NLNET LABS?
Makers of fine open source software since 1999
BGP?
RPKI!

10+ years experience
CERTIFICATION AUTHORITY FOR DELEGATED RPKI
Suspicous event hijacks Amazon traffic for 2 hours, steals cryptocurrency

Almost 1,300 addresses for Amazon Route 53 rerouted for two hours.

DAN GOODIN · 4/24/2018, 9:00 PM

Amazon lost control of a small number of its cloud services IP addresses for two hours on Monday when hackers exploited a known Internet-protocol weakness that let them to redirect traffic to rogue destinations. By subverting Amazon’s domain-resolution service, the attackers made cryptocurrency website MyEtherWallet.com and stole about $150,000 in digital coins from unwitting end users. They may have targeted other Amazon customers as well.

Traffic for Google, Apple, Facebook, Microsoft and other tech giants routed through Russia, experts believe it was an intentional BGP Hijacking.

Last week a suspicious event routed traffic for major tech companies (i.e. Google, Facebook, Apple, and Microsoft) through a previously unknown Russian Internet provider. The event occurred on Wednesday, researchers who investigated it believe the traffic was intentionally hijacked.

The incident involved the Internet’s Border Gateway Protocol that is used to route traffic among Internet backbones, ISPs, and other large networks.

How Pakistan knocked YouTube offline (and how to make sure it never happens again)

YouTube becoming unreachable isn’t the first time that Internet addresses were hijacked. But if it spurs interest in better security, it may be the last.
SEPARATE COMPONENTS

CERTIFICATION AUTHORITY
creates & signs

PUBLICATION SERVER
makes available
RELYING PARTY
SOFTWARE

validated
cache
DELEGATED RPKI

• Run an RPKI Certification Authority (CA) as a child of the RIR/NIR/LIR
• Install and maintain software yourself
• Generate your own certificate, have it signed by the parent CA
• Publish signed objects yourself, or ask a third party to do it for you
DELEGATED RPKI

- You are operationally independent from the parent
- Allows tight integration and automation with your own systems
- If you run a global network, you can operate a single system rather than maintain ROAs in up to five web interfaces
- You are in control of the ROA publication interval
- You can delegate or offer RPKI as a service to your customers
FUNDING?
Krill

RPKI Certification Authority
and Publication Server
KRILL CURRENT FUNCTIONALITY

✔ Event sourcing architecture with CLI and API

✔ Up: operate under multiple parent CAs

✔ Down: act as a parent for multiple child CAs

✔ Creation of ROAs

✔ Built-in Publication Server

✔ Allow remote publication
KRILL ROADMAP

- Web-based User Interface “Lagosta” – Coming soon!
- Optional package, with multi-language support
- ROA suggestions, based on global or local view
- Packaging (Docker, *NIX Distributions)
- Monitoring (e.g. Prometheus)
- Multi-master and HSM support (if desired)
KRILL STRETCH GOALS

- Integration with IPAM Solutions
- Just-in-Time Authorisations
- Cloud Provider Marketplace offerings
- Krill-as-a-Service
LAGOSTA PREVIEW
## Certificate Authorities

### Repository

<table>
<thead>
<tr>
<th>Type</th>
<th>Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remote</td>
<td>Service URI: <a href="https://krill-ui-dev.nlnetlabs.nl/rfc8181/ca">https://krill-ui-dev.nlnetlabs.nl/rfc8181/ca</a></td>
</tr>
</tbody>
</table>

### Parents

<table>
<thead>
<tr>
<th>Handle</th>
<th>Kind</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>rfc4182</td>
</tr>
<tr>
<td>v4</td>
<td>10.0.0.0/8</td>
</tr>
<tr>
<td>----</td>
<td>----------</td>
</tr>
<tr>
<td>v6</td>
<td>::/128</td>
</tr>
</tbody>
</table>

### Route Authorizations

<table>
<thead>
<tr>
<th>ASN</th>
<th>Prefix</th>
<th>Max Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>8587</td>
<td>10.1.0.0/22</td>
<td>24</td>
</tr>
<tr>
<td>199664</td>
<td>10.0.0.0/22</td>
<td></td>
</tr>
<tr>
<td>3333</td>
<td>10.2.0.0/22</td>
<td>0</td>
</tr>
</tbody>
</table>
SYSTEM REQUIREMENTS
HARDWARE & CONNECTIVITY

- Certificate Authority
  - Modest hardware is fine for most use cases
  - No HSM needed; keys on disk are fine, really
- Publication Server
  - Offered by NIC.br as a service
  - Publishing yourself will have all normal consequences of a public service
NLNET LABS RUNS 2 CPU / 2 GB RAM

top - 15:32:17 up 4 days, 59 min, 3 users, load average: 0.00, 0.00, 0.00
Tasks: 82 total, 1 running, 81 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.2 us, 0.2 sy, 0.0 ni, 99.5 id, 0.0 wa, 0.0 hi, 0.0 si, 0.2 st
MiB Mem : 1997.7 total, 858.2 free, 110.5 used, 1029.0 buff/cache
MiB Swap: 0.0 total, 0.0 free, 0.0 used. 1689.1 avail Mem

<table>
<thead>
<tr>
<th>PID</th>
<th>USER</th>
<th>PR</th>
<th>NI</th>
<th>VIRT</th>
<th>RES</th>
<th>SHR</th>
<th>%CPU</th>
<th>%MEM</th>
<th>TIME+</th>
<th>COMMAND</th>
</tr>
</thead>
<tbody>
<tr>
<td>1532</td>
<td>do-agent</td>
<td>20</td>
<td>0</td>
<td>706936</td>
<td>25308</td>
<td>11464</td>
<td>S</td>
<td>0.0</td>
<td>1.2</td>
<td>1:06.53 do-agent</td>
</tr>
<tr>
<td>4230</td>
<td>krill</td>
<td>20</td>
<td>0</td>
<td>831436</td>
<td>21820</td>
<td>13912</td>
<td>S</td>
<td>0.0</td>
<td>1.1</td>
<td>5:52.01 krill</td>
</tr>
<tr>
<td>20190</td>
<td>www-data</td>
<td>20</td>
<td>0</td>
<td>75016</td>
<td>11096</td>
<td>5312</td>
<td>S</td>
<td>0.0</td>
<td>0.5</td>
<td>6:29.93 nginx</td>
</tr>
<tr>
<td></td>
<td>root</td>
<td>20</td>
<td>0</td>
<td>170448</td>
<td>10264</td>
<td>7856</td>
<td>S</td>
<td>0.0</td>
<td>0.5</td>
<td>0:07.94 systemd</td>
</tr>
<tr>
<td>220</td>
<td>root</td>
<td>20</td>
<td>0</td>
<td>43096</td>
<td>9864</td>
<td>8336</td>
<td>S</td>
<td>0.3</td>
<td>0.5</td>
<td>3:08.91 systemd-journal</td>
</tr>
<tr>
<td>4268</td>
<td>root</td>
<td>20</td>
<td>0</td>
<td>71856</td>
<td>8596</td>
<td>6420</td>
<td>S</td>
<td>0.0</td>
<td>0.4</td>
<td>0:00.00 nginx</td>
</tr>
<tr>
<td>9854</td>
<td>root</td>
<td>20</td>
<td>0</td>
<td>16928</td>
<td>8436</td>
<td>7072</td>
<td>S</td>
<td>0.3</td>
<td>0.4</td>
<td>0:00.21 sshd</td>
</tr>
<tr>
<td>20188</td>
<td>www-data</td>
<td>20</td>
<td>0</td>
<td>72392</td>
<td>8344</td>
<td>5244</td>
<td>S</td>
<td>0.0</td>
<td>0.4</td>
<td>1:35.86 nginx</td>
</tr>
<tr>
<td>2613</td>
<td>root</td>
<td>20</td>
<td>0</td>
<td>21024</td>
<td>8220</td>
<td>7100</td>
<td>S</td>
<td>0.0</td>
<td>0.4</td>
<td>0:00.06 systemd</td>
</tr>
</tbody>
</table>
WHAT IF IT BREAKS?

- No DNSSEC horror story; e.g. unavailable zone due to signing mishap
- RPKI provides a positive statement on routing intent
- Lose your keys? Hardware failure?

All routes will eventually fall back to the “NotFound” state, as if RPKI were never used
WHATEVER YOU DO, GO ALL IN!

- It’s better to create **no** ROAs than **bad** ones
- Once you start create ROAs, **maintain** them!
- Make RPKI part of standard operations
- Set up monitoring and alerting
- Train your first line help desk
VIBRANT ECO SYSTEM
RELYING PARTY SOFTWARE

- FORT Validator, by NIC.mx & LACNIC (in C)
- Routinator 3000, by NLnet Labs (in Rust)
- OctoRPKI, by Cloudflare (in Go)
- RIPE NCC RPKI Validator (in Java)
- Dragon Research Labs Validating Cache (in Python)
- RPSTIR, by Raytheon BBN Technologies (in C)
- OpenBSD rpki-client(1) (in C)
RPKI SUPPORT

• FORT Project to support routing security in Latin America

• Documentation and FAQ – rpki.readthedocs.io
  • Community driven, allows translations

• Public mailing list with 200+ subscribers – rpki@nlnetlabs.nl

• Commercial support with SLA on Krill and Routinator