Network Working Group
Request for Comments: 5635
Category: Informational

W. Kumari
Google
D. McPherson
Arbor Networks

August 2009

Remote Triggered Black Hole Filtering with Unicast Reverse Path Forwarding (uRPF)

Abstract

Remote Triggered Black Hole (RTBH) filtering is a popular and effective technique for the mitigation of denial-of-service attacks. This document expands upon destination-based RTBH filtering by outlining a method to enable filtering by source address as well.

Status of This Memo

This memo provides information for the Internet community. It does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

Table of Contents

1. Introduction

This document expands upon the technique outlined in "Configuring BGP to Block Denial-of-Service Attacks" [RFC3882] to demonstrate a method that allows for filtering by source address(es).

Network operators have developed a variety of techniques for mitigating denial-of-service (DoS) attacks. While different techniques have varying strengths and weaknesses, from an implementation perspective, the selection of which method to use for each type of attack involves evaluating the tradeoffs associated with each method.

A common DoS attack directed against a customer of a service provider involves generating a greater volume of attack traffic destined for the target than will fit down the links from the service provider(s) to the victim (customer). This traffic "starves out" legitimate traffic and often results in collateral damage or negative effects to other customers or the network infrastructure as well. Rather than having all destinations on their network be affected by the attack, the customer may ask their service provider to filter traffic destined to the target destination IP address(es), or the service provider may determine that this is necessary themselves, in order to preserve network availability.

One method that the service provider can use to implement this filtering is to deploy access control lists on the edge of their network. While this technique provides a large amount of flexibility in the filtering, it runs into scalability issues, both in terms of the number of entries in the filter and the packet rate.

Most routers are able to forward traffic at a much higher rate than they are able to filter, and they are able to hold many more forwarding table entries and routes than filter entries. RTBH filtering leverages the forwarding performance of modern routers to filter more entries and at a higher rate than access control lists would otherwise allow.

However, with destination-based RTBH filtering, the impact of the attack on the target is complete. That is, destination-based RTBH filtering injects a discard route into the forwarding table for the target prefix. All packets towards that destination, attack traffic AND legitimate traffic, are then dropped by the participating routers,thereby taking the target completely offline. The benefit is that collateral damage to other systems or network availability at the customer location or in the ISP network is limited, but the negative impact to the target itself is arguably increased.

By coupling unicast Reverse Path Forwarding (uRPF) [RFC3704] techniques with RTBH filtering, BGP can be used to distribute discard routes that are based not on destination or target addresses, but on source addresses of unwanted traffic. Note that this will drop all traffic to/from the address, and not just the traffic to the victim.

This document is broken up into three logical parts: the first outlines how to configure destination-based RTBH, the second covers source-based RTBH, and the third part has examples and a history of the technique.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

3. Destination Address RTBH Filtering

3.1. Overview

A discard route is installed on each edge router in the network with the destination set to the discard (or null) interface. In order to use RTBH filtering for a single IP address (or prefix), a BGP route for the address to be filtered is announced, with the next-hop set as

the discard route. This causes traffic to the announced network prefix to be forwarded to the discard interface so that it does not transit the network wasting resources or triggering collateral damage to other resources along the path towards the target.

While this does "complete" the attack in that the target address(es) are made unreachable, collateral damage is minimized. It may also be possible to move the host or service on the target IP address(es) to another address and keep the service up, for example, by updating associated DNS resource records.

3.2. Detail

Before deploying RTBH filtering, there is some preparation and planning that needs to occur and decisions that need to be made. These include:

- What are the discard addresses?

- What are the discard BGP communities?

- What is the largest prefix that can be black-holed?

- What is the smallest advertisement that your provider will accept?

Steps to configure destination-based RTBH filtering:

Step 1. Select Your Discard Address Schema

An address is chosen to become the "discard address". This is often chosen from 192.0.2.0/24 (TEST-NET [RFC3330]), or from RFC 1918 [RFC1918] space. Multiple addresses allow an operator to configure multiple static routes, one for each incident:

```
192.0.2.1 = Incident #1
192.0.2.2 = Incident #2
192.0.2.3 = Incident #3
192.0.2.4 = Incident #4
192.0.2.5 = Incident #5
```

Customer #1, who has a DDoS (Distributed DoS) attack can be pointed to discard route 192.0.2.1. Customer #2 can be pointed to discard route 192.0.2.2. If capable, the router can then count the drops for each, providing some level of telemetry on the volume of drops as well as status of an ongoing attack. A consistent address schema facilitates operations.

Step 2. Configure the Discard Route(s) on Each Router

A route for the "discard address" is installed on the routers that form the edge/perimeter of the network in all routers in the network or some subset (e.g., peering, but not customer, etc.). The destination of the route is set to the "discard" or "null" interface. This route is called the "discard route". Implementation experience demonstrates the value of configuring each ingress router with a capability for dropping traffic via RTBH filtering.

Step 3. Configure the RTBH BGP Policy on Each Router

A BGP policy is configured on all routers that have the discard route so that routes announced with a chosen community will have their next-hop set to the discard address. The BGP policy should be made restrictive so that only BGP routes covering a defined number of hosts addresses will be accepted. That is, typically, only specific /32s are necessary. Shorter prefix blocks may also be required or desirable, for example, if larger numbers of attack targets are located within a single prefix, or the employment of this mechanism is to drop traffic bound for specific networks. When filtering based on shorter prefixes, extreme caution should be used as to avoid collateral damage to other hosts that reside within those address blocks. Full implementations will have multiple communities, with each community used for different parts of a provider’s network and for different security incidents.

Step 4. Configure the Safety Egress Prefix Filters

There is always a chance that the triggering BGP update could leak from the network and so egress prefix filters are strongly recommended. These egress prefix filter details may vary, but experience has demonstrated that the following works:

- Deny all prefixes longer than the longest prefix that you expect to announce. For example, if the longest prefix that you expect to announce is /24, deny all prefixes of length /25 though /32. If your triggering BGP update is only /32s, then this egress prefix filter will add a safe measure in case the NO_EXPORT community does not work.

- Deny all communities used for triggering RTBH filtering. This is also a "safety" measure in case the NO_EXPORT community does not work.

Step 5: Configure the Trigger Router

Configure the trigger router, workstation, or other device so that adding and removing the triggers can be done easily and quickly. The BGP update should have the NO_EXPORT community as a mandatory attribute. An egress prefix filter or policy that prevents RTBH filtering prefixes in the /8 to /24 range is also recommended as a safety tool. The trigger router can be set up as an iBGP (Internal BGP) route reflector client that does not receive any prefixes from its BGP peer. This allows a low-cost router/workstation to be used as the trigger router.

Using the RTBH filtering:

1: When RTBH filtering is desired for a specific address, that address is announced from a trigger router (or route server), tagged with the chosen "RTBH" community and with the NO_EXPORT community, and passed via iBGP. The receiving routers check the BGP policy, set the next-hop to be the discard route, resulting in a Forwarding Information Base (FIB) entry pointing to a discard address.

2: Traffic entering the network will now be forwarded to the discard interface on all edge routers and will therefore be dropped at the edge of the network, saving resources.

2.1: Multiple Discard Addresses for Incident Granularity. This technique can be expanded by having multiple discard addresses, routes, and communities to allow for monitoring of the discarded traffic volume on devices that support multiple discard interfaces. As mentioned earlier, each router can have a discard address schema to allow the operator to distinguish multiple incidents from each other -- making it easier to monitor the life-cycle of the incidents.

2.2: Multiple "Trigger Communities" for Network-Wide Granularity. The network can be sectioned into multiple communities, providing the operator with an ability to drop in discrete parts of their network. For example, the network can be divided into the following communities (where XXX represents the operator's AS number):

```
XXX:600 RTBH filtering on all routers
XXX:601 RTBH filtering on only peering routers
XXX:602 RTBH filtering on only customers who peer BGP
XXX:603 RTBH filtering on data centers (to see if the
        data center is the source of attack)
```

XXX:604 RTBH filtering on all customers (to see how many customers are being used by the attacker)

Some diligent thinking is required to develop a community schema that provides flexibility while reflecting topological considerations.

2.3: "Customer-Triggered" RTBH filtering. The technique can also be expanded by relaxing the Autonomous System (AS) path rule to allow customers of a service provider to enable RTBH filtering without interacting with the service provider's trigger routers. If this is configured, an operator MUST only accept announcements from the customer for prefixes that the customer is authorized to advertise, in order to prevent the customer from accidentally (or intentionally) black-holing space that they are not allowed to advertise.

A common policy for this type of setup would first permit any longer prefix within an authorized prefix only if the black hole communities are attached, append NO_EXPORT, NO_ADVERTISE, or similar communities, and then also accept from the customer the original aggregate prefix that will be advertised as standard policy permits.

Extreme caution should be used in order to avoid leaking any more specifics beyond the local routing domain, unless policy explicitly aims at doing just that.

4. Source Address RTBH Filtering

In many instances, denial-of-service attacks sourced from botnets are being configured to "follow DNS". (The attacking machines are instructed to attack www.example.com, and re-resolve this periodically. Historically, the attacks were aimed simply at an IP address and so renumbering www.example.com to a new address was an effective mitigation.) This makes it desirable to employ a technique that allows black-holing to be based on source address.

By combining traditional RTBH filtering with unicast Reverse Path Forwarding (uRPF), a network operator can filter based upon the source address. uRPF performs a route lookup of the source address of the packet and checks to see if the ingress interface of the packet is a valid egress interface for the packet source address (strict mode) or if any route to the source address of the packet exists (loose mode). If the check fails, the packet is typically dropped. In loose mode, some vendors also verify that the destination route does not point to an invalid next-hop -- this allows source-based RTBH filtering to be deployed in networks that

cannot implement strict (or feasible path) mode uRPF. Before enabling uRPF (in any mode), it is vital that you fully understand the implications of doing so:

- Strict mode will cause the router to drop all ingress traffic if the best path back to the source address of the traffic is not the interface from which the traffic was received. Asymmetric routing will cause strict mode uRPF to drop legitimate traffic.

- Loose mode causes the router to check if a route for the source address of the traffic exists. This may also cause legitimate traffic to be discarded.

It is hoped that in the future, vendors will implement a "DoS-mitigation" mode in addition to the loose and strict modes -- in this mode, the uRPF check will only fail if the next-hop for the source of the packet is a discard interface.

By enabling the uRPF feature on interfaces at predetermined points in their network and announcing the source address(es) of attack traffic, a network operator can effectively drop the identified attack traffic at specified devices (ideally ingress edge) of their network based on source address.

While administrators may choose to drop traffic from any prefix they wish, it is recommended when employing source-based RTBH filtering inter-domain that explicit policy be defined that enables peers to only announce source-based RTBH routes for prefixes that they originate.

4.1. Steps to Deploy RTBH Filtering with uRPF for Source Filtering

The same steps that are required to implement destination address RTBH filtering are taken with the additional step of enabling unicast Reverse Path Forwarding on predetermined interfaces. When a source address (or network) needs to be blocked, that address (or network) is announced using BGP tagged with a community. This will cause the route to be installed with a next-hop of the discard interface, causing the uRPF check to fail and the packets to be discarded. The destination-based RTBH filtering community should not be used for source-based RTBH filtering, and the routes tagged with the selected community should be carefully filtered.

The BGP policy will need to be relaxed to accept announcements tagged with this community to be accepted even though they contain addresses not controlled by the network announcing them. These announcements must NOT be propagated outside the local AS and should carry the NO_EXPORT community.

As a matter of policy, operators SHOULD NOT accept source-based RTBH announcements from their peers or customers, they should only be installed by local or attack management systems within their administrative domain.

5. Security Considerations

The techniques presented here provide enough power to cause significant traffic forwarding loss if incorrectly deployed. It is imperative that the announcements that trigger the black-holing are carefully checked and that the BGP policy that accepts these announcements is implemented in such a manner that the announcements:

- Are not propagated outside the AS (NO_EXPORT).

- Are not accepted from outside the AS (except from customers).

- Except where source-based filtering is deployed, that the network contained in the announcement falls within the address ranges controlled by the announcing AS (i.e., for customers that the address falls within their space).

6. Acknowledgments

I would like to thank Joe Abley, Ron Bonica, Rodney Dunn, Alfred Hoenes, Donald Smith, Joel Jaeggli, and Steve Williams for their assistance, feedback and not laughing *too* much at the quality of the initial versions.

I would also like to thank all of the regular contributors to the OPSEC Working Group and Google for 20% time :-)

The authors would also like to thank Steven L Johnson and Barry Greene for getting this implemented and Chris Morrow for publicizing the technique in multiple talks.

7. References

7.1. Normative References

[RFC1918] Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, February 1996.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3330] IANA, "Special-Use IPv4 Addresses", RFC 3330, September 2002.

[RFC3704] Baker, F. and P. Savola, "Ingress Filtering for Multihomed Networks", BCP 84, RFC 3704, March 2004.

[RFC3882] Turk, D., "Configuring BGP to Block Denial-of-Service Attacks", RFC 3882, September 2004.

7.2. Informative References

[Greene2001] Greene Barry Raveendran and Jarvis Neil, "Unicast Reverse Path Forwarding (uRPF) Enhancements for the ISP-ISP Edge", ftp://ftp-eng.cisco.com/cons/isp/documents/uRPF_Enhancement.pdf, 2001.

Appendix A. Cisco Router Configuration Sample

This section provides a partial configuration for configuring RTBH filtering on a Cisco router. This is not a complete configuration and should be customized before being used.

Announcing router:

```
! The discard route
ip route 192.0.2.1 255.255.255.255 Null0
!
! Matches and empty AS-PATH only.
ip as-path access-list 10 permit ^$
!
! This route-map matches routes with tag 666 and sets the next-hop
! to be the discard route.
route-map remote-trigger-black-hole permit 10
 match tag 666
 set ip next-hop 192.0.2.1
 set local-preference 200
 set community no-export
 ! The community used internally to tag RTBH announcements.
 set community 65505:666
 set origin igp
!
route-map remote-trigger-black-hole permit 20
!
router bgp 65505
 no synchronization
 bgp log-neighbor-changes
 redistribute static route-map remote-trigger-black-hole
 no auto-summary
!
! An example IP that we are applying RTBH filtering to.
! All traffic destined to 10.0.0.1 will now be dropped!
ip route 10.0.0.1 255.255.255.255 null0 tag 666
!
```

Filtering router:

```
!
! The discard route
ip route 192.0.2.1 255.255.255.255 Null0
!
! Matches and empty AS-PATH only.
ip as-path access-list 10 permit ^$
!
route-map black-hole-filter permit 10
 match ip address prefix-list only-host-routes
 match as-path 10
```

```
 match community 65505:666 no-export
!
! Don't accept any other announcements with the RTBH community.
route-map black-hole-filter deny 20
 match community 65505:666
!
route-map black-hole-filter permit 30
!
! An interface for source-based RTBH filtering with uRPF loose mode.
interface FastEthernet 0/0
ip verify unicast source reachable-via any
```

Appendix B. Juniper Configuration Sample

This section provides a partial configuration for configuring RTBH filtering on a Juniper router. This is not a complete configuration and should be customized before being used.

Announcing router:

```
routing-options {
   static {
       /* EXAMPLE ATTACK SOURCE */
       route 10.11.12.66/32 {
           next-hop 192.0.2.1;
           resolve;
           tag 666;
       }
       /* EXAMPLE ATTACK DESTINATION */
       route 10.128.0.2/32 {
           next-hop 192.0.2.1;
           resolve;
           tag 666;
       }
   }
   autonomous-system 100;
}

protocols {
   bgp {
       group ibgp {
           type internal;
           export rtbh;
           neighbor 172.16.0.2;
       }
   }
}
```

```
      policy-options {
         policy-statement rtbh {
             term black-hole-filter {
                 from {
                     tag 666;
                     route-filter 0.0.0.0/0 prefix-length-range /32-/32;
                 }
                 then {
                     local-preference 200;
                     origin igp;
                     community set black-hole;
                     community add no-export;
                     next-hop 192.0.2.1;
                     accept;
                 }
             }
         }
         community black-hole members 100:666;
         community no-export members no-export;
      }
```

Filtering router:

```
      policy-statement black-hole-filter {
         from {
             protocol bgp;
             as-path LocalOnly;
             community black-hole;
             route-filter 0.0.0.0/0 prefix-length-range /32-/32;
         }
         then {
             community set no-export;
             next-hop 192.0.2.1;
         }
      }
      community black-hole members 100:666;
      community no-export members no-export;


      routing-options {
         forwarding-table {
             unicast-reverse-path feasible-paths;
         }
         static {
             route 192.0.2.1/32 discard;
         }
      }
```

```
interfaces {
   xe-1/0/0 {
       vlan-tagging;
       mtu 9192;
       unit 201 {
           vlan-id 201;
           family inet {
               rpf-check;
               address 10.11.12.1/24;
           }
       }
   }
}
```

Appendix C. A Brief History of RTBH Filtering

Understanding the history and motivation behind the development of a technique often helps with understanding how to best utilize the technique. In this spirit, we present a history of unicast RPF and RTBH filtering.

This section has been provided by Barry Raveendran Greene:

Unicast RPF Loose Check (uRPF Loose Check) was created by Neil Jarvis and Barry Greene to be used with destination-based RTBH as a rapid reaction tool to DDoS attacks. The requirements for this rapid reaction tool was based on post mortem conversation after the February 2000 attacks on several big content hosting companies. The summary of the requirement became the "Exodus Requirement" which stated:

> We need a tool to drop packets based on source IP address that can be pushed out to over 60 routers within 60 seconds, be longer than a thousand lines, be modified on the fly, and work in all your platforms filtering at line rate.

A variety of options were looked at to meet this requirement, from reviving Common Open Policy Service (COPS), to pushing out Access Control Lists (ACLs) with BGP, creating a new protocol. In 2000, the quickest way to meet the "Exodus requirement" was to marry two functions. First, modify unicast RPF so that the interface check was no longer required and to make sure that a "null" or discard route would drop the packet (i.e., loose check). Second, the technique where BGP is used to trigger a distributed drop is dusted off and documented. Combining these two techniques was deemed a fast way to put a distributed capability to drop packets out into the industry.

To clarify and restate, uRPF loose check was created as one part of a rapid reaction tool to DDoS attacks that "drop packets based on source IP address that can be pushed out to over 60 routers with in 60 seconds, be longer than a thousand lines, be modified on the fly, and work in all your platforms filtering at line rate". The secondary benefits of using uRPF Loose Check for other functions is a secondary benefit -- not the primary goal for its creation.

To facilitate the adoption to the industry, uRPF Loose Check was not patented. It was publicly published and disclosed in "Unicast Reverse PathForwarding (uRPF) Enhancements for the ISP-ISP Edge" [Greene2001].

Authors' Addresses

Warren Kumari
Google
1600 Amphitheatre Parkway
Mountain View, CA 94043

EMail: warren@kumari.net

Danny McPherson
Arbor Networks, Inc.

EMail: danny@arbor.net