OpenDaylight & PacketFence install guide

for PacketFence version 7.4.0

OpenDaylight & PacketFence install guide

by Inverse Inc.

Version 7.4.0 - Jan 2018

Table of Contents

About this Guide

This guide has been created in order to help sales engineers, product managers, or network specialists demonstrate the PacketFence capabilities on-site with an existing or potential customer. It can also provide guidelines to setup a proof of concept for a potential PacketFence deployment using the OpenDaylight SDN controller.

Assumptions

- You have a configured PacketFence environment that is currently working;
- You have, or will create, a server with CentOS 6 minimal installed on it (for the OpenDaylight server);
- The IP address of the OpenDaylight controller is 192.168.1.10;
- The IP address of the PacketFence server is 192.168.1.5.

Installation

Step 1: Install CentOS 6 minimal

You will first need a server with the minimal installation of CentOS 6 packages on it. This server needs two network adapters that are on the same network and at least 2 GB of RAM.

Step 2: Install the necessary packages

```
yum install git wget java-1.7.0-openjdk java-1.7.0-openjdk-devel
```

In order to build the OpenDaylight controller, you will need to install Apache Maven.

```
wget https://raw.githubusercontent.com/monksy/centos-maven-install/master/maven-
install.sh
sh maven-install.sh
```

OS configuration

You will now need to stop iptables and make sure it doesn't start on boot.

```
service iptables stop
chkconfig iptables off
```

Add the server name in `/etc/hosts`.

Step 3: Install the OpenDaylight controller

First, you will need to clone the OpenDaylight code repository on your server.

```
mkdir /var/odl
cd /var/odl
git clone https://git.opendaylight.org/gerrit/p/controller.git
cd controller
```

Add these maven options to your environment.

```
export MAVEN_OPTS="-Xmx1024m -XX:MaxPermSize=512m"
```

The PacketFence plugin has been tested on commit id ec7cac6937c6beacf4d6dd9b44bb998d7b0068d4 of the OpenDaylight controller. To use the specific commit that was used by Inverse during testing, issue the following command. Note that this step is optional but in case of any issues, try to use this specific commit id.

```
git checkout ec7cac6937c6beacf4d6dd9b44bb998d7b0068d4
```

Install the controller distribution.

```
mvn clean install -DskipTests
```

At the end you should see an output similar to this

```
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 41:24.140s
[INFO] Finished at: Tue Oct 28 19:52:03 EDT 2014
[INFO] Final Memory: 346M/931M
[INFO] ------------------------------------------------------------------------
```

Step 4: Install the PacketFence plugin

Download the PacketFence plugin from the official source code repository:

```
wget https://github.com/inverse-inc/packetfence/raw/feature/opendaylight/addons/
opendaylight/target/odlpf-0.1.jar
```

Now copy it to the plugin directory

```
cp odlpf-0.1.jar /var/odl/controller/opendaylight/distribution/opendaylight/
target/distribution.opendaylight-osgipackage/opendaylight/plugins/
```

And we can now start the OpenDaylight controller

```
sh /var/odl/controller/opendaylight/distribution/opendaylight/target/
distribution.opendaylight-osgipackage/opendaylight/run.sh
```

After 2-3 minutes, OpenDaylight will be started. Press enter a few times and you should see the `osgi>` prompt. Verify that the plugin is properly loaded by typing *ss odlpf*. The state needs to be ACTIVE.

```
osgi> ss odlpf
"Framework is launched."


id  State       Bundle
38  ACTIVE      ca.inverse.odlpf_0.1.0
```

Now validate you have access to the OpenDaylight administration interface on http://192.168.1.10:8080

Step 5: Patching PacketFence

The SDN modules are not included with the current version of PacketFence.

In order to obtain the functionalities you will need to apply a patch on your current PacketFence installation.

Warning

This could break a working PacketFence installation. Install this on a non-production PacketFence server.

This patch is based on the most recent version of PacketFence so be sure to upgrade to the latest version available before applying the patch.

```
cd /usr/local/pf
wget http://inverse.ca/downloads/sdn.patch
patch -p1 < sdn.patch
```

In case of any conflicts please post the output of the patch command on the PacketFence development mailing list.

Once you are done, restart PacketFence using `service packetfence restart`

Step 6: Configuration

In PacketFence configure the webservices username and password under **Configuration → Integration → Webservices**

Still in PacketFence, add a new switch for the OpenDaylight controller

In `/usr/local/pf/conf/switches.conf` add the OpenDaylight controller configuration

```
[192.168.1.10]
type=OpenDaylight
mode=production
IsolationStrategy=DNS
wsTransport=HTTP
wsUser=admin
wsPwd=admin
```

Where :

- **type** is the module to use. Should be left to OpenDaylight
- **mode** should be set to production
- **IsolationStrategy** is for selecting the way the devices get isolated. This should be set to DNS in most cases.
- **wsUser** is the user to connect to the OpenDaylight NorthBound API
- **wsPwd** is the password to connect to the OpenDaylight NorthBound API
- **wsTransport** is the protocol that should be used to connect to the OpenDaylight NorthBound API. If you set it to HTTPS, make sure the proper certificates are in place.

In `/usr/local/pf/conf/iptables.conf`, add the following lines in the management section

```
# DNS (for the SDN stack)
-A input-management-if --protocol udp --match udp --dport 53  --jump ACCEPT
-A input-management-if --protocol tcp --match tcp --dport 53  --jump ACCEPT
# PORTAL (for the SDN stack)
-A input-management-if --protocol tcp --match tcp --dport 80  --jump ACCEPT
-A input-management-if --protocol tcp --match tcp --dport 443  --jump ACCEPT
```

The PacketFence OpenDaylight plugin requires a configuration file on the OpenDaylight controller in `/etc/packetfence.conf`. Here is an example :

```
host=192.168.1.5
port=9090
user=web
pass=services
controller_ip=192.168.1.10
pf_dns_ip=192.168.1.10
pf_dns_mac=0050569d000b
```

Where :

- **host** is the management ip of your PacketFence server
- **port** is the port of the webservices (default for PacketFence is 9090)
- **user** is the username of the PacketFence webservices
- **pass** is the password of the PacketFence webservices

- **controller_ip** is the IP of the OpenDaylight controller
- **pf_dns_ip** is the IP address on which PacketFence listens on for the DNS requests (should be your management)
- **pf_dns_mac** is the MAC address of the interface on which PacketFence listens on for the DNS requests (should be your management)

Step 7: Configure your network equipment

You should now configure your network equipment to communicate with the OpenDaylight controller.

Configure the equipment to forward unknown packets on non-uplink ports to the PacketFence server.

For now, the uplink port has to be port 1 on your equipment and is a limitation of the current architecture.

Step 8: Test

Connect a device to your network equipment and in the terminal where you have the OpenDaylight server running you should see JSON information about what is happening now.

Here is the valid output of a packet in event

```
Pkt. to /10.0.0.6 received by node 01:00:00:50:56:9d:00:05 on connector 2
Packet src port -20180 packet source int 2
{"id":"1","method":"sdn_authorize","params":
{"port":"2","controller_ip":"192.168.1.10",
"mac":"84:3a:4b:dc:09:ec","switch_id":"01:00:00:50:56:9d:00:05"},"jsonrpc":"2.0"}
{"jsonrpc":"2.0","id":"1","result":[{"strategy":"DNS","action":"isolate"}]}
Installing DNS outbound redirect flow
Installing return flow
Setting 84:3a:4b:dc:09:ec,-20180 38:22:d6:6c:8c:f5,10.0.0.6,2
```

Once the first event hits your OpenDaylight controller, if you don't have a plugin to manage the non-DNS traffic you should install flows that allow communication of the device on the network. Usually forwarding packets to and from the uplink should be enough. Make sure the flows you install have a lower priority than 1000. PacketFence uses priority 1000 and higher.

If your network equipment supports reactive forwarding with OpenDaylight, then it can be used in parallel with PacketFence to handle the non-DNS traffic.

Now once your computer obtains an IP address, doing an nslookup of any domain name on the connected device should now return a CNAME pointing to the PacketFence server.

Opening your browser will redirect you to the PacketFence captive portal. Register on the captive portal and then the DNS redirection will be deactivated.

Since the device may use DNS caching, the original domain may be unavailable, a restart of the browser could also be necessary.