IPv6 Security Lab - RTBH

Exploring Remotely Triggered Black Hole Filtering

Background

RTBH, as it is known as, is a commonly used technique to assist with the mitigation of Distributed Denial of Service Attacks. Remotely triggered blackhole filtering is a technique that provides the ability to drop undesirable traffic at the ingress into the network. RTBH provides a method for quickly dropping undesirable traffic at the edge of the network, based on either source addresses or destination addresses by forwarding it to a null0 interface. A typical deployment scenario for RTBH would require running internal Border Gateway Protocol (iBGP) at the access and aggregation points and configuring a separate device in the network operations centre (NOC) to act as a trigger. For destination-based drops, the triggering device sends iBGP updates to the edge that sets the next-hop of the victim's IP address to the null0 interface. Source-based drops are similar but it relies on the pre-existing deployment of uRPF which drops a packet if its source is "invalid"; invalid includes routes to Null0. Using the same mechanism for destination-based drops, a BGP update is sent, and this update sets the next hop for a source to Null0. Now all traffic entering an interface with uRPF enabled drops traffic from that source.

Demonstrating the use of RTBH

This exercise will demonstrate how to configure RTBH, and then will demonstrate its use in dealing with undesired traffic targeted at a particular destination.

To make this work, the team running AS10 will "attack" a target in AS20. The team running AS20 will "attack" a target in AS30. The team running AS30 will "attack" a target in AS40. The team running AS40 will "attack" a target in AS50. The team running AS50 will "attack" a target in AS60. And the team running AS60 will "attack" a target in AS10.

Configuring a Null Route

On all routers in the autonomous system configure a static route for the host address 0100::1 pointing to the null0 interface. (0100::/64 is listed in the IANA registry as a Discard Prefix – it is not routed on the Internet.)

```
ipv6 route 100::1/128 null0
```

Trigger Router - Step 1

We will now select one router in your autonomous system to be the trigger router. The trigger router will be the Access Router in each AS (not a normal function for a network operator's access router, but one we will use here in the lab).

The trigger router will announce the address which we want the whole AS to block traffic to. On this router we will configure a route-map called v6blackhole-trigger which will set policy to announce a specifically identified prefix with next-hop pointing to a null interface.

Note that we set community to be *no-export* (which means the prefix will not be announced to any other AS) and we set community to <ASN>:666 to identify this route for our internal policy as being a blackhole route (standard industry practice). Here is an example - replace X to make your correct AS number:

```
route-map v6blackhole-trigger permit 10
 description Look for blackholed routes
 match tag 66
 set ipv6 next-hop 100::1
 set local-preference 200
 set origin igp
 set community no-export
 set community X0:666
!
route-map v6blackhole-trigger deny 20
 description deny everything else – default
!
```

Trigger Router - Step 2

With the route-map now configured on the trigger router, we now add the route-map into the BGP process on the trigger router so that destinations we want to block will be distributed around our AS by iBGP. For example:

```
router bgp X0
 address-family ipv6
  redistribute static route-map v6blackhole-trigger
```

Which will redistribute all static routes configured on the router through the route-map black-hole-trigger. Any static routes matching the conditions in the route-map will have their next-hops set to 100::1 address.

Verify that the iBGP and eBGP sessions in the AS all have “send-community” configuration enabled – Cisco IOS does not send communities by BGP by default so it will have to be activated if not already configured. Do this for all peer-groups and other BGP peerings. For example:

```
router bgp X0
 address-family ipv6
  neighbor ibgp-partial send-community
```

Trigger Router - Step 3

If the trigger router uses iBGP with neighbour routers, it cannot be configured with “next-hop-self”. This is because Cisco IOS will replace the next-hop installed by the redistribute statement’s route-map with the IPv6 address of the local router. Which is no good – we want the next hop to be specifically 100::1 for the destination we want to blackhole. This is normally not a problem, as the Trigger Router usually resides in the NOC and won’t introduce any prefixes into the network apart from the ones which have to be blackholed. However, it is a problem here, as the routers we are using will all learn routes from their eBGP neighbours.

This means for the Trigger Router we have to remove the *next-hop-self* statements we added to the iBGP at the start of the workshop, and replace them with a route-map which will do the same task, but not change the next-hop of the blackholed routes. So on the Access Router, which we are using for our trigger, we need to do this:

```
router bgp X0
 address-family ipv6
  no neighbor ibgp-partial next-hop-self
```

Now we need to create a route-map to replace the *next-hop-self* statement we just removed, and then apply it to the iBGP peer-group. The route-map will test for blackhole routes (they have community <ASN>:666 set), and not do anything with them. It will then set the next-hop for all the remaining routes to the loopback address of the local router. Here is an example for the Access Router:

```
ip community-list 66 permit X0:666
!
route-map v6nhs permit 10
 match community 66
!
route-map v6nhs permit 20
 set ipv6 next-hop 2001:DB8:X0::3
!
router bgp 30
 address-family ipv6
  neigh ibgp-partial route-map v6nhs out
 !
```

Once the route-map has been applied to the iBGP neighbours, the iBGP sessions should be refreshed outbound – example for AS30:

```
clear bgp ipv6 unicast 30 out
```

And then you should see prefixes learned from the trigger router with appropriate next-hops set. The blackhole routes will have 100::1 as their next-hop, as you will see in the next step.

Testing the RTBH set up - Part 1

The groundwork for the RTBH is now in place. This will let us use the trigger router as a remote triggering device for dropping packets based on destination address in your network. All we need to do to have our AS null route traffic to a particular destination is to create a static route to this destination with a tag of 66 – this tag is matched by the route-map resulting in the prefix having its next hop address set to 100::1.

Now that the groundwork is in place, let us test our RTBH set up. To do this we enlist the support of one of our neighbour AS teams. As mentioned earlier, AS20 will ask AS10 to attack their network; AS30 will ask AS20 to attack their network; and so on. Here are suggested target addresses to be used within each AS:

AS	Target
10	2001:DB8:10::FF
20	2001:DB8:20::FF
30	2001:DB8:30::FF
40	2001:DB8:40::FF

AS	Target
50	2001:DB8:50::FF
60	2001:DB8:60::FF

On the respective trigger routers, create a static route to null for the above host addresses, tagging the null route with the number 66. Here is an example - replace X to make your AS number:

```
ipv6 route 2001:DB8:X0::FF/128 null0 tag 66
```

Once the null route is in place, verify the entry in the router's routing table. You should see the route to null for the selected trigger address.

Testing the RTBH set up - Part 2

Now the trigger router has been prepared, check the other routers in the same AS to see what the BGP table and Routing table entries are for the route we want to test RTBH for. Verify that the next hop address for selected trigger address on the router is 100::1 and that you have a routing table entry for 100::1 pointing to the Null interface. If not, please check with your team members in your AS.

Testing the RTBH set up - Part 3

Now ask your "attacker" AS to send a stream of ping packets to the destination address you are using as the target. The simplest way to do it is if each router in the neighbour AS sets up an extended ping, for example as below (where C1 in AS10 is pinging the target address in AS20):

```
C1#ping ipv6
Target IPv6 address: 2001:DB8:20::FF
Repeat count [5]: 10000
Datagram size [100]:
Timeout in seconds [2]:
Extended commands? [no]:
Sweep range of sizes? [no]:
Type escape sequence to abort.
Sending 10000, 100-byte ICMP Echos to 2001:DB8:20::FF, timeout is 2 seconds:
```

Check on the external interfaces to your neighbouring AS (the one which is "attacking" your target address). Do you see the interface counters going up showing there is traffic? Also check the Netflow you configured on your external facing interfaces for the previous exercise – do you see the ICMPv6 packets heading in to the "attacked" destination?

Now check on the trigger router? Do you see any traffic inbound to it from any of the iBGP neighbours?

Summary

If the previous step demonstrates success in "defeating" the "attack", then you will have successfully configured RTBH for your AS. This is a common tool used by many network operators globally, and is considered mandatory by many end-users when searching for connectivity options from their upstream providers.