Application Note

Stateful Firewall, IPS or IDS Load-Balancing

Document version: v1.0

Last update: 8th November 2013

Purpose

Improve scallability of the security layer

Limitations when Load-Balancing firewalls

Firewall Load-Balancing is a complex task.
Please bear in mind the current document introduces the most common required features.
Please keep in mind the few points below:

- Unfortunatelly, VPN endpoints can't be load-balanced **unless** the firewalls NAT the VPN traffic with a local IP address known by the Load-Balancer
- Only deterministic load-balancing algorithm using source or destination IPs, or both, are compatible with stateful firewall load-balancing
- It is not recommended to translate addresses (NAT) with such infrastructure. That said, the current document will introduce it, so you will see how complicated the rules can be
- There must be a single default gateway per firewall

Each firewall load-balancing project is different and **this document is not exhaustive**, so don't hesitate to contact Exceliance through pre-sales or support team before starting

Complexity

Versions concerned

- Aloha 5.5 and above

Changelog

- 2013-10-16: Initial Version

Introduction

Nowadays, Firewalls are not only layer3 / layer4 packet inspectors. They also bring features such as intrusion prevention/detection system (IPS/IDS), anti-virus, anti-spam, content filtering, etc...
These processes consume a lot of resources and a single Firewall is most of the time not powerful enough. Many of them must be deployed and used in parallel to handle the traffic load.

Those firewalls must be **stateful**: they keep track of network streams states which cross them.
It means both ways of communication must pass through the same firewall, otherwise traffic not corresponding to an initialized state will be discarded by the other firewalls.

Synopsis

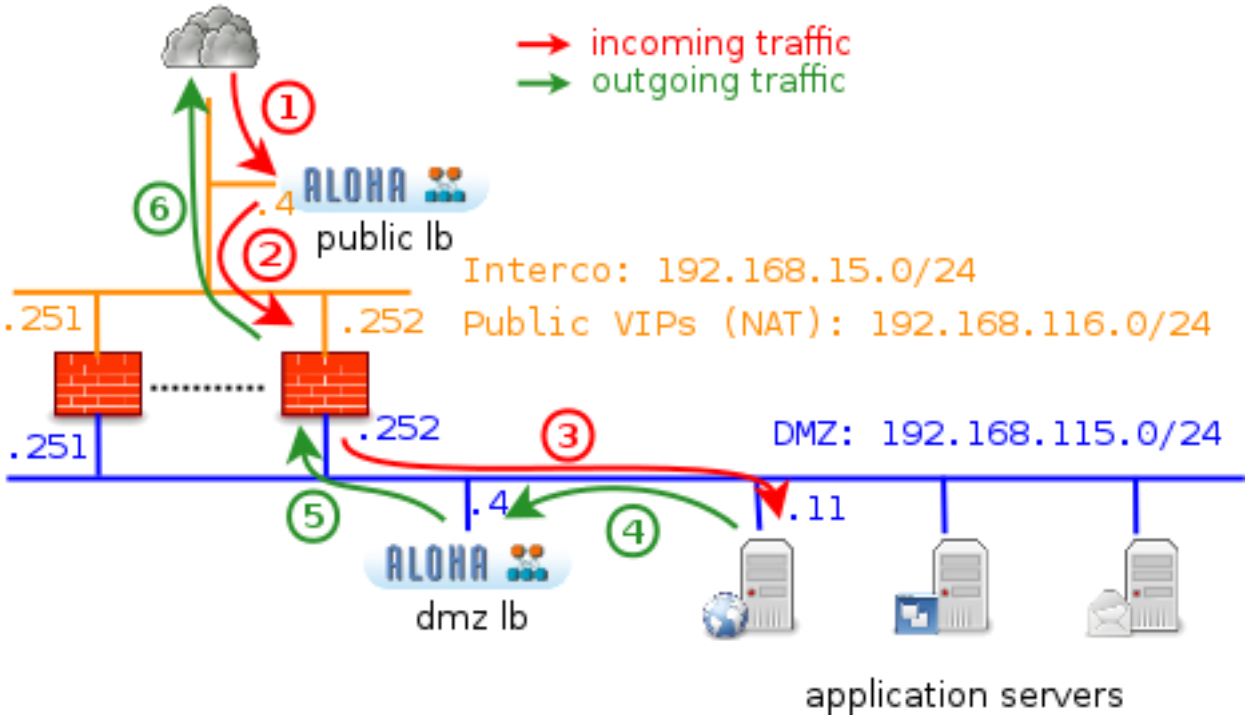
When Load-Balancing stateful firewalls, one must load-balance traffic per DMZ, with Load-Balancers on both public and private side (for each DMZ).

Diagram and flows

The Load-Balancer must be configured in **Direct Server Return (DSR)** mode, also known as **gateway**.
The core routers must route traffic to the **Public load-balancer** Virtual IP. In the meantime the default gateway for all **application servers** in the **DMZ (or internal)** must be configured with the **Load-Balancer** Virtual IP.

Diagram

The diagram below shows how the **Load-Balancers** and the **Firewalls** are connected.



Flows

On the diagram above, we can see both incoming and outgoing traffic through the different pieces of the architecture.
There are basically 6 steps for each connection crossing the platform:

1. incoming connection is routed to the **public LB** by the core network routers
2. the **public LB** chooses a **FW** based on its configuration then forward the connection to it
3. the **FW** gets the connection, analyses it then forward it to the **application server**
4. the **application server** answers through its default gateway, the **DMZ LB**
5. the **DMZ LB** forwards the response to the **FW** which managed the incoming connection *(remember, stateful FW...)*
6. the **FW** can forward the packet to the client, without passing through the **public LB**.

Configuration examples

In order to properly load-balance **Firewalls**, you must configure 2 features in the **Aloha Load-Balancer**:

1. **Flow Manager**: used to describe flows to apply load-balancing on
2. **Layer 4 LB**: load-balancing itself, choose the right **firewall** using the appropriate hash, and perform health checking

The Load-Balancers are deployed using a single network interface, named **eth0**. These examples could also apply on **Bonding** and **Vlan** interfaces as well.

Standard configuration without NAT

In this mode, there is no NAT, hence we can use a hash on both source and destination IP addresses.

public LB

The **public LB** has to balance incoming traffic from **external clients** to **FWs**.
Hence, here is the **flow manager** configuration describing it:

```
flow f_firewall director d_firewall
  match iface eth0
```

Below, the corresponding Load-Balancing configuration:

```
director d_firewall
  balance srcdst
  mode gateway
  check interval 10 timeout 2
  option arpcheck
  server firewall1 192.168.15.251 weight 10 check
  server firewall2 192.168.15.252 weight 10 check
```

dmz LB

The **dmz LB** has to balance outgoing traffic from **internal servers** to **external clients**.
Hence, here is the **flow manager** configuration describing it:

```
flow f_firewall director d_firewall
  match iface eth0
```

Below, the corresponding Load-Balancing configuration:

```
director d_firewall
  balance srcdst
  mode gateway
  check interval 10 timeout 2
  option arpcheck
  server firewall1 192.168.115.251 weight 10 check
  server firewall2 192.168.115.252 weight 10 check
```

Outbound NAT required

Outbound NAT may be required to allow some **internal servers** to access third party services on the internet, such as software updates.
Basically, the **application server** IP address will be NATed to the **Firewall** IP address (this could be any other IP address hosted on the **Firewall**).
So load-balancing outgoing traffic (on the **DMZ LB**) using the default rule (with the source+destination hash) is fine since the new public IP address will be re-used to route incoming traffic by the **Public LB**. Incoming traffic will be simply routed by the **public LB** to the IP address hosted by the **Firewall**.
The current example explains how to allow the application servers to go out on internet. Of course, the **Firewall** must be configured to allow some ports and to NAT this traffic.

public LB

The **public LB** has to route incoming traffic based on the destination IP, which is the **Firewall** one (or any other known IP hosted by the **Firewall**).
The **flow manager** has to match this traffic before the standard rule (which matches everything). Below, the corresponding **Flow Manager** configuration:

```
# exception rules for IPs dedicated to outbound NAT
flow f_fw_out permit
  match iface eth0 dst 192.168.15.251
  match iface eth0 dst 192.168.15.252

flow f_firewall director d_firewall
  match iface eth0
```

Below, the corresponding **Layer 4 LB** configuration:

```
director d_firewall
  balance srcdst
  mode gateway
  check interval 10 timeout 2
  option arpcheck
  server firewall1 192.168.15.251 weight 10 check
  server firewall2 192.168.15.252 weight 10 check
```

dmz LB

Below, the corresponding **Flow Manager** configuration:

```
flow f_firewall director d_firewall
  match iface eth0
```

Below, the corresponding **Layer 4 LB** configuration:

```
director d_firewall
  balance srcdst
  mode gateway
  check interval 10 timeout 2
  option arpcheck
  server firewall1 192.168.115.251 weight 10 check
  server firewall2 192.168.115.252 weight 10 check
```

Inbound NAT required

Inbound NAT may be required to give access to services hosted on a private IP address: the **Firewall** has to translate a public IP address to the private IP address of the **server** in the DMZ.
We can only rely on the **external IP** address, the one from the client, since the destination IP address will be changed by the **Firewall** from a public one to a private one.

The current example explains how to give access to the web server hosted on the private IP address 192.168.115.11 through the "public IP" 192.168.116.11.

Each time you need to open a new service to the outside world, you have to update the **Firewall** configuration and the LB as well

public LB

The **public LB** has to balance incoming traffic based on the external IP using a **source hash** algorithm.
The **flow manager** has to match this traffic before the standard rule (which matches everything).
This is basically all the traffic with destination port TCP 80 and the desired IP as the destination IP (192.168.116.11 in our case):

```
flow f_fw_in  director d_fw_in
  match iface eth0 dst 192.168.116.11 proto tcp dstport 80

flow f_firewall director d_firewall
  match iface eth0
```

Below, the Load-Balancing configuration corresponding:

```
# source ip hash for inbound NAT
director d_fw_in
  balance source
  mode gateway
  check interval 10 timeout 2
  option arpcheck
  server firewall1 192.168.15.251 weight 10 check
  server firewall2 192.168.15.252 weight 10 check

director d_firewall
  balance srcdst
  mode gateway
  check interval 10 timeout 2
  option arpcheck
  server firewall1 192.168.15.251 weight 10 check
  server firewall2 192.168.15.252 weight 10 check
```

We can rely only on the **source** address, since the server address (destination) will be NATed by the FW.

dmz LB

The **dmz LB** has to balance outgoing traffic from the internal web server **external clients**.
Hence, here is the **flow manager** configuration describing it:

```
flow f_fw_in  director d_fw_in
  match iface eth0 src 192.168.115.11 proto tcp srcport 80

flow f_firewall director d_firewall
  match iface eth0
```

Below, the Load-Balancing configuration corresponding:

```
# destination ip hash for inbound NAT
director d_fw_in
  balance dest
  mode gateway
  check interval 10 timeout 2
  option arpcheck
  server firewall1 192.168.115.251 weight 10 check
  server firewall2 192.168.115.252 weight 10 check

director d_firewall
  balance srcdst
  mode gateway
  check interval 10 timeout 2
  option arpcheck
  server firewall1 192.168.115.251 weight 10 check
  server firewall2 192.168.115.252 weight 10 check
```

We can rely only on the **destination** address, since the server address (source) will be NATed by the FW.

Both Inbound and Outbound NAT

Of course, both outbound and inbound NAT can be performed at the mean time, but remember a "full NAT" (NATing both source and destination of a single flow) is easily doable.
To do both at the same time, simply append **IN** and **OUT** rules BEFORE the generic rule at the end.