Opnsense Firewall with OpenC2 Basic Commands to show a publisher and subscribe model of implementation.

The Opnsense Firewall has the following enabled

Consumer (the subscriber)

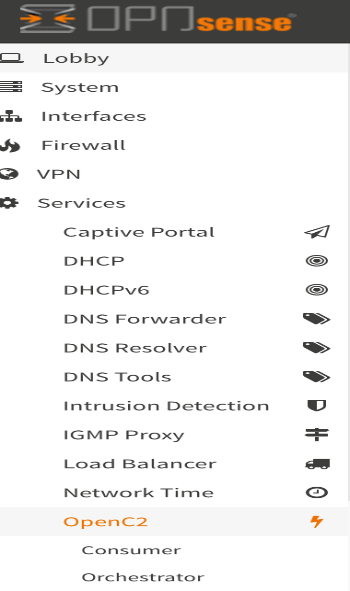
Orchestrator (the publisher)

A single Opnsense Firewall can act as both a consumer and orchestrator however that is only for evaluation and testing only

Configuring OpnSense Firewall with OpenC2

To configure the Opnsense Firewall with OpenC2 a user must do the following

Login to the web interface of the OpnSense Firewall Figure 1

Default <https://192.168.1.1> on the Port 0 interface of the physical firewall.

Login as a root user (administrator)

On the left hand panel click down into Services (see Figure 1)

Navigate to OpenC2

There are two sub menus

Consumer

A consumer is the function that a firewall is a subscriber and receives commands from a publisher - an orchestrator.

Orchestrator

An orchestrator is the function that is built onto the firewall in case there is not another system in the network that can act as an publisher. The role of an orchestrator is to publish various commands to one or more devices that are subscribed to the orchestrator as a consumer. These commands are current manually configured by a person in the web ui. The reference design is a showcase of what can be done visually. All of the functionality can be done via scripting and automation using many other tools that can send commands in a rapid fashion.

Setting up a consumer

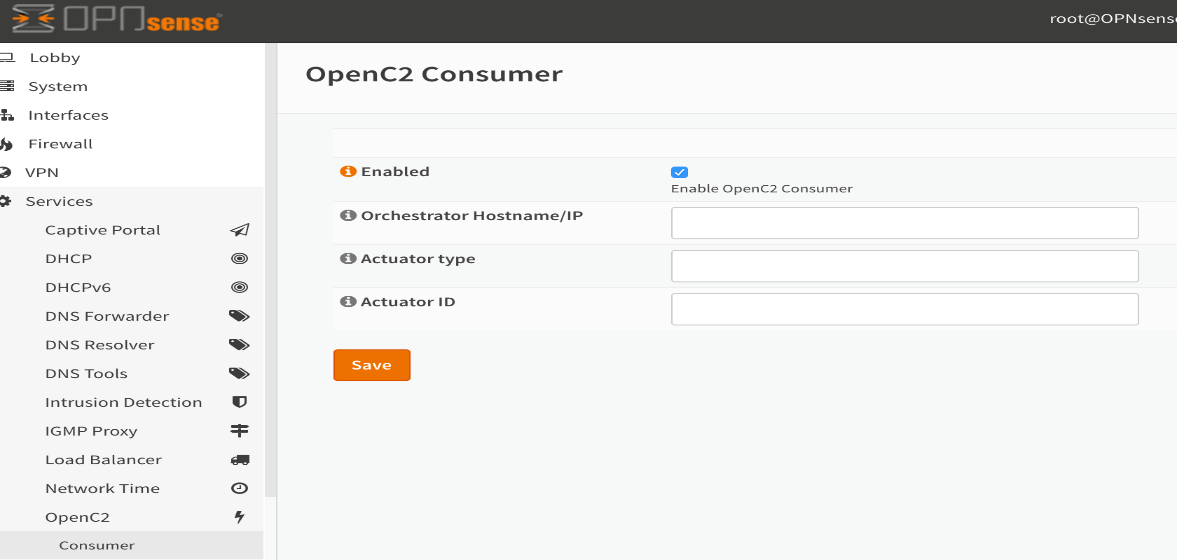
There are several choices. To enable OpenC2 to command this firewall, the first option is to enable the consumer (See Figure 2)

Select Enable (check mark)

In the field Orchestrator Hostname/IP - enter the orchestrator IP address or hostname if it resolves. If it is the local orchestrator, the IP should be 192.168.1.1

Actuator type and Actuator ID are optional fields used to identify the firewall in the OpenC2 ecosystem. Use these fields when specific commands should only be send to this specific firewall.

Figure 2

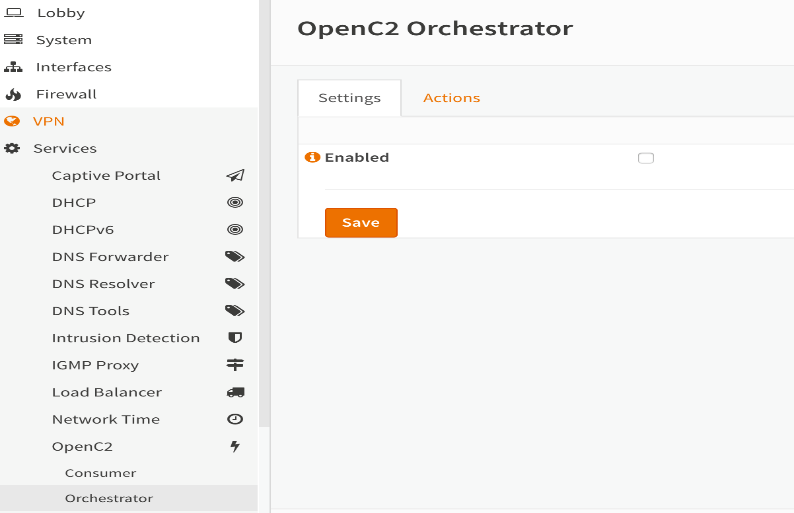


Setting up an orchestrator

At the Settings Tab

Enable the orchestrator and save the settings to enable OpenC2 functionality on the firewall to act as a publisher. See Figure 3

Figure 3



At the Actions Tab

There are several fields to configure for an orchestrator to function on the network

(See Figure 4)

Input the orchestrator connection string. The reference system uses port 5555

For the reference architecture, it should be itself which is local IP 127.0.0.1 as:

<tcp://127.0.0.1:5555>

Raw Action - is a raw JSON formatted action that a user may input a command to the firewall if they know the format of the string and action type. This has been added as additional functionality to showcase all functionality without specific section menus can be used in an orchestrator to send all commands the firewall supports

Block Network

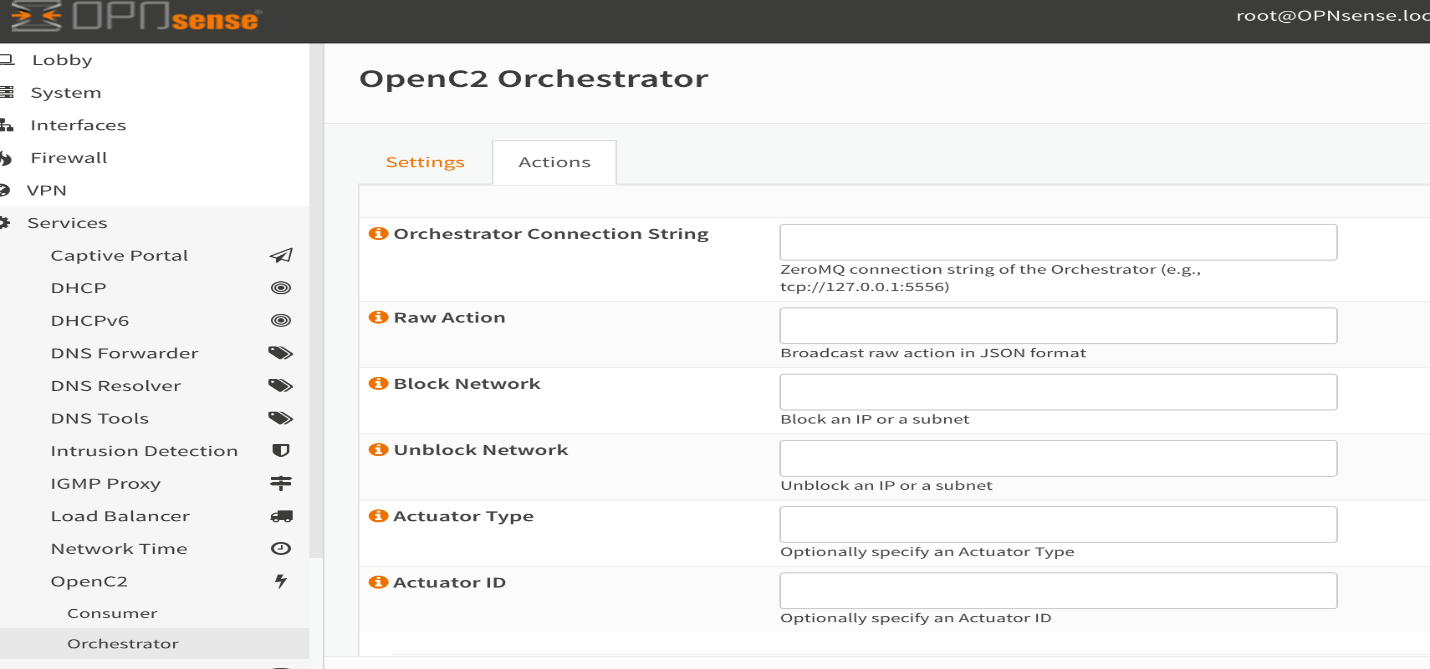
Enter a subnet or IP address to be blocked by the firewall

Unblock Network

Enter a subnet or IP address that was previously put in the block network to be unblocked

Enter an Actuator Type and Actuator ID that matches a firewall that these commands should only go to in the network - if the firewall openc2 consumer configuration has the same actuator type and actuator id then the commands will be executed on the firewall.

Figure 4



Explanation of the block / unblock network functionality

The basic functionality of the firewall functions are being leveraged on the opnsense functionality via the following command line:

/sbin/pfctl

this is the basic packet filtering command in the opnsense firewall.

The following two files control the functionality of the commands:

edge-action-deny (for blocking)

edge-action-allow (for unblocking)

The commands create an IP table called “blockedpeeps" and tracks the IP’s that are issued via the openc2 messages to block or unblock from the orchestrator to the publisher.

All of this is invisible to the end user of the firewall that is the consumer. 100% automated and no knowledge or insight needed to manage the actions performed from the orchestrator in the reference design.

This all assumes there is a person at the orchestrator creating commands to be issued. The reference implementation does not have automated commands being created and issued by an orchestrator currently.