How do I filter traffic using DNS Lookup in NetScaler ADC Load Balancer?

Use Case

Use domain name in place of source or destination IP address to filter IP traffic and secure your network from unauthorized access.

Introduction

Access Control List (ACL) is the classic way of adding basic network security that acts as a firewall for controlling traffic in and out of a subnet. An ACL rule may take source IP address, destination IP address, destination port, protocol, and an associated action like allow or deny. In NetScaler ADC load balancer, following actions are supported –

- ALLOW—Process the packet.
- BRIDGE—Bridge the packet to the destination without processing it.
- DENY—Drop the packet.

While ACL works by processing the IP addresses, in today's Layer 7 networks, there are cases where the IP is not fixed and not known in advance. Today we depend more on "domain name" or FQDN to identify a source or target which can translate to dynamically generated IP address.

Hence today's requirement is to add ACL like filtering which can take fully qualified domain name (FQDN) as input and take allow or deny action over it. In NetScaler, this is possible using responder policy. So Responder module becomes your L7 ACL which can take ACL like actions by doing DNS resolution.

In NetScaler, responder policy can be bound to a virtual server or can be bound globally. When it is bound globally, all incoming traffic are evaluated against the rule of this policy. If it evaluates true then respective action is taken. Here is an example of denying access to network resources based on domain name achieved through Responder module.

Configuration Steps in NetScaler ADC

Step 1: Add DNS Name Server

Please see how o add a Name Server in NetScaler in this article - http://support.citrix.com/article/CTX109556.

Step 2: Add Responder Policy

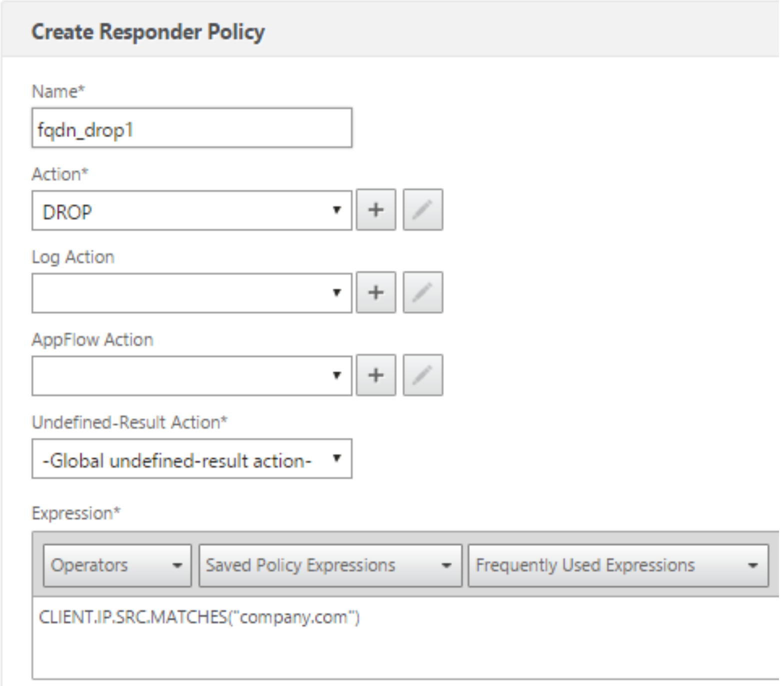
The FQDN added in this step gets resolved by Name Server added in previous step. So effectively, the FQDN resolves to an IP address and the rule matches incoming client IP address to that IP address and takes a decision.

CLI:

```
> add responder policy fqdn_drop1 "CLIENT.IP.SRC.MATCHES(\"<FQDN>\")" DROP
```

GUI:

In NetScaler GUI, go to Configuration -> AppExpert -> Responder -> Policies.



Create Responder Policy

Name*: fqdn_drop1

Action*: DROP

Log Action:

AppFlow Action:

Undefined-Result Action*: -Global undefined-result action-

Expression*:

```
CLIENT.IP.SRC.MATCHES("company.com")
```

Step 3: Binding Policy Globally

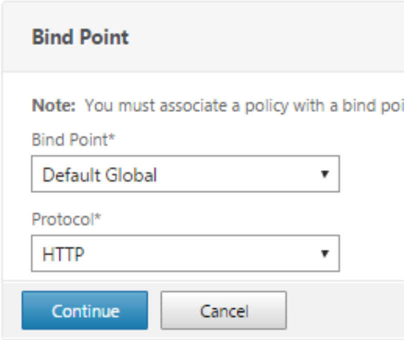
Responder policy shall be bound globally to be effective for all the inbound traffic in NetScaler. You can also bind it to a virtual server and the rules will be applicable only when request is sent to the virtual server IP. Here, binding globally is displayed.

CLI:

```
> bind responder global fqdn_drop1 100
 Done
```

GUI:

Go to Configuration -> AppExpert -> Responder -> Policies -> Policy Manager. Select the bind point as "Default Global" or "Override Global" as you want this policy to be bound globally. Select the protocol over which traffic will be coming.



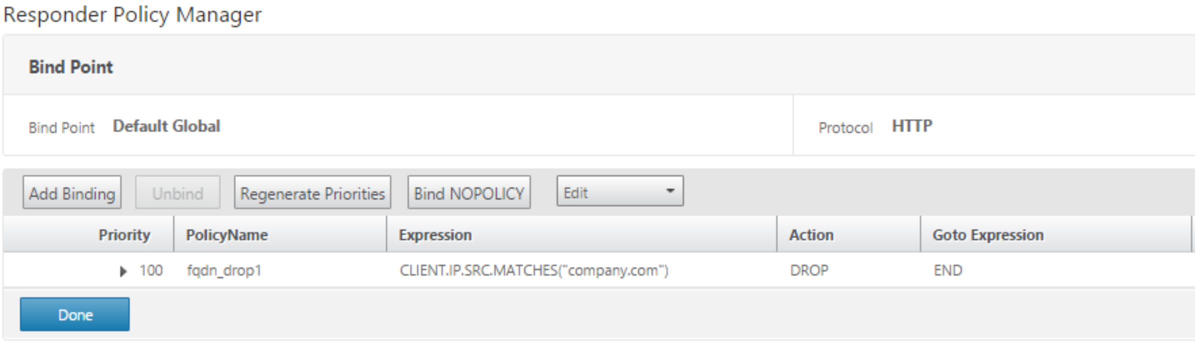
Bind Point

Note: You must associate a policy with a bind poi

Bind Point*: Default Global

Protocol*: HTTP

Then go to “Add Binding”, select the policy from the list and bind specifying the priority number.



Responder Policy Manager

Bind Point

Bind Point Default Global

Protocol HTTP

Priority	PolicyName	Expression	Action	Goto Expression
100	fqdn_drop1	CLIENT.IP.SRC.MATCHES("company.com")	DROP	END