

## ACTOR SUPPORT FOR WEB APPLICATION FIREWALLS (WAFS)

One of the use cases for Security Validation is sending traffic through a web application firewall (WAF). In such a test, an HTTPS Network Action would be run between two Actors, with the WAF in between them.



The WAF is configured as a virtual address for the test interface of the destination Actor. In doing so, the source Actor can send traffic through it.

In some environments, a WAF may require a liveness check (heartbeat) to pass for it to forward traffic to a destination. In general, an Actor only listens while performing an Action for which it is a destination. Thus, the Actor does not have a suitable endpoint for a WAF liveness check (which should always be available).

If it is not possible to disable the WAF liveness check for an Actor positioned behind a WAF, then follow the instructions to [configure a heartbeat listener](#) on the test interface of an Actor.

### Heartbeat playbook requirements

Your Actor must meet the following requirements to use the WAF heartbeat playbook:

- Installer-based Actor on RHEL 7 or CentOS 7
- Separate management and test interfaces
- Actor registered using Push communication mode
- Validation network control is set to True
- Test interface has static IP address

### Configure a heartbeat listener

You have two scripts to choose from to configure the heartbeat listener. The first automatically defines the variables for you and the second walks you through defining the variables.

#### Run the automatic WAF heartbeat configuration

1. Run the playbook using the following command:

```
$ sudo /opt/apps/verodin/node/contrib/playbook_exes/heartbeat_config_auto
```

2. When prompted, review the configuration, confirm that everything is correct, and answer **yes** to continue.



If the configuration is not correct and needs to be changed, answer **no**. Instead, use the manual configuration playbook to define the variables using the instructions that follow.

3. Add a virtual address in the Director to the primary test interface public IP of the Actor.

#### Run the manual WAF heartbeat configuration

1. Run the playbook, supplying appropriate config values when prompted.

```
$ sudo /opt/apps/verodin/node/contrib/playbook_exes/heartbeat_config
```

- The `sub_ip` is **192.168.250.1** most of the time. You only need to change this value if this IP is within the subnet of the test interface.
- The user and group variables are the values you defined when installing the Actor. The default for variables is generally **nodeone**.

2. Add a virtual address in the Director to the primary test interface public IP of the Actor.

## Test the configuration

Use the following tests to check the configuration.

### Heartbeat test

- To test the heartbeat, run the following command:

```
$ curl -k https://<test ip>/heartbeat.html
```

If it is working correctly, it should return the following:

```
<html>
<head>
<title>Heartbeat</title>
</head>
<body>
<p>&hearts; beats<p>
</body>
</html>
```

### Action test

- For a non-heartbeat test, run the following command:

```
$ curl -k https://<test ip>/<any_other_page>
```

A 502 error is expected since our Actor only listens when an Action is run.

```
<html>
<head><title>502 Bad Gateway</title></head>
<body>
<center><h1>502 Bad Gateway</h1></center>
<hr><center>nginx/1.16.1</center>
</body>
</html>
```

- Next, try to run an HTTPS Network Action with this Actor as destination. If the request is not blocked by the WAF, the Action should reach the destination Actor.