosync/Pacemaker configuration to take in consideration the changes made to systemd for PacketFence services. Before 7.0, PacketFence used to be controlled via a single systemd unit file while now it uses a multiple services grouped in targets. In order to mimic the single service behavior that was in previous versions, a unit file is provided here: <https://github.com/inverse-inc/packetfence/blob/devel/packetfence-active-passive.service>. You should install this file in `/etc/systemd/system/packetfence.service` and make sure there are no other leftovers of `packetfence.service` unit files on your system.

Then, you must adjust the systemd default target so PacketFence doesn't start on boot and note that this should be done on every future upgrade of your system.

```
# systemctl set-default multi-user.target
```

You should then change your Corosync configuration for MariaDB and PacketFence to the following:

```
primitive MariaDB systemd:packetfence-mariadb \  
    op start timeout=60s interval=0 \  
    op stop timeout=60s interval=0 \  
    op monitor interval=20s timeout=30s  
primitive PacketFence systemd:packetfence \  
    op start timeout=300s interval=0 \  
    op stop timeout=300s interval=0 \  
    op monitor interval=300s timeout=300s
```

# Upgrading from a version prior to 7.1.0

---

## Multiple DNS servers per domain

---

The PacketFence Active Directory Domains integration now supports multiple DNS servers to be specified to find a DC. For this reason the parameter `dns_server` has been renamed to `dns_servers` in `domain.conf`. In order to automatically rename the parameters, run the following command:

```
sed -i.bak "s/^dns_server/dns_servers/g" /usr/local/pf/conf/domain.conf
```

## Add default values to new auth source parameters

---

```
/usr/local/pf/addons/upgrade/to-7.1-authentication-conf.pl
```

## Fix the Ubiquiti typo

---

In order to use the Ubiquiti switch module that has been renamed, run the following command:

```
sed -i.bak "s/Ubiquity/Ubiquiti/g" /usr/local/pf/conf/switches.conf
```

## Instagram source changes

---

Due to a change in the API of Instagram please change the scope if you are using an Instagram OAuth2 source. Replace `scope=email` by `scope=basic` in `conf/authentication.conf` under the section `[Instagram Source]`.

## Database schema update (all Linux distributions)

---

Changes have been made to the database schema. You will need to update it accordingly. An SQL upgrade script has been provided to upgrade the database from the 7.0 schema to 7.1.

To upgrade the database schema, run the following command:

```
mysql -u root -p pf -v < /usr/local/pf/db/upgrade-7.0.0-7.1.0.sql
```

# Upgrading from a version prior to 7.2.0

---

## Ability to «pin» a domain DC

---

PacketFence is now able to instruct Samba to «pin» a DC for authentication or use all of them. You should instruct Samba to connect to all domain controllers by adding the following to each of your domains in domain.conf:

```
sticky_dc=*
```

And then regenerate the domain configuration:

```
/usr/local/pf/bin/pfcmd fixpermissions  
/usr/local/pf/bin/pfcmd configreload hard  
/usr/local/pf/bin/pfcmd generatedomainconfig
```

## Change to sponsor CC address

---

The CC address for sponsors is now BCC. In order to adjust the configuration, execute the following:

```
cd /usr/local/pf  
sed -i "s/sponsorship_cc/sponsorship_bcc/g" conf/authentication.conf
```

## Changes to authentication sources codebase

---

Any custom authentication sources forms and templates would need to be copied to the new location.

Templates `/usr/local/pf/html/pfappserver/root/authentication/source/type/` → `/usr/local/pf/html/pfappserver/root/config/source/type/`

Forms `/usr/local/pf/html/pfappserver/lib/pfappserver/Form/Config/Authentication/Source` → `/usr/local/pf/html/pfappserver/lib/pfappserver/Form/Config/Source`

## Database schema update (all Linux distributions)

---

Changes have been made to the database schema. You will need to update it accordingly. An SQL upgrade script has been provided to upgrade the database from the 7.1 schema to 7.2.

To upgrade the database schema, run the following command:

```
mysql -u root -p pf -v < /usr/local/pf/db/upgrade-7.1.0-7.2.0.sql
```

# Upgrading from a version prior to 7.3.0

---

## Device Registration

---

You will need to remove anything related to [device\_registration] in the conf/pf.conf file. Once done, you will need to reconfigure any device registration policy using the following instructions: [https://packetfence.org/doc/PacketFence\\_Installation\\_Guide.html#\\_devices\\_registration](https://packetfence.org/doc/PacketFence_Installation_Guide.html#_devices_registration)

## Changes to authentication.conf and domain.conf regarding realms and source matching

---

You have to run the following script to change the configuration:

```
/usr/local/pf/addons/upgrade/to-7.3-authentication-conf.pl
```

## MariaDB database read-only mode

---

There was, in some cases, an issue where the database cluster was put in a read-only mode which then prevent it to comes back gracefully.

A modification have been made to now use the wsrep\_ready state of the DB as a read only indicator. Therefore, PacketFence will stop putting the DB in read only on quorum + primary loss of MariaDB and trust wsrep\_ready instead

Ensure you merge changes in the galera section of `conf/mariadb/mariadb.conf.tt.rpmnew` into `conf/mariadb/mariadb.conf.tt`

Once completed, update the file `/usr/local/pf/conf/currently-at` to match the new release number (PacketFence 7.3.0).



## Database schema update (all Linux distributions)

---

Changes have been made to the database schema. You will need to update it accordingly. An SQL upgrade script has been provided to upgrade the database from the 7.2 schema to 7.3.

To upgrade the database schema, run the following command:

```
mysql -u root -p pf -v < /usr/local/pf/db/upgrade-7.2.0-7.3.0.sql
```

# Upgrading from a version prior to 7.4.0

---

## New LinkedIn domain list

---

If you use social login with LinkedIn OAuth2, you will need to adjust the list of domains that are passthroughs in the LinkedIn source.

For all your LinkedIn sources, change the domains to:

```
www.linkedin.com,api.linkedin.com,*.licdn.comlatform.linkedin.com
```

## Portal redirection timer

---

The redirection timer configuration (length of the timer bar at the end of the portal) has been moved from the fencing section to the captive\_portal section. More precisely, it has moved from `fencing.redirtimer` to `captive_portal.network_redirect_delay`.

## Database schema update (all Linux distributions)

---

Changes have been made to the database schema. You will need to update it accordingly. An SQL upgrade script has been provided to upgrade the database from the 7.3 schema to 7.4.

To upgrade the database schema, run the following command:

```
mysql -u root -p pf -v < /usr/local/pf/db/upgrade-7.3.0-7.4.0.sql
```

Once completed, update the file `/usr/local/pf/conf/currently-at` to match the new release number (PacketFence 7.4.0).

# Upgrading from a version prior to 8.0.0

---

## Realms upgrade

---

The way PacketFence detects whether or not it should strip the realm out of a username when performing authentication and authorisation has been moved to the realms. Moreover, it is now configurable based on the context (login on the captive portal or administration interface, as well as when performing authorization in RADIUS 802.1x)

In order to migrate your configuration, you can use the following script which will guide you through the migration:

```
/usr/local/pf/addons/upgrade/to-8.0-authentication-conf.pl
```

## Fingerbank v2

---

### Device names

Packetfence now uses Fingerbank v2 for improved device profiling. Since this new version brings new device names, a rename of the previous data is necessary.

In order to do so, execute the following script:

```
/usr/local/pf/addons/upgrade/to-8.0-fingerbank-db-data.pl
```

### Necessary API key

Fingerbank doesn't release its signature database anymore and has moved to an API centric approach for device profiling. In order for device profiling to keep working on your installation, you must make sure that there is an API key configured for Fingerbank in your PacketFence installation.

In order to do so, you should make sure you have the following in `/usr/local/fingerbank/conf/fingerbank.conf`



### Note

In order to request an API key, you can visit the following URL: <https://api.fingerbank.org/users/register>

```
```
```

```
api_key=YOUR_API_KEY_GOES_HERE ```
```



### Warning

Fingerbank v1 and v2 **do not** use the same infrastructure. The accounts (API keys) created on `fingerbank.inverse.ca` before the 8.0 release have been migrated to `api.fingerbank.org`. Still, you should make sure that you have the correct API key configured in `fingerbank.conf` by looking at your profile on <https://api.fingerbank.org/users/register>. If you have a corporate account, then you can safely assume its been migrated, you can email [fingerbank@inverse.ca](mailto:fingerbank@inverse.ca) for a confirmation. If you use a Github account and you have tried Fingerbank v2 prior to the PacketFence 8.0 release, **then your API key will be different**. Make sure you update `fingerbank.conf` in that case.

If you manage a large scale environment, you'll want to make sure your account can perform an unlimited amount of API requests on Fingerbank so that device profiling works correctly in a consistent way. In order to obtain this, contact [fingerbank@inverse.ca](mailto:fingerbank@inverse.ca). Note that most Inverse customers are entitled to free unlimited usage of the Fingerbank Cloud API.

## Changes to the default switch roles

---

The default roles that were returned using "Role by Switch Role" have been removed. If you were relying on them to be returned in the RADIUS response, then you need to add them back in the default switch in the *Roles* tab.

The previous values were:

- `registration: registration`
- `isolation: isolation`
- `macDetection: macDetection`
- `inline: inline`
- `voice: voice`

This is should only be necessary if you are using ACL assignment on your switches and using the default names that were there in PacketFence before.

## Removal of the graphite database

---

PacketFence doesn't use graphite anymore for its dashboard. It is recommended to delete the graphite database although this is purely optional.

In order to do so, execute the following:

```
mysql -u root -p -e "drop database pf_graphite"
```

## Changes to DNS filters

---

The `$qname` parameter need to be removed from `dns_filters.conf`

In order to do so, execute the following command:

```
sed -i -e 's/\$qname//g' /usr/local/pf/conf/dns_filters.conf
```

## Database schema update (all Linux distributions)

---

Changes have been made to the database schema. You will need to update it accordingly. An SQL upgrade script has been provided to upgrade the database from the 7.4 schema to 8.0.

To upgrade the database schema, run the following command:

```
mysql -u root -p pf -v < /usr/local/pf/db/upgrade-7.4.0-8.0.0.sql
```

Once completed, update the file `/usr/local/pf/conf/currently-at` to match the new release number (PacketFence 8.0.0).

# Upgrading from a version prior to 8.1.0

---

## Changes on unreg\_on\_accounting\_stop parameter

---

The global configuration parameter `unreg_on_acct_stop` has been moved in the connection profile. So if you enabled it then make sure to enable it now in the connection profile.

## Database schema update (all Linux distributions)

---

Changes have been made to the database schema. You will need to update it accordingly. An SQL upgrade script has been provided to upgrade the database from the 7.4 schema to 8.0.

To upgrade the database schema, run the following command:

```
mysql -u root -p pf -v < /usr/local/pf/db/upgrade-8.0.0-8.1.0.sql
```

Once completed, update the file `/usr/local/pf/conf/currently-at` to match the new release number (PacketFence 8.1.0).

# Upgrading from a version prior to 8.2.0

---

## Queue Stats maintenance job removal

---

The `queue_stats` maintenance job has been deprecated in favor of using `pfstats`. In order to remove configuration related to this maintenance job, run:

```
/usr/local/pf/addons/upgrade/to-8.2-pfmon-conf.pl
```

## Upgrade pfdetect perl regex to the go RE2 regex

---

The `pfdetect` was moved from perl to go so all rule regexes have to be converted to the RE2 regex syntax. RE2 is mostly compatible with the perl regex syntax. More information on the RE2 syntax can be found here <https://github.com/google/re2/wiki/Syntax>. To upgrade the regex run:

```
/usr/local/pf/addons/upgrade/to-8.2-pfdetect-conf.pl
```

Any perl regex that cannot be converted will be displayed and should be fixed.

## Upgrade realm.conf to be tenant aware

---

The realms are now multi-tenant aware, in order to upgrade your configuration to have the existing realms use the default tenant, execute the following script:

```
/usr/local/pf/addons/upgrade/to-8.2-realm-conf.pl
```

## The `api_user` table has been deprecated

---

Any users in that were in the `api_user` table should be migrated to PacketFence local account (password table)

## Upgrade pf user privileges

---

Starting from 8.2, stored routines will be dump **with** the PacketFence database. The user created at the installation (`pf` by default) in database need to have additional privileges to do that task.

To upgrade the privileges of that user, run the following command:

```
/usr/local/pf/addons/upgrade/to-8.2-upgrade-pf-privileges.sh
```

## Update `connection_type` from `WIRED_MAC_AUTH` to `Ethernet-NoEAP`

---

We merged the `WIRED_MAC_AUTH` and `Ethernet-NoEAP` to `Ethernet-NoEAP` so the configuration needs to be updated, to do that run:

```
sed -i "s/WIRED_MAC_AUTH/Ethernet-NoEAP/g" /usr/local/pf/conf/profiles.conf /usr/local/pf/conf/vlan_filters.conf /usr/local/pf/conf/radius_filters.conf /usr/local/pf/conf/switch_filters.conf /usr/local/pf/conf/authentication.conf
```

## Database schema

---

Changes have been made to the database schema. You will need to update it accordingly. An SQL upgrade script has been provided to upgrade the database from the 8.1 schema to 8.2.

To upgrade the database schema, run the following command:

```
mysql -u root -p pf -v < /usr/local/pf/db/upgrade-8.1.0-8.2.0.sql
```



Once completed, update the file `/usr/local/pf/conf/currently-at` to match the new release number (PacketFence 8.2.0).

# Upgrading from a version prior to 8.3.0

---

## Upgrade pf.conf to rename configuration parameters

---

We moved `radius_authentication_methods` section to `radius_configuration` and moved all the radius configuration parameters in this new section. To upgrade your configuration execute the following script:

```
/usr/local/pf/addons/upgrade/to-8.3-rename-pf-conf-parameters.pl
```

## Upgrade authentication.conf to add searchattributes parameter

---

We added a new parameter in AD and LDAP authentication sources to be able to do 802.1x authentication with any unique ldap attributes. This parameter "searchattributes" need to be added in the existing authentication sources. To apply this configuration execute the following script:

```
/usr/local/pf/addons/upgrade/to-8.3-authentication-searchattributes.pl
```

## Adjustment to the encoding of the configuration files and templates

---

Configuration and templates in the admin were previously being saved as latin1 instead of utf8.

This script will convert all latin1 config file to utf8

```
/usr/local/pf/addons/upgrade/to-8.3-conf-latin1-to-utf8.sh
```

## Database schema

---

Changes have been made to the database schema. You will need to update it accordingly. An SQL upgrade script has been provided to upgrade the database from the 8.2 schema to 8.3.

To upgrade the database schema, run the following command:

```
mysql -u root -p pf -v < /usr/local/pf/db/upgrade-8.2.0-8.3.0.sql
```

Once completed, update the file `/usr/local/pf/conf/currently-at` to match the new release number (PacketFence 8.3.0).

# Upgrading from a version prior to 9.0.0

---

## Support for Debian 8 dropped

---

Debian 8 will not be supported anymore for versions 9.0.0 and above. You should instead use Debian 9 now as it is currently the only supported Debian version.

## Necessity to use MariaDB

---



### Note

This only applies to users using an external database server. If your database is hosted on the same server as PacketFence whether you are in cluster or standalone, this requires no attention.

Users hosting an external database for PacketFence will need to run a recent version of MariaDB as it will be the only supported database backend. Failure to use MariaDB may result in errors in the database migration script.

In order to migrate to MariaDB, it is suggested to create a new database server and perform an export of the data through `mysqldump` and import it in the new server.

The recommended MariaDB version for PacketFence is currently 10.1.21

A recent version of MySQL can also work but going forward, the only tested database engine will be MariaDB.

## Deprecate the classic dhcp filters

---

The previous dhcp filters engine has been deprecated in favor of the new one who is able to modify the dhcp answer on the fly.

## Violations have been renamed to Security Events

---

The violations have been renamed to security events. In order to make the appropriate changes in your configuration, execute the following script:

```
/usr/local/pf/addons/upgrade/to-9.0-security-events.sh
```

## Removed MAC detection setting

---

The MAC detection setting in the switches has been removed. In order to cleanup the switches configuration for the removal of this setting, execute the following script:

```
/usr/local/pf/addons/upgrade/to-9.0-remove_mac_detection.sh
```

## Modifications to accounting cleanup

---

Accounting cleanup is now done via a pfmon task (acct\_cleanup) instead of the database backup and maintenance script. Make sure you adjust the cleanup window in pfmon's configuration (Configuration→System Maintenance→Maintenance) if necessary. Also note that the default retention for the accounting data has been lowered to 1 day instead of 1 week like it was before.

## Admin roles configuration

---

In order to upgrade the Admin rights, run the following commands

```
cd /usr/local/pf
sed -i "s/SERVICES/SERVICES_READ/g" /usr/local/pf/conf/adminroles.conf
sed -i "s/REPORTS/REPORTS_READ/g" /usr/local/pf/conf/adminroles.conf
```

## Database schema

---

Changes have been made to the database schema. You will need to update it accordingly. An SQL upgrade script has been provided to upgrade the database from the 8.3 schema to 9.0.

To upgrade the database schema, run the following command:

```
mysql -u root -p pf -v < /usr/local/pf/db/upgrade-8.3.0-9.0.0.sql
```

Once completed, update the file `/usr/local/pf/conf/currently-at` to match the new release number (PacketFence 9.0.0).

# Upgrading from a version prior to 9.1.0

---

## Now possible to disable a domain

---

In order to add the necessary enabled flag to your existing domains, run the following command:

```
/usr/local/pf/addons/upgrade/to-9.1-add-domain-conf.pl
```

## pfperl-api port

---

The port of the pfperl-api service has changed, in order to adjust the existing configuration, run the following command:

```
/usr/local/pf/addons/upgrade/to-9.1-update-api.conf.sh
```

## LinkedIn OAuth2

---

The LinkedIn API calls have changed drastically. On top of the new LinkedIn modules that are part of the update, you will need to change the following parameter in all your existing LinkedIn sources:

```
API URL of logged user -> https://api.linkedin.com/v2/emailAddress?  
q=members&projection=(elements*(handle~))
```

## VLAN pool configuration

---

The VLAN pool strategy configuration has been moved to the connection profiles.

In order to migrate the current setting of pf.conf into profiles.conf, you will need to run the following command:

```
/usr/local/pf/addons/upgrade/to-9.1-move-vlan-pool-technique-parameter.pl
```

## Remove Useragent Triggers

---

The useragent and user\_agent security event triggers have been deprecated. Performing HTTP User-Agent based detection is extremely inefficient given the very dynamic nature of HTTP User-Agents. You should instead be using the device trigger which leverages the device profiling performed by Fingerbank. In order to remove any existing useragent trigger, execute the following script:

```
/usr/local/pf/addons/upgrade/to-9.1-security-events-remove-useragent.pl
```

## Self service portal

---

The device registration configuration file has been removed in favor of using a configuration file for all the self service portal features (status page + device registration).

In order to migrate your configuration, run the following script:

```
/usr/local/pf/addons/upgrade/to-9.1-selfservice-conf.pl
```

## Password of the day rotation

---

Password of the day source now uses access duration values to rotate password.

In order to migrate your configuration, run the following script:

```
/usr/local/pf/addons/upgrade/to-9.1-update-potd.pl
```



## Database schema

---

Changes have been made to the database schema. You will need to update it accordingly. An SQL upgrade script has been provided to upgrade the database from the 9.0 schema to 9.1.

To upgrade the database schema, run the following command:

```
mysql -u root -p pf -v < /usr/local/pf/db/upgrade-9.0.0-9.1.0.sql
```

Once completed, update the file `/usr/local/pf/conf/currently-at` to match the new release number (PacketFence 9.1.0).

# Upgrading from a version prior to 9.2.0

---

## Merge of all RPM packages into one (RHEL / CentOS only)

---



### Note

This step needs to be done **before** packages upgrade.

Starting from now, PacketFence will be released as an unique RPM package for **x86\_64** architectures. To remove properly older RPM packages, you need to follow these steps:

1. Follow instructions mentioned in [Stop all PacketFence services](#) section and stop before starting packages upgrades
2. Uninstall old RPM without running post-uninstallation steps:

```
rpm -e --nodeps --noscripts packetfence-config

# run only if packetfence-remote-arp-sensor has been installed
rpm -e --nodeps --noscripts packetfence-remote-arp-sensor
```

3. Recopy previous [filename]pfconfig.conf filename to its original location:

```
mv -f /usr/local/pf/conf/pfconfig.conf.rpmsave /usr/local/pf/conf/pfconfig.conf
```

4. Upgrade PacketFence packages by following instructions in [Packages upgrades](#) section for RHEL / CentOS based systems
5. Continue upgrade procedure

At the end of upgrade procedure, you should have only one RPM package called [package]**packet-fence**. If you previously installed [package]**packetfence-release** package in order to have PacketFence repository installed, this one has been upgraded to latest version.

## New GPG key for Debian installations (Debian only)

---



### Note

This step needs to be done **before** packages upgrade.

In order to install new versions of Debian packages, you will need to add a new GPG key to your system:

```
wget -O - https://inverse.ca/downloads/GPG_PUBLIC_KEY | sudo apt-key add -
```

You can safely remove the oldest one:

```
sudo apt-key del FE9E84327B18FF82B0378B6719CDA6A9810273C4
```

## Database schema

---

Changes have been made to the database schema. You will need to update it accordingly. An SQL upgrade script has been provided to upgrade the database from the 9.1 schema to 9.2.

To upgrade the database schema, run the following command:

```
mysql -u root -p pf -v < /usr/local/pf/db/upgrade-9.1.0-9.2.0.sql
```

Once completed, update the file [filename]/usr/local/pf/conf/currently-at to match the new release number (PacketFence 9.2.0):

```
cat /usr/local/pf/conf/pf-release > /usr/local/pf/conf/currently-at
```

# Upgrading from a version prior to 9.3.0

---

## Execute script action doesn't use sudo anymore

---

Execute script action in security events doesn't use `[command]sudo` anymore to run scripts. Consequently, you should ensure that `pf` user is:

- able to read and execute these scripts
- able to run commands inside these scripts (without `sudo`)

## Database schema

---

Changes have been made to the database schema. You will need to update it accordingly. An SQL upgrade script has been provided to upgrade the database from the 9.2 schema to 9.3.

To upgrade the database schema, run the following command:

```
mysql -u root -p pf -v < /usr/local/pf/db/upgrade-9.2.0-9.3.0.sql
```

Once completed, update the file `[filename]/usr/local/pf/conf/currently-at` to match the new release number (PacketFence 9.3.0):

```
cat /usr/local/pf/conf/pf-release > /usr/local/pf/conf/currently-at
```

# Upgrading from a version prior to 10.0.0

---

## Kernel development package

---



### Note

This step needs to be done **before** packages upgrade.

In this version we need to have the kernel development package that matches your current kernel version in order to build the Netflow kernel module.

## RHEL / CentOS based systems

```
yum install kernel-devel-$(uname -r)
```

The headers for your specific kernel may not be published anymore in the CentOS repository. If that is the case, then perform the following prior to the upgrade:

```
yum update kernel
reboot
yum install kernel-devel-$(uname -r)
```



### Note

Be sure to follow instructions in [the section called “Rebooting after services have been stopped”](#) section to ensure services will not restart.

## Debian based systems

```
apt install linux-headers-$(uname -r)
```

## Timezone

---

The timezone set in `pf.conf` will be set on the operating system every time PacketFence reloads its configuration. For this reason, you should review the timezone setting in the general section of `pf.conf` (System Configuration → General Configuration in the admin). If its empty, PacketFence will use the timezone that is already set on the server and you don't have anything to do. Otherwise, it will set the timezone in this setting on the operating system layer for consistency which may modify the timezone setting of your operating system. In this case you should ensure that you reboot the server after completing all the steps of the upgrade so that the services start with the right timezone.

## Tracking configuration service enabled by default

---

`packetfence-tracking-config` service is now enabled by default. It means that all manual changes to configuration files will be recorded, including passwords.

You can disable this service from PacketFence web admin if you don't want such behavior.

## New PacketFence PKI in Golang

---



### Note

If you do not use the PacketFence PKI, you can safely ignore this step

PacketFence-pki is deprecated in favour of the new PacketFence PKI written in Golang. If you previously used the PacketFence-pki you will need to migrate from the SQLite database to MariaDB. To migrate, be sure that the database is running and the new PKI too and do the following:

```
/usr/local/pf/addons/upgrade/to-10.0-packetfence-pki-migrate.pl
```

Next edit the PKI providers (Configuration → PKI Providers) and redefine the profile to use. Finally, if you use OCSP then change the URL to use this one: <http://127.0.0.1:22225/api/v1/pki/ocsp>

## New MariaDB Galera recovery service

---

This release adds a new service that will automatically attempt to recover broken Galera cluster members and can also perform a full recovery of a Galera cluster. These automated decisions may lead to potential data loss. If this is not acceptable for you disable the galera-autofix service in pf.conf or in "System Configuration→Services". More details and documentation is available in the "The galera-autofix service" section of the clustering guide.

## Removal of currently-at file and configurator display

---

The file [filename]/usr/local/pf/conf/currently-at is no longer needed, it can be removed:

```
rm /usr/local/pf/conf/currently-at
```

You also need to disable access to configurator by running:

```
printf '\n[advanced]\nconfigurator=disabled\n' >> /usr/local/pf/conf/pf.conf
```

## Database Privileges

---

Some queries now need CREATE TEMPORARY TABLE privilege. You will be prompted for the MariaDB root password when running this script:

```
/usr/local/pf/addons/upgrade/to-10.0-upgrade-pf-privileges.sh
```

## Filter Engine

---

We are now using a new format for the VLAN/DNS/DHCP/RADIUS/Switch filters. This script will convert the old format to the new one:

```
/usr/local/pf/addons/upgrade/to-10.0-filter_engines.pl
```

## Database schema

---

Changes have been made to the database schema. You will need to update it accordingly. An SQL upgrade script has been provided to upgrade the database from the 9.3 schema to 10.0.

To upgrade the database schema, run the following command:

```
mysql -u root -p pf -v < /usr/local/pf/db/upgrade-9.3.0-10.0.0.sql
```



# Upgrading from a version prior to 10.1.0

---

## RADIUS attributes in authentication sources

---

RADIUS attributes used in rules of authentication sources are now prefixed by `radius_request`. This script will add the prefix:

```
/usr/local/pf/addons/upgrade/to-10.1-authentication-prefix.pl
```

## Changes in RADIUS configuration for better LDAP support

---

In order to improve LDAP support when using RADIUS, new files and configuration parameters have been added. This script will update your current configuration:

```
/usr/local/pf/addons/upgrade/to-10.1-move-radius-configuration-parameters.pl
```

## RADIUS filter templates

---

RADIUS filters now support templated values like switch templates. This script will update your RADIUS filters to new format:

```
/usr/local/pf/addons/upgrade/to-10.1-radius-filter-template.pl
```

## New EAP configuration parameter in realm.conf file

---

A new EAP parameter has been added to [filename]realm.conf file. This script will add this parameter to your current configuration file:

```
/usr/local/pf/addons/upgrade/to-10.1-realm-conf.pl
```

## Status of rules

---

It's now possible to enable/disable rules in authentication sources. This script will add the new **status** parameter:

```
/usr/local/pf/addons/upgrade/to-10.1-rule-status.pl
```

## Support for CoA in Unifi controllers

---

Support for CoA for Unifi AP is now supported but requires to have the latest controller and AP firmware available. Make sure you run the latest version of the controller and firmware if you use Ubiquiti equipment.

## Database schema

---

Changes have been made to the database schema. You will need to update it accordingly. An SQL upgrade script has been provided to upgrade the database from the 10.0.0 schema to 10.1.0.

To upgrade the database schema, run the following command:

```
mysql -u root -p pf -v < /usr/local/pf/db/upgrade-10.0.0-10.1.0.sql
```

# Upgrading from a version prior to X.Y.Z

---

## Database schema

---

Changes have been made to the database schema. You will need to update it accordingly. An SQL upgrade script has been provided to upgrade the database from the X.X.X schema to X.Y.Z.

To upgrade the database schema, run the following command:

```
mysql -u root -p pf -v < /usr/local/pf/db/upgrade-X.X.X-X.Y.Z.sql
```

# Archived upgrade notes

---

## Upgrading from a version prior to 4.0.0

---

Upgrading an old version of PacketFence to v4 will be quite an endeavor. While it's entirely possible if done meticulously, we suggest you start from scratch and move your customizations and nodes information over to your new installation.

### Database schema update

The temporary password table has been extended to include roles information. Moreover, an "admin" user is now automatically created. The default password is also "admin". Finally, a new table has been added for saved searches in the new Web administrative interface.

```
mysql -u root -p pf -v < /usr/local/pf/db/upgrade-3.6.1-4.0.0.sql
```

### Other important changes

PacketFence v4 received a major overhaul, especially regarding the authentication sources. Authentication modules found in `conf/authentication/` are no longer being used and have been replaced by the `conf/authentication.conf` file. While this file can be hand-edited, you should create your authentication sources and perform roles-mapping using the Configuration > Users > Sources page from PacketFence's Web administrative interface.

Also, in PacketFence v4, the VLANs can be assigned in `conf/switches.conf` by constructing the parameter names from the VLAN names and the `Vlan` suffix. The VLAN names must match one of the default names (registration, isolation, macDetection, inline, and voice) or one of the defined roles. If you were using custom VLANs, you must create a new role per VLAN and assign them accordingly.

Other key changes were done, such as:

- moved remediation templates in `html/captive-portal/templates/violations` and converted them to Template Toolkit
- dropped `guests_admin_registration.category`
- dropped `guests_self_registration.access_duration`
- dropped `guests_self_registration.category`
- dropped `guests_self_registration.sponsor_authentication`

- dropped `guests_self_registration.sponsors_only_from_localdomain`
- dropped `ports.listeners`
- dropped `registration.auth` and `registration.default_auth`
- dropped `registration.maxnodes`
- dropped `registration.expire_*` and `registration.skip_*`
- dropped `trapping.blacklist`
- dropped support for `resetVlanAllPort` in `bin/pfcmd_vlan`
- dropped `sbin/pfredirect` binary
- splitted the `httpd` services in three: `httpd.admin`, `httpd.portal` and `httpd.webservices`
- domain-name is no longer required in each section of `networks.conf`

For all parameters related to authentication (categories, access duration, sponsor authentication, etc.), you should now set proper actions in the `conf/authentication.conf` file.

Finally, the `pf` must be sudoer access to the `/sbin/ip` (and others) binary. As root, please do:

```
echo "pf ALL=NOPASSWD: /sbin/iptables, /usr/sbin/ipset, /sbin/ip, /sbin/vconfig, /
sbin/route, /sbin/service, /usr/bin/tee, /usr/local/pf/sbin/pfdhcplistener, /bin/
kill, /usr/sbin/dhcpd, /usr/sbin/radiusd" >> /etc/sudoers
```

## Upgrading from a version prior to 4.0.1

---

This release only fixes various bugs and doesn't need the database schema to be modified. Simply update the file `/usr/local/pf/conf/currently-at` to match the new release number. === Upgrading from a version prior to 4.0.2

This release only fixes various bugs and doesn't need the database schema to be modified. Simply update the file `/usr/local/pf/conf/currently-at` to match the new release number.

LDAP SSL and STARTTLS is now correctly implemented. Make sure the server you specify in `authentication.conf` supports the encryption type requested on the port configured. Failure to do so will break LDAP and Active Directory authentication.

## Upgrading from a version prior to 4.0.3

---

You need to downgrade the version of `perl-Net-DNS` and `perl-Net-DNS-Nameserver` to version 0.65-4 in order to fix the issue with `pfdns` crashing.

## Upgrading from a version prior to 4.0.4

---

The parameter `guest_self_reg` in the `profiles.conf` file is no longer necessary. The self-registration is now automatically enabled if at least one external authentication source is selected (Email, SMS, SponsorEmail, or Oauth2).

## Upgrading from a version prior to 4.0.5

---

This release adds a new dependency on the Perl module `Apache::SSLLookup`. Once installed, update the file `/usr/local/pf/conf/currently-at` to match the new release number.

## Upgrading from a version prior to 4.0.6

---

### Changes to authentication API

The method `pf::authentication::authenticate` now expects an array of `pf::authentication::Source` objects instead of an array of source IDs.

The methods `getSourceByType`, `getInternalSources`, and `getExternalSources` of the module `pf::Portal::Profile` now return `pf::authentication::Source` objects instead of source IDs.

## Upgrading from a version prior to 4.1.0

---

### Database schema update

The `category` column in the `temporary_password` should not be mandatory.

Also, the `access_level` of the `temporary_password` table is now a string instead of a bit string.

Make sure you run the following to update your schema:

```
mysql -u root -p pf -v < /usr/local/pf/db/upgrade-4.0.0-4.1.0.sql
```

## Configuration changes

The parameters `trapping.redirecturl` and `trapping.always_use_redirecturl` from `pf.conf` (or `pf.conf.defaults`) were moved to the default portal profile in `profiles.conf`.

The parameter `registration.range` has been deprecated. Make sure you remove it from your configuration file.

The action `set_access_level` of authentication sources in `authentication.conf` must now match one of the admin roles defined in `adminroles.conf`. The previous level `4294967295` must be replaced by `ALL` and the level `0` by `NONE`.

Adjust your configuration files accordingly.

Once the configuration completed, update the file `/usr/local/pf/conf/currently-at` to match the new release number.

## Upgrading from a version prior to 4.2.0

---

### Database schema update

The person table has many new columns that can be used for registration.

The node table has new columns to store the time and bandwidth balances of a node.

The node table has also a new column to keep the audit-session-id from the RADIUS request to use with the CoA.

Added a new column `config_timestamp` in `radius_nas` table.

The locationlog table has new columns to store the switch IP and MAC when using dynamic controllers.

New table for inline (layer 3) accounting.

New table for WRIX data.

Make sure you run the following to update your schema:

```
mysql -u root -p pf -v < /usr/local/pf/db/upgrade-4.1.0-4.2.0.sql
```

### Configuration changes

The parameter `guests_self_registration.mandatory_fields` from `pf.conf` (or `pf.conf.defaults`) was moved to the default portal profile in `profiles.conf`.

The parameters `registration.gaming_devices_registration` and `registration.gaming_devices_registration_role` are replaced with `registration.device_registration` and `registration.device_registration_role`.

Adjust your configuration files accordingly.

The captive portal has been rewritten using the Catalyst MVC framework. Any customization to the previous CGI scripts will need to be ported to the new architecture.

Once the configuration completed, update the file `/usr/local/pf/conf/currently-at` to match the new release number.

## Upgrading from a version prior to 4.3.0

---

### Database schema update

The person table has 2 new column to keep the portal and the source used to authenticate.

The tables `email_activation` and `sms_activation` have been merged in a table named `activation`. It has an additional column to keep the portal used to register.

Make sure you run the following to update your schema:

```
mysql -u root -p pf -v < /usr/local/pf/db/upgrade-4.2.0-4.3.0.sql
```

### Configuration changes

The parameters `VlanMap` and `RoleMap` have been added in `switches.conf`; be sure to add them in the [default] switch section.

The OAuth passthroughs will not be activated unless `trapping.passthrough` in `pf.conf` is enabled. Make sure you enable it if you have OAuth authentication sources (Google, Facebook, Github, LinkedIn and Windows Live).

Once the configuration is completed, update the file `/usr/local/pf/conf/currently-at` to match the new release number.

## Upgrading from a version prior to 4.4.0

---

### Database schema update

Introduced the `iplog_history` table for easier cleanup of the existing `iplog` table.

Make sure you run the following to update your schema:

```
mysql -u root -p pf -v < /usr/local/pf/db/upgrade-4.3.0-4.4.0.sql
```



## Cache serialization

The serialization of the objects in the cache changed, making all the previous cached objects invalid. With PacketFence completely stopped do :

```
rm -fr /usr/local/pf/var/cache/*
```

Once completed, update the file `/usr/local/pf/conf/currently-at` to match the new release number (PacketFence 4.4.0).

## Upgrading from a version prior to 4.5.0

---

### Database schema update

The class table has a new column `delay_by`.

Make sure you run the following to update your schema:

```
mysql -u root -p pf -v < /usr/local/pf/db/upgrade-4.4.0-4.5.0.sql
```

### Violation configuration

A new parameter `delay_by` has been introduced in the violation configuration. Make sure to add the following to the `defaults` section of `conf/violations.conf` to avoid any problem.

```
delay_by=0s
```

Once completed, update the file `/usr/local/pf/conf/currently-at` to match the new release number (PacketFence 4.5.0).

## Upgrading from a version prior to 4.6.0

---

### Database schema update

The `locationlog` and `locationlog_history` table have 2 new columns `stripped_user_name` and `realm`. We added new INDEX on `iplog`, `violation` and `locationlog` tables.

Make sure you run the following to update your schema:

```
mysql -u root -p pf -v < /usr/local/pf/db/upgrade-4.5.0-4.6.0.sql
```

## Violation template pages language handling

Code to match violation template pages have been reworked. Make sure to lowercase FR to fr in french template files name.

## Realm configuration

Realm are now managed by Freeradius server so if your users authenticate with a username like [username@acme.com](mailto:username@acme.com) then add the realm acme.com in the Radius Realms configuration menu and in your Active Directory source select *Use stripped username*.

Once completed, update the file `/usr/local/pf/conf/currently-at` to match the new release number (PacketFence 4.6.0).

## Upgrading from a version prior to 4.7.0

---

### Database schema update

The `node` table has a new column (`machine_account`).

Make sure you run the following to update your schema:

```
mysql -u root -p pf -v < /usr/local/pf/db/upgrade-4.6.0-4.7.0.sql
```

Once completed, update the file `/usr/local/pf/conf/currently-at` to match the new release number (PacketFence 4.7.0).

## Upgrading from a version prior to 5.0.0

---

Upgrading a version of PacketFence older than 4.1 to v5 will be a complex undertaking. While it's entirely possible if done meticulously, we suggest you start from scratch and move your customizations and nodes information over to your new installation.

Please note that the sections below are cumulative. That is to say, if you are upgrading from version 4.3 to version 5.0 you must apply in order all changes in between the two versions, including database schema changes.

As always, taking a complete backup of your current installation is strongly recommended. A backup should contain a copy of all PacketFence files as well as a copy of the database. You can take a backup of the `pf` directory with the following command:

```
tar -C /usr/local -czf /root/packetfence.tar.gz pf
```

A backup of the database can be taken using the procedure described in the next section.

## Database schema update

Before making any changes to your database, ensure that you have a backup. A complete database backup can be taken using this command:

```
mysqldump --opt --routines -u root -p pf | gzip > /root/packetfence_db.sql.gz
```

If your database is more than a few hundred megabytes, you may also want to consider using a tool such as Percona XtraBackup which makes for much faster restores than mysqldump.

Multiple changes have been made to the database schema. You will need to update it accordingly. Since we will be dropping and recreating the *iplog* table it is essential that you have a backup if you need the data it contains.

Make sure you run the following to update your schema:

```
mysql -u root -p pf -v < /usr/local/pf/db/upgrade-4.7.0-5.0.0.sql
```

## Configuration changes

You must manually enter the MySQL password of the pf user in the `conf/pfconfig.conf` file. The MySQL password is saved in the `conf/pf.conf` file under the `[database]` section. Copy the following from `conf/pf.conf` to `conf/pfconfig.conf`:

```
pass=$YOURPASSWORDHERE
```

## Violations configuration

The violation triggers have been reworked for the new Fingerbank integration. We highly suggest you copy `conf/violations.conf.example` over `conf/violations.conf` and then reconfigure any violations you had before.

Also, make sure you adjust the following triggers to their new ID (Can be found under *Configuration*→*Fingerbank*):

- `USERAGENT` becomes `user_agent`
- `VENDORMAC` becomes `mac_vendor`

The `OS` trigger has been deprecated over the new `dhcp_fingerprint` trigger. You will need to adjust these triggers to the new ids as well as renaming them.

## iptables changes

The iptables configuration file doesn't use the generated rules `%%input_mgmt_guest_rules%%` anymore. Make sure you remove this line from `conf/iptables.conf`.

Also a lot of additions were made to the iptables configuration file. Make sure you add the new rules in `conf/iptables.conf.example` to your existing iptables file or execute the following command to replace the whole file.

```
cp /usr/local/pf/conf/iptables.conf.example /usr/local/pf/conf/iptables.conf
```

## Using EAP local authentication

If you are using EAP MS-CHAP local authentication, meaning your 802.1x connections authenticate against your local database, you will need to make sure you deactivate password encryption in the database. In the administration interface, go in *Configuration* → *Advanced* and set *Database passwords hashing method* to **plaintext**

Once completed, update the file `/usr/local/pf/conf/currently-at` to match the new release number (PacketFence 5.0.0).

## Upgrading from a version prior to 5.1.0

---

### Database schema update

Multiple changes have been made to the database schema. You will need to update it accordingly.

Make sure you run the following to update your schema:

```
mysql -u root -p pf -v < /usr/local/pf/db/upgrade-5.0.0-5.1.0.sql
```

### pfsetvlan and snmptrapd

These two services have been disabled by default. If you are using SNMP traps enforcement on your switches (like port-security), make sure you re-enable them in *Configuration* → *Services*.

### Active Directory domain join

The Microsoft Active Directory domain join configuration is now part of PacketFence. A migration script has been made so you can migrate an existing domain join into this configuration. Note that this step is not mandatory, as the old join method is still supported. But if you do not perform this step, you will not see its configuration from the PacketFence web administrative interface.

Simply execute the following script and follow its instructions `/usr/local/pf/addons/AD/migrate.pl`

Once completed, update the file `/usr/local/pf/conf/currently-at` to match the new release number (PacketFence 5.1.0).

## Upgrading from a version prior to 5.2.0

---

### Database schema update

Multiple changes have been made to the database schema. You will need to update it accordingly.

Make sure you run the following to update your schema:

```
mysql -u root -p pf -v < /usr/local/pf/db/upgrade-5.1.0-5.2.0.sql
```

### Database monitoring host

If you are using an Active/Active cluster, you will need to adjust the monitoring database host to point to your database as it is not forced anymore.

In `conf/pf.conf` :

```
[monitoring]
db_host=127.0.0.1
```

### New *portal* interface type

If you are using email registration, web-auth enforcement (external captive-portal), device registration feature, or anything that would require to access the captive portal from outside the registration/isolation VLANs, you might want (actually, you need otherwise it will no longer works!) to add the *portal* type to the existing *management* interface.

In `conf/pf.conf` :

```
[interface eth42]
type=management,portal
```

Once completed, update the file `/usr/local/pf/conf/currently-at` to match the new release number (PacketFence 5.2.0).

## Upgrading from a version prior to 5.3.0

---

### Database schema update

Changes have been made to the database schema. You will need to update it accordingly.

Make sure you run the following to update your schema:

```
mysql -u root -p pf -v < /usr/local/pf/db/upgrade-5.2.0-5.3.0.sql
```

## Debian and Ubuntu

A downgrade in a package version may cause an error when trying to upgrade.

If you receive this error:

```
The following packages have unmet dependencies:
 packetfence : Depends: libhtml-formhandler-perl (= 0.40013-2) but 0.40050-2 is to
 be installed
E: Unable to correct problems, you have held broken packages.
```

Run the following commands:

```
# dpkg -r --ignore-depends=packetfence libhtml-formhandler-perl
# apt-get install libhtml-formhandler-perl libtemplate-autofilter-perl libmoop-
perl
# apt-get install packetfence packetfence-config packetfence-pfcmd-suid libdist-
checkconflicts-perl libimport-into-perl
```

Once completed, update the file `/usr/local/pf/conf/currently-at` to match the new release number (PacketFence 5.3.0).

## Upgrading from a version prior to 5.4.0

---

### Database schema update

Changes have been made to the database schema. You will need to update it accordingly.

Make sure you run the following to update your schema:

```
mysql -u root -p pf -v < /usr/local/pf/db/upgrade-5.3.0-5.4.0.sql
```

### Authentication sources rules rework

Authentication sources rules have been reworked in a way to differentiate an *authentication* rule and an *administration* rule. Codewise, that means that codeflow will look into specific types of rules depending of the use case.

Please take a minute or two to go through the existing rules for each of the authentication sources and make sure there is no *administration* class actions into an *authentication* class rule and vice versa, otherwise the "invalid" action will be ignored.

Authentication sources rules structure is as follow:

- *authentication* rule class available actions:
  - Set role (set\_role)
  - Set access duration (set\_access\_duration)
  - Set unregistration date (set\_unreg\_date)
- *administration* rule class available actions:
  - Set access level of Web admin (set\_access\_level)
  - Mark as sponsor (mark\_as\_sponsor)

For example, if an existing rule is as follow:

- Name: AllAdmins
- Class: No class defined since the class attribute is new
- Conditions: ...
- Actions:
  - Set access level of Web admin → ALL
  - Set role → default
  - Set access duration → 24H

That existing rule will default to the *authentication* class if none is being set. If that's the case, the first action "Set access level of Web admin" will then be ignored.

To replicate that existing rule with the new classes, you would have to create two separate rules, as follow:

Rule for *administration* purposes

- Name: AllAdmins\_admin
- Class: administration
- Conditions: ...
- Actions:
  - Set access level of Web admin → ALL

Rule for *authentication* purposes

- Name: AllAdmins\_auth
- Class: authentication
- Conditions: ...

- Actions:
  - Set role → default
  - Set access duration → 24H

Configuration will be validated on every start / restart so that "bogus" authentication sources / rules can be identified.

## OAuth2 authentication sources changes

The Facebook API now requires to specify the fields to be defined in the query. In all your facebook sources, change the parameter `protected_resource_url` to [https://graph.facebook.com/me?fields=id,name,email,first\\_name,last\\_name](https://graph.facebook.com/me?fields=id,name,email,first_name,last_name)

Change the parameter `scope` to `user,user:email` in all your Github sources as PacketFence is now fetching the email address of the user when registering with Github.

## StatsD configuration changes

`monitoring.statsd_host` and `monitoring.statsd_port` have been removed from `pf.conf`. If you have specified a specific host or port, remove them from your configuration and change them in `/usr/local/pf/lib/pf/StatsD.pm`

Once completed, update the file `/usr/local/pf/conf/currently-at` to match the new release number (PacketFence 5.4.0).

## Upgrading from a version prior to 5.5.0

---

### Database schema update

Changes have been made to the database schema. You will need to update it accordingly.

Make sure you run the following to update your schema:

```
mysql -u root -p pf -v < /usr/local/pf/db/upgrade-5.4.0-5.5.0.sql
```

### VLAN Filter configuration changes

The VLAN filter has been reworked to use a more generalized syntax to allow more complex filters to be created.

This mean nested conditions no longer need to specify the attribute in the condition.

So the following attribute



```
[condition]
filter=node_info
attribute=category
operator=is
value=default
```

Should be rewritten as

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

The older syntax is still supported but will be deprecated in a future release.

The operators `match` and `match_not` has changed their behavior. They will match (or not match) the exact string given in the condition. The following condition

```
[condition]
filter=node_info.computername
operator=match
value=^Bob
```

Will match `node_info.computername` only if it contains `^Bob`. It will not match if `node_info.computername` start with `Bob`

If you need to use a regex then use the `regex/regex_not` operator. So the following condition should be changed from

```
[condition]
filter=node_info.mac
operator=match
value=^00:
```

To the following

```
[condition]
filter=node_info.mac
operator=regex
value=^00:
```

## pf.conf configuration file changes

The following parameters have been removed from `pf.conf`. Make sure to remove them from your file if configured.

- `alerting.wins_server`
- `alerting.admin_netbiosname`

## violations.conf configuration file changes

Violations have been reworked and configuration changes are necessary in order to maintain functionality.

In violations.conf the following actions have been renamed, please update them accordingly.

- trap → reevaluate\_access
- email → email\_admin

The following action have been removed from the violations :

- popup

Also in violations.conf, the parameter whitelisted\_categories has been renamed into whitelisted\_roles

## Billing configuration change

The parameter **billing\_engine** of the Portal Profiles has been deprecated. Remove it from all your profiles configuration in `/usr/local/pf/conf/profiles.conf`.

The billing engine of PacketFence has been reworked completely.

It will require to reconfigure existing billing providers from scratch as there is no retro-compatibility with the previous configuration.

Please see the Administration Guide for details on how to configure the billing engine.

## Mod\_qos configuration changes

Mod\_qos configuration has been moved from "services" to "captive\_portal" section. Make sure to apply the appropriate changes if needed.

Once completed, update the file `/usr/local/pf/conf/currently-at` to match the new release number (PacketFence 5.5.0).

# Upgrading from a version prior to 5.6.0

---

## Database schema update

Changes have been made to the database schema. You will need to update it accordingly.

Make sure you run the following to update your schema:

```
mysql -u root -p pf -v < /usr/local/pf/db/upgrade-5.5.0-5.6.0.sql
```

## Extension points changes

The file `lib/pf/vlan/custom.pm` has now been renamed to `lib/pf/role/custom.pm`. Most of the customizations that used to be made in `vlan/custom.pm` can now be handled by configuring a `vlan` filter. You should take a good look at your existing `vlan/custom.pm` and consider porting the changes to `conf/vlan_filters.conf`.

## VLAN filters changes

The scopes for the VLAN filters have changed. The following have been renamed according to these rules:

NormalVlan → RegisteredRole  
 RegistrationVlan → RegistrationRole  
 ViolationVlan → ViolationRole  
 InlineVlan → InlineRole

If you have defined any filters in `/usr/local/pf/conf/vlan_filters.conf`, make sure to rename all references to the left hand side with the new names on the right hand side.

## Default type for the switches

The default type for the switches now needs to be set explicitly. Add the following line in the default section of `/usr/local/pf/conf/switches.conf`

```
type=Generic
```

Once completed, update the file `/usr/local/pf/conf/currently-at` to match the new release number (PacketFence 5.6.0).

# Upgrading from a version prior to 5.7.0

---

## Suricata violation trigger renaming

With the introduction of the ability to trigger a violation based on a MD5 hash detected by Suricata, a new trigger type has been introduced, requiring the modification of the actual *suricata* trigger. Make sure to go through your violations configuration and change any *suricata* trigger name for *suricata\_event*.

## Database schema update

Changes have been made to the database schema. You will need to update it accordingly.

Make sure you run the following to update your schema:

```
mysql -u root -p pf -v < /usr/local/pf/db/upgrade-5.6.0-5.7.0.sql
```

Once completed, update the file `/usr/local/pf/conf/currently-at` to match the new release number (PacketFence 5.7.0).

## Upgrading from a version prior to 6.0.0

---

Upgrading PacketFence from a version older than v6.0.0 will be a complex undertaking. While it's entirely possible if done meticulously, we suggest you start from scratch and move your customizations and nodes information over to your new installation.

### Devices parking

The new registration devices parking requires that you add the following violation in `/usr/local/pf/conf/violations.conf`

```
[1300003]
priority=1
desc=Parking violation
max_enable=3
grace=10m
actions=log,reevaluate_access
enabled=Y
auto_enable=Y
vlan=registration
trigger=Internal::parking_detected
```

### Chained authentication

The chained source has been deprecated in favor of a fully customizable flow in the captive portal.

Make sure you delete the source **BEFORE** upgrading your installation.

Once you upgrade, configure a portal module for each of your sources and a chained one that contains both. Refer to the administration guide for a detailed example.

### Redesigned captive portal

The parameter `mandatory_fields` of the Portal Profiles has been deprecated. Remove it from all the profiles in `profiles.conf`

To configure mandatory fields in the portal, refer to the *Portal Modules* section of the Administration guide

You need to add the `root_module` parameter to your default portal profile. In `profiles.conf` add `root_module=default_policy` to the default portal profile

## Changes to OAuth2 sources callback URL

All the OAuth2 sources you have configured (Facebook, Github, Google, LinkedIn, Twitter, Windows Live) need to be adjusted as the redirect URL is now the same for all the types.

In the admin interface change **Portal URL** from `https://YOUR_HOSTNAME/oauth2/SOURCE_TYPE` to `https://YOUR_HOSTNAME/oauth/callback` (where `SOURCE_TYPE` would be the lower case name of the source type). Note that this parameter is named `redirect_url` in the configuration file.

## Changes to Cisco Web auth

Use the `Cisco::Catalyst_2960` switch module instead of the `Cisco::Catalyst_2960_http` as switch type.

Use the `Cisco::WLC` switch module instead of the `Cisco::WLC_http` as switch type.

The `portalURL` configuration parameter is now configured per-role so make sure you have `http://ip_portal/$session_id` assigned to the registration role in the **Role by Web Auth URL** section of the switch configuration.

See the Network Device configuration guide for additional details.

## SMS carrier database table

Google Project Fi have been added as a supported carrier. Since an ID is hardcoded on creation of a new entry in the `sms_carrier` database table, a manual intervention may be required in the case the database schema update fails.

## pf.conf configuration parameters

`expire` and `maintenance` section have been reworked and `expire` section is no longer a thing. Make sure to adjust configuration parameter accordingly if needed;

- `expire.node` is now `maintenance.node_cleanup_window`
- `expire.iplog` is now `maintenance.iplog_cleanup_window`
- `expire.locationlog` is now `maintenance.locationlog_cleanup_window`
- `expire.radius_audit_log` is now `maintenance.radius_audit_log_cleanup_window`
- `expire.traplog` is now `maintenance.traplog_cleanup_window`

## node category / role

The `REJECT` role is now a default standard role. If you already have such role, make sure no conflict exists.

Also, add the following line to the default section of `switches.conf` :

```
REJECTVlan = -1
```

## Changes to the generated smb.conf

If you have a domain configured directly in PacketFence (in *Configuration→Domains*), you need to re-generate the associated configuration files as changes have been made to the samba configuration.

Using the CLI `/usr/local/pf/bin/pfcmd generatedomainconfig` or in the admin interface in *Configuration→Domains*, click *Refresh domain configuration*

## Upgrade from FreeRADIUS 2 to FreeRADIUS 3

PacketFence 6 relies on FreeRADIUS 3 rather than FreeRADIUS 2 as provided in PacketFence 5. The configuration files, directory layout and "unlang" directives have changed significantly. The packaging will automatically rename the existing `raddb` directory to `raddb-pre6`. All your existing configuration and certificates (if stored under `raddb/certs`) should be preserved but may need to be merged with the new `raddb` directory layout if you customized them.

The configuration files under `conf/radiusd/` **example have also changed. Make sure to compare them to your `conf/radiusd/` files** if you have any customizations, and merge any `*.rpmnew` files that may have been created by the packaging.

The default location for the FreeRADIUS server certificates has changed from `conf/ssl/` to `raddb/certs/`. The configuration of the certificates location is in `conf/radiusd/eap.conf`. You may point it to any valid certificate and key by setting the value of `certificate_file` and `private_key_file` respectively. It is not recommended to use the same server certificate for the HTTP services and the RADIUS server as the requirements for each are different. Reusing the same certificate will work, but you would be well advised to consider separate certificates.

Finally, the database schema for the RADIUS accounting tables and stored procedures have changed. Make sure to apply the database changes as described in the following section.

## Database schema update

Significant changes have been made to the database schema. You will need to update it accordingly. An SQL upgrade script has been provided to upgrade the database from the 5.7 schema to 6.0.

Since the schema of the `radacct` table has been reworked, the script will rename the existing table to `radacct2` and insert its content into the new `radacct` table. If your existing `radacct` table is large (as is sometimes the case), the operation may take a long time and consume a significant amount of disk space. Make sure to have plenty of both before running the upgrade script.

You can estimate the size of the existing `radacct` table by running the following command:

```
mysql> SELECT table_name AS "Table", round((((data_length + index_length) / 1024 / 1024), 2) "Size in MB" FROM information_schema.TABLES WHERE table_schema = "pf" AND table_name = "radacct";
```

You should have at least twice as much space as that table uses in the filesystem on which the MySQL data directory is mounted (usually `/var/lib/mysql`).

If you do not have enough space or time, you may consider truncating the `radacct` table (or simply deleting some of the rows) before running the upgrade script.

When ready, run the following to update your schema:

```
mysql -u root -p pf -v < /usr/local/pf/db/upgrade-5.7.0-6.0.0.sql
```

Once completed, update the file `/usr/local/pf/conf/currently-at` to match the new release number (PacketFence 6.0.0).

You will also want to drop the `radacct2` table from the database as it will no longer be needed.

## Upgrading from a version prior to 6.1.0

---

Significant changes have been made to the database schema. You will need to update it accordingly. An SQL upgrade script has been provided to upgrade the database from the 6.0 schema to 6.1.

To upgrade the database schema, run the following command:

```
mysql -u root -p pf -v < /usr/local/pf/db/upgrade-6.0.0-6.1.0.sql
```

Once completed, update the file `/usr/local/pf/conf/currently-at` to match the new release number (PacketFence 6.1.0).

## Dynamically created local secret

The management IP(s) of PacketFence are now defined as switches with a forced RADIUS secret defined in `/usr/local/pf/conf/local_secret`. Make sure you reconfigure the secret in the file if necessary and that this file is synchronized on all your cluster members if that applies. Note that this doesn't affect the RADIUS secret you have configured for wireless controllers and switches. It only affects RADIUS requests that originate from the management IP(s)

## Changes to LinkedIn source

A change to the authorize URL of LinkedIn was made. Make sure to change the *API Authorize Path* in all your LinkedIn source to `/uas/oauth2/authorization`.

## Upgrading from a version prior to 6.2.0

---

Changes have been made to the database schema. You will need to update it accordingly. An SQL upgrade script has been provided to upgrade the database from the 6.1 schema to 6.2.

To upgrade the database schema, run the following command:

```
mysql -u root -p pf -v < /usr/local/pf/db/upgrade-6.1.0-6.2.0.sql
```

Once completed, update the file `/usr/local/pf/conf/currently-at` to match the new release number (PacketFence 6.2.0).

## Upgrading from a version prior to 6.2.1

---

Changes have been made to the `httpd.admin` configuration. Make sure you copy the `conf/httpd.conf.d/httpd.admin.tt.example` file over `conf/httpd.conf.d/httpd.admin.tt`. If you customized that file in any way, you will have to merge the changes.

Restart the `httpd.admin` process once that is done by running `/usr/local/pf/bin/pfcmd service httpd.admin restart`

Once completed, update the file `/usr/local/pf/conf/currently-at` to match the new release number (PacketFence 6.2.1).

## Upgrading from a version prior to 6.3.0

---

Changes have been made to the database schema. You will need to update it accordingly. An SQL upgrade script has been provided to upgrade the database from the 6.2 schema to 6.3.

To upgrade the database schema, run the following command:

```
mysql -u root -p pf -v < /usr/local/pf/db/upgrade-6.2.0-6.3.0.sql
```

Once completed, update the file `/usr/local/pf/conf/currently-at` to match the new release number (PacketFence 6.3.0).

## RADIUS configuration file changes

The following file: `/usr/local/pf/conf/radiusd/eap.conf` was modified to use `TemplateToolkit`, you will need to replace it by `/usr/local/pf/conf/radiusd/eap.conf.example`, make sure to re-edit the new file and add your certificate if needed.

## Samba cache directory changed

Rejoining the domains from PacketFence GUI is required.

Go under **Configuration**→**RADIUS**→**Domains** and click **Rejoin** for each domain configured.

## Configuration changes to the Provisioning and Scanning

The configuration of the Scan engines and the Provisioners has been reworked to use the Fingerbank device IDs in the OS matching. `scan.conf` and `provisioning.conf` need to be migrated to



use the new values. A migration script should be run `# /usr/local/pf/addons/upgrade/to-6.3-os-rewrite.pl` to migrate the configuration. This will output the migrated configuration in `/usr/local/pf/conf/provisioning.conf.new` and `/usr/local/pf/conf/scan.conf.new`. First run the script and then validate that their content is fine. Once that is done, copy the files over the original ones using :

```
``` # cp /usr/local/pf/conf/provisioning.conf.new /usr/local/pf/conf/provisioning.conf # cp /usr/local/pf/conf/scan.conf.new /usr/local/pf/conf/scan.conf # /usr/local/pf/bin/pfcmd configreload hard ```
```

## Fingerbank database moving to MySQL (optional but highly suggested)

The Fingerbank database can now be hosted in the same MySQL database PacketFence uses.

In order to do so, you need to collect the database credentials from the PacketFence configuration:

```
``` # /usr/local/pf/bin/pfcmd pfconfig show resource::Database $VAR1 = { pass => myPassword, db => pf, user => pf, port => 3306, host => localhost }; ```
```

Now, you need to create the database and assign the proper rights to the user by executing the following commands:

```
``` # mysql -u root -p -e "CREATE DATABASE pf_fingerbank" # mysql -u root -p -e "GRANT ALL PRIVILEGES ON pf_fingerbank.* TO pf@% IDENTIFIED BY myPassword" # mysql -u root -p -e "GRANT ALL PRIVILEGES ON pf_fingerbank.* TO pf@localhost IDENTIFIED BY myPassword" ```
```

Replace **myPassword** by the password displayed (**pass**) when running the first command.

Next, head to *Configuration*→*Fingerbank Settings* in the web administration interface and configure the following parameters:

- MySQL host : set this to the value of **host** you got from running the command above.
- MySQL port : set this to the value of **port** you got from running the command above.
- MySQL username : set this to the value of **user** you got from running the command above.
- MySQL password : set this to the value of **pass** you got from running the command above.
- MySQL database : set this to **pf\_fingerbank**.

After saving those new parameters, at the top of the same page, click *Initialize MySQL database* to start the import process. Once that is completed, you will receive an e-mail to the one configured for alerting and PacketFence will start using the MySQL backend for the Fingerbank database.

## Upgrading from a version prior to 6.4.0

---

### Database schema updates

Changes have been made to the database schema. You will need to update it accordingly. An SQL upgrade script has been provided to upgrade the database from the 6.3 schema to 6.4.

To upgrade the database schema, run the following command:

```
mysql -u root -p pf -v < /usr/local/pf/db/upgrade-6.3.0-6.4.0.sql
```

### Changes to web authentication configuration

Rework of the external captive portal capabilities involves some significant changes in the switch modules configuration. Some switch modules have been moved to other ones and some others have been removed. Please adjust the configuration (type) accordingly within switches.conf.

- AeroHIVE::AP\_http → AeroHIVE::AP
- Meraki::AP\_http → Meraki::MR
- Meraki::AP\_http\_V2 → Meraki::MR\_v2
- Xirrus:AP\_http → Xirrus

To instruct a switch module to perform external captive portal enforcement, a new switch configuration parameter have been added. Make sure to adjust the following parameter to your needs in switches.conf

```
``` ExternalPortalEnforcement = Y ```
```

External captive portal URLs have also changed. Change them accordingly depending on the type of equipment you use:

- AeroHIVE: [http://portal\\_IP/AeroHIVE::AP](http://portal_IP/AeroHIVE::AP)
- Aruba: [http://portal\\_IP/Aruba](http://portal_IP/Aruba)
- Cisco Catalyst 2960: [http://portal\\_IP/Cisco::Catalyst\\_2960](http://portal_IP/Cisco::Catalyst_2960)
- Cisco WLC: [http://portal\\_IP/Cisco::WLC](http://portal_IP/Cisco::WLC)
- CoovaChilli: [http://portal\\_IP/CoovaChilli](http://portal_IP/CoovaChilli)
- Meraki: [http://portal\\_IP/Meraki::MR](http://portal_IP/Meraki::MR)
- Ruckus: [http://portal\\_IP/Ruckus](http://portal_IP/Ruckus)
- Xirrus: [http://portal\\_IP/Xirrus](http://portal_IP/Xirrus)

Where `portal_ip` is the IP Address (or DNS name) of your captive portal as it was configured before

## Changes to WMI

If you use WMI, you must modify `conf/wmi.conf` in order to make sure that a namespace parameter is defined for each rule. For example, you could have:

```
...
```

```
request=select * from Win32_Process where name=CcmExec.exe action=[sccm] namespace=ROOT\cimv2 on_tab=1 ````
```

## Changes to default cronjob

Upon PacketFence installation, a default cronjob will be in `/etc/cron.d/`. You should make sure you do not invoke the `/usr/local/pf/addons/backup-and-maintenance.sh` script from any other cronjob.

Once completed, update the file `/usr/local/pf/conf/currently-at` to match the new release number (PacketFence 6.4.0).

# Upgrading from a version prior to 6.5.0

---

## Database schema updates

Changes have been made to the database schema. You will need to update it accordingly. An SQL upgrade script has been provided to upgrade the database from the 6.4 schema to 6.5.

To upgrade the database schema, run the following command:

```
mysql -u root -p pf -v < /usr/local/pf/db/upgrade-6.4.0-6.5.0.sql
```

## Custom code warning

The method signature of `pf::node::node_register` has been modified. Make sure you adjust any custom code / external scripts to handle the new returned values.

## Switches Configuration

You must rename "controllerPort" to "disconnectPort" in your `switches.conf` configuration file. You can automate this using:

```
cd /usr/local/pf
find . -name "switches.conf" -exec sed -i "s/controllerPort/disconnectPort/g" '{}' \;
```

## Eduroam

Eduroam authentication source is now an "exclusive" authentication source rather than an "external" one. That being said, make sure to adjust portal profile accordingly (an "exclusive" authentication source can be the only one configured in a portal profile).

## Improved Logging

In order to be sure all your logging facilities use the new logging backend which ensures the processes will not die in case of a logging failure, you must execute the following command:

```
cd /usr/local/pf
find conf/log.conf.d/ -type f -exec sed -i.bak "s/Log::Log4perl::Appender::File/
pf::log::FileAppender/g" {} \; ; find conf/log.conf.d/ -name '*.bak' -delete
```

## Email templates

The email templates have been moved from `/usr/local/pf/conf/emails/` to `/usr/local/pf/html/captive-portal/templates/emails/` as they are now configurable by portal profile. Also you can configure the language in which PacketFence should send emails to the administrator in the Advanced section of the configuration.

Make sure you run the following command after upgrading:

```
/usr/local/pf/bin/pfcmd cache configfiles clear
```

## Violations

When whitelisting roles in a violation, the registration role will now match unregistered devices where before it would never match. Make sure to go through violations that may include this role to make sure it is relevant.

## Database schema updates

The "configfile" and "traplog" database tables are now deprecated. If you wish to reclaim the disk space used by those two database tables, they should be manually removed.

Once completed, update the file `/usr/local/pf/conf/currently-at` to match the new release number (PacketFence 6.5.0).

## Default RoleMap for the switches

If you were using the default `RoleMap = Y` in the `conf/switches.conf` it's disabled by default now. You will need to put `RoleMap = Y` under your switches or switch group configuration.

# Additional Information

---

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

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

# Commercial Support and Contact Information

---

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

Inverse (<http://inverse.ca>) offers professional services around PacketFence to help organizations deploy the solution, customize, migrate versions or from another system, performance tuning or aligning with best practices.

Hourly rates or support packages are offered to best suit your needs.

Please visit <http://inverse.ca/> for details.

# GNU Free Documentation License

---

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