Implementing BCP38 with uRPF on Junipers

Scot Colburn for Pete Siemsen, FRGP
colburn@ucar.edu
October 15th, 2018
BCP38 DSDDoSaaS Denying the Service of Distributed Denial of Service as a Service!

Source-address Assurance (SAA) Check
Reverse-path Forwarding (RPF)
What spoofing is

For purposes of this talk you are an ISP or regional. You are the good guy.

You provide service for your customers.

Suppose your customers have some bad guys.

They create packets with fake source addresses. Or, malware-infected PCs do it. Or, their network just leaks bogus packets.

You forward them to the Internet. The Internet suffers (think DDoS attacks). With fake source addresses, those bad guys are hard to find.

You should try to defend the Internet from the bad guys.
What is BCP38?

**Best Current Practice #38** is: “Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing”

Ensure that you forward only “good” packets from your customers.

Good packets have source addresses inside your customers’ IP blocks.

You should block all other packets.
Why doing it with manual router filters is tedious

You could install packet filters.

For each customer, the packet filter will list the same blocks as your existing BGP filter. You do have BGP filters, right? Hello?

When the customer changes their IP blocks (it could happen) you’ll have to update two places in your router.

Two similar, repetitive, redundant lists of the same thing, with the same stuff.

Smells bad. Easy to screw up.
How uRPF helps automate the filters

Unicast Reverse Path Forwarding does filtering using the routing information.

URPF performs a route table lookup on an IP packet’s source address, and checks the incoming interface.

If that interface is feasibly one the router could use to reach the source address, it’s a “good” packet. If not, we can take extreme measures...
Juniper's implementation of uRPF

The router can log and/or block “bad” packets.

You can try before you buy: log bad packets for a while, then log-and-block, then block only.

It’s implemented per-interface. You choose which customers go first.

It’s implemented like other packet filters, so you can install terms to implement exceptions if needed.
Challenges with uRPF

Routing asymmetry - multiple interfaces to the same destination.

Solution: use “feasible-path” on Junipers.

We don’t do uRPF to *our* provider connections - too many routes to consider.
Juniper config lines - 1. interfaces

interfaces {
    xe-0/0/0 {
        unit 893 {
            apply-groups input-packet-filter
            vlan-id 893
            family inet {
                rpf-check fail-filter rpf-fail-discard
                address 129.19.64.122/30
            }
        }
    }
}
filter rpf-fail-discard {
    term whitelist-sources {
        from {
            source-prefix-list {
                local-subnets-v4;
                rpf-whitelist-sources;
            }
        }
        then accept;
    }
    term final-actions {
        then {
            count rpf-fail;
            syslog;
            discard;
        }
    }
}
Juniper config lines - 3. use feasible path

```plaintext
routing-options {
    forwarding-table {
        unicast-reverse-path feasible-paths;
    }
}
```
Typical log lines

May 29 07:47:52 192.43.217.144 fpc2 PFE_FW_SYSLOG_IP: FW: xe-2/0/4.963 D udp 10.119.132.228 50.7.124.48 21388 10001 (3 packets)

May 29 07:47:52 192.43.217.144 fpc0 PFE_FW_SYSLOG_IP: FW: xe-0/1/3.878 D tcp 47.8.19.207 172.18.4.209 63479 21 (1 packets)
How the FRGP presented the idea to customers

The FRGP wants to be a good Internet citizen, right?

There’s no legitimate reason for you to emit bad packets, right?

First, we’ll show you your bad packets, for several months.

You can quietly take care of your dirty laundry.

On a flag day, instead of logging-and-forwarding, we’ll start blocking.

Lots of warning time, plenty of announcements.
How the FRGP deployed uRPF gently

Wrote a Perl program.

Reads Juniper syslog files. Writes HTML files (updates a website).

Shows customers statistics about their bad packets.

Correlates, summarizes, simplifies logs.

Shows src/dest address and port. Customers can track them down and kill them.
more - How the FRGP deployed uRPF gently

Perl program runs once a day.

Hooked to logrotate: syslogs rotated, then 24-hour log is processed.

Replaces web pages every day.

Simple - no history deeper than one day.
Unicast Reverse Path Forwarding check logs

RPF filters will be changed to block packets according to the FRGP RPF change schedule.

This web page shows log data gathered by the FRGP routers about non-compliant packets. Non-compliant packets are packets received from an FRGP participant that contain source addresses that don't match the participant's IP range(s).

<table>
<thead>
<tr>
<th>What</th>
<th>Router</th>
<th>Interface</th>
<th>non-compliant packets dropped</th>
<th>forwarded</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCCS global</td>
<td>core-13</td>
<td>xe-2/0/4.963</td>
<td>1507334</td>
<td>0</td>
</tr>
<tr>
<td>CCCS research</td>
<td>core-13</td>
<td>xe-2/0/4.964</td>
<td>59404</td>
<td>0</td>
</tr>
<tr>
<td>CTN vrf:research</td>
<td>core-13</td>
<td>xe-0/0/0.981</td>
<td>1695</td>
<td>0</td>
</tr>
<tr>
<td>JWU</td>
<td>core-13</td>
<td>xe-0/0/0.1032</td>
<td>3000</td>
<td>0</td>
</tr>
<tr>
<td>JWU vrf:research</td>
<td>core-13</td>
<td>xe-0/0/0.1033</td>
<td>52</td>
<td>0</td>
</tr>
<tr>
<td>NEON Global II</td>
<td>core-13</td>
<td>ge-1/1/8.857</td>
<td>423</td>
<td>0</td>
</tr>
<tr>
<td>State:WY east</td>
<td>core-13</td>
<td>xe-0/1/3.886</td>
<td>6360</td>
<td>0</td>
</tr>
<tr>
<td>UIS-Hosting</td>
<td>core-13</td>
<td>xe-0/0/0.893</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>UW east</td>
<td>core-13</td>
<td>xe-0/1/3.878</td>
<td>1050</td>
<td>0</td>
</tr>
<tr>
<td>UW east vrf:research</td>
<td>core-13</td>
<td>xe-0/1/3.879</td>
<td>648</td>
<td>0</td>
</tr>
<tr>
<td><strong>totals</strong></td>
<td></td>
<td></td>
<td><strong>1950792</strong></td>
<td><strong>0</strong></td>
</tr>
</tbody>
</table>
Logs of non-compliant packets dropped by the FRGP router. These packets were seen inbound on interface xe-2/0/4.963 on router core-13. **1507334** packets were detected between **May 29 07:35:21** and **May 30 07:01:52**.

This table doesn't show them all. To limit the size of the table, it has been intentionally limited to show data about only the first 200 unique source/dest IP pairs from the log file.

<table>
<thead>
<tr>
<th>Packet count</th>
<th>Source</th>
<th>Destination</th>
<th>Source Port</th>
<th>Destination Port</th>
<th>Protocol</th>
<th>Date/Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>66719</td>
<td>10.119.132.233</td>
<td>50.7.78.226</td>
<td>19070</td>
<td>10001</td>
<td>udp</td>
<td>May 29 07:35:23 - May 30 07:01:52</td>
</tr>
<tr>
<td>66651</td>
<td>10.119.132.233</td>
<td>50.7.124.48</td>
<td>19070</td>
<td>10001</td>
<td>udp</td>
<td>May 29 07:35:24 - May 30 07:01:51</td>
</tr>
<tr>
<td>66272</td>
<td>10.119.132.233</td>
<td>198.16.70.106</td>
<td>19070</td>
<td>10001</td>
<td>udp</td>
<td>May 29 07:35:22 - May 30 07:01:52</td>
</tr>
<tr>
<td>65971</td>
<td>10.119.132.228</td>
<td>50.7.124.48</td>
<td>21388</td>
<td>10001</td>
<td>udp</td>
<td>May 29 07:35:30 - May 30 07:01:52</td>
</tr>
<tr>
<td>65931</td>
<td>10.119.132.228</td>
<td>178.162.208.234</td>
<td>21388</td>
<td>10001</td>
<td>udp</td>
<td>May 29 07:35:22 - May 30 07:01:48</td>
</tr>
<tr>
<td>65630</td>
<td>10.119.132.228</td>
<td>198.255.32.138</td>
<td>21388</td>
<td>10001</td>
<td>udp</td>
<td>May 29 07:35:25 - May 30 07:01:50</td>
</tr>
<tr>
<td>65529</td>
<td>10.119.132.233</td>
<td>168.1.83.89</td>
<td>19070</td>
<td>10001</td>
<td>udp</td>
<td>May 29 07:35:24 - May 30 07:01:49</td>
</tr>
<tr>
<td>64498</td>
<td>10.119.132.228</td>
<td>104.199.156.58</td>
<td>21388</td>
<td>10240</td>
<td>udp</td>
<td>May 29 07:35:25 - May 30 07:01:48</td>
</tr>
<tr>
<td>64333</td>
<td>10.119.132.228</td>
<td>198.255.6.34</td>
<td>21388</td>
<td>10001</td>
<td>udp</td>
<td>May 29 07:35:22 - May 30 07:01:52</td>
</tr>
<tr>
<td>63989</td>
<td>10.119.132.233</td>
<td>198.255.6.34</td>
<td>19070</td>
<td>10001</td>
<td>udp</td>
<td>May 29 07:35:24 - May 30 07:01:52</td>
</tr>
<tr>
<td>63389</td>
<td>10.119.132.233</td>
<td>58.7.114.59</td>
<td>38833</td>
<td>40001</td>
<td>udp</td>
<td>May 29 07:35:24 - May 30 07:01:51</td>
</tr>
</tbody>
</table>
Logs of non-compliant packets dropped by the FRGP router. These packets were seen inbound on interface xe-0/0/0.1032 on router core-13. **3000** packets were detected between **May 29 07:35:21** and **May 30 07:01:52**.

<table>
<thead>
<tr>
<th>Packet count</th>
<th>Source</th>
<th>Destination</th>
<th>Source Port</th>
<th>Destination Port</th>
<th>Protocol</th>
<th>Date/Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1407</td>
<td>10.208.254.133</td>
<td>192.168.121.2</td>
<td>11004</td>
<td>11004</td>
<td>udp</td>
<td>May 29 07:35:54 - May 30 07:01:49</td>
</tr>
<tr>
<td>58</td>
<td>10.208.254.86</td>
<td>23.46.48.24</td>
<td>various</td>
<td>80</td>
<td>tcp</td>
<td>May 29 14:34:51 - May 30 06:35:26</td>
</tr>
<tr>
<td>48</td>
<td>10.208.254.20</td>
<td>23.46.48.24</td>
<td>various</td>
<td>80</td>
<td>tcp</td>
<td>May 29 09:02:35 - May 30 04:03:13</td>
</tr>
</tbody>
</table>
more: How the FRGP deployed uRPF gently

Let customers chew on the gathered statistics for a few months.

Schedule “turning on blocking” flexibly. Tell them “Any problems, we’ll back out and regroup.”

Remember, you can turn on blocking one one interface at a time. You don’t have to do everyone at once.
Gotchas

You may run into issues with multiply-connected customers.

Some routers respond to pings using a source address from an interface other than the ingress interface. To fix, use a whitelist of approved IP addresses.
Questions?

Perl program available here:

www.frgp.net/generate-rpf-webpages.pl

siemens@ucar.edu