AS_PATH Verifcation
Using ASPA

Alexander Azimov aa@qrator.net, Eugene Bogomazov eb@qrator.net,
Eugene Uskov eu@qrator.net, Randy Bush randy@psg.com,
Job Snijders job@ntt.net, Keyur Patel keyur@arrcus.com,
Russ Housley housley@vigilsec.com
## BGP Quadrant

<table>
<thead>
<tr>
<th></th>
<th>BGP Hijacks</th>
<th>BGP Route Leaks</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Mistake</strong></td>
<td>IRR Filters; ROA;</td>
<td>IRR Filters; <strong>Route Leak Detection Draft</strong></td>
</tr>
<tr>
<td><strong>Malicious</strong></td>
<td><strong>BGPSec</strong></td>
<td><strong>BGPSec</strong></td>
</tr>
</tbody>
</table>

**BGPSec** refers to BGP Security, a protocol designed for enhancing the security of the BGP routing protocol.
To secure BGP, do we require *attacker* to support BGPSec?
<table>
<thead>
<tr>
<th>Mistake</th>
<th>BGP Hijacks</th>
<th>BGP Route Leaks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mistake</td>
<td>IRR Filters; ROA;</td>
<td>IRR Filters;</td>
</tr>
<tr>
<td>Malicious</td>
<td>![exclamation mark]</td>
<td>![exclamation mark] Route Leak Detection Draft</td>
</tr>
</tbody>
</table>
Goals

• Detect invalid AS_PATHs;
• Detect malformed AS_PATHs;
• Incremental Deployment;
• Lightweight
  • Do not add new message types in BGP;
  • Do not add signatures in BGP.
Anomaly Propagation
Anomaly Propagation

If we can stop propagation at the level of c2p and p2p – we are done!
A Beautiful Note

If valid route is received from customer or peer it MUST have only customer-to-provider pairs in its AS_PATH.

Then if we have a validated database of customer-to-provider pairs we will be able to verify routes received from customers and providers!
Autonomous System Provider Authorization

ASPA

ASPA := {
    customer_asn – signer
    provider_asn – authorized to send routes to upper providers or peers
    AFI – IPv4 or IPv6
}
Boundary Cases

• Transit-free networks;
  $\text{ASPA}_0 = \{\text{AS}_1, 0\}$;

• Complex relations
  Symmetric ASPAs: $\{\text{AS}_1, \text{AS}_2\}, \{\text{AS}_2, \text{AS}_1\}$;
Pair Verification (AS1, AS2)

1. Retrieve all cryptographically valid ASPAs in a selected AFI with a customer value of AS1. This selection forms the set of candidate ASPAs.

2. If the set of candidate ASPAs is empty, then the procedure exits with an outcome of unknown.

3. If there is at least one candidate ASPA where the provider field is AS2, then the procedure exits with an outcome of valid.

4. Otherwise, the procedure exits with an outcome of invalid.
AS_PATH Verification

1. If the closest AS in the AS_PATH is not the receiver's neighbor ASN then procedure halts with the outcome "invalid";
2. If in one of AS_SEQ segments there is a pair (AS(I-1), AS(I)) is "invalid" then the procedure also halts with the outcome "invalid";

ROA {x.x.x.x, AS1}
ASPA {AS1, AS2}
ASPA {AS2, AS3}
ASPA {AS3, 0}
AS_PATH Verification

1. If the closest AS in the AS_PATH is not the receiver's neighbor ASN then procedure halts with the outcome "invalid";

2. If in one of AS_SEQ segments there is a pair (AS(I-1), AS(I)) is "invalid" then the procedure also halts with the outcome "invalid";

Route: x.x.x.x
AS_PATH: AS4

ROA {x.x.x.x, AS1}
ASPA {AS1, AS2}
ASPA {AS2, AS3}
ASPA {AS3, 0}
AS_PATH Verification

1. If the closest AS in the AS_PATH is not the receiver's neighbor ASN then procedure halts with the outcome "invalid";

2. If in one of AS_SEQ segments there is a pair (AS(I-1), AS(I)) is "invalid" then the procedure also halts with the outcome "invalid";

ROA {x.x.x.x, AS1}
ASPA {AS1, AS2}
ASPA {AS2, AS3}
ASPA {AS3, 0}

Route: x.x.x.x
AS_PATH: AS4 AS1
AS_PATH Verification

1. If the closest AS in the AS_PATH is not the receiver's neighbor ASN then procedure halts with the outcome "invalid";

2. If in one of AS_SEQ segments there is a pair (AS(I-1), AS(I)) is "invalid" then the procedure also halts with the outcome "invalid";

Route: x.x.x.x
AS_PATH: AS4 AS2 AS1

ROA {x.x.x.x, AS1}
ASPA {AS1, AS2}
ASPA {AS2, AS3}
ASPA {AS3, 0}
AS_PATH Verification

1. If the closest AS in the AS_PATH is not the receiver's neighbor ASN then procedure halts with the outcome "invalid";

2. If in one of AS_SEQ segments there is a pair (AS(I-1), AS(I)) is "invalid" then the procedure also halts with the outcome "invalid";

Route: x.x.x.x
AS_PATH: AS2 AS1

ROA {x.x.x.x, AS1}
ASPA {AS1, AS2}
ASPA {AS2, AS3}
ASPA {AS3, 0}
Automation: BGP Roles

OPEN with **customer** role

OPEN with **peer** role

Can be fully automated using BGP Roles.
Limitations

• Replay attacks by transit ISPs against its customers;
• Transit ISP can malform AS_PATH that is sent to customers.
Open Questions

• AS_SETs – should we be aggressive?
• Marking malformed routes – attribute vs GRSH?
• ASPA update – how it should affect existing routes?
Summary

• ASPA – it’s simple, it scales;
• Works for both route leaks and hijack detection;
• Low computational cost;
• Doesn’t change the protocol itself;
• Works on existing RPKI infrastructure;
• Brings benefit at state of partial adoption.
## BGP Quadrant: Possible Future

<table>
<thead>
<tr>
<th></th>
<th>BGP Hijacks</th>
<th>BGP Route Leaks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mistake</td>
<td>ROA</td>
<td>ASPA</td>
</tr>
<tr>
<td>Malicious</td>
<td>ROA + ASPA</td>
<td>ROA + ASPA</td>
</tr>
</tbody>
</table>