DEFENDING RPKI

DKNOG15

Job Snijders

AGENDA

01 THREATS TO RPKI

02 HIDING IN THE CROWD

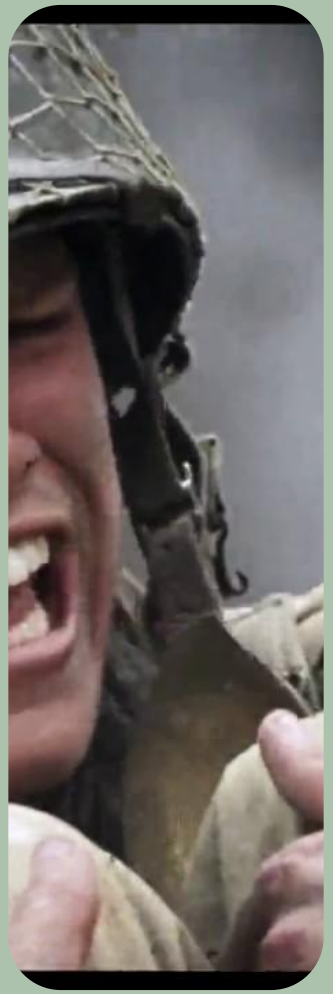
03 PRELIMINARY RESULTS

04 THE PLAN

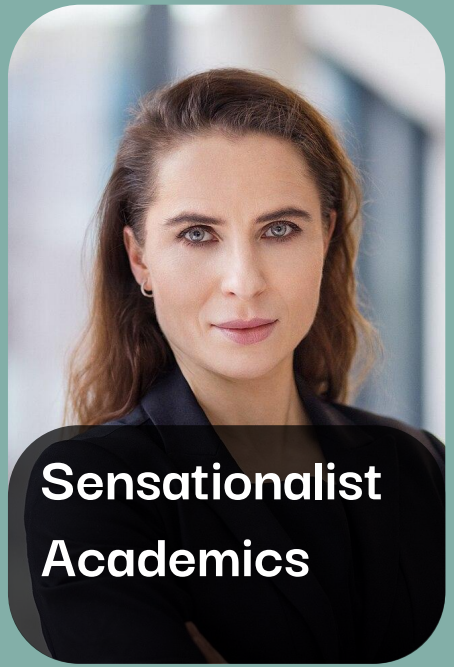
05 QUESTIONS

THREATS TO RPKI

1

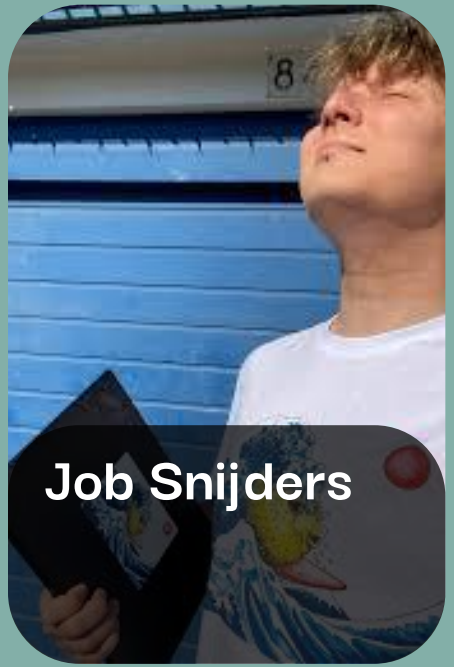


MEET THE TEAM



Sensationalist Academics

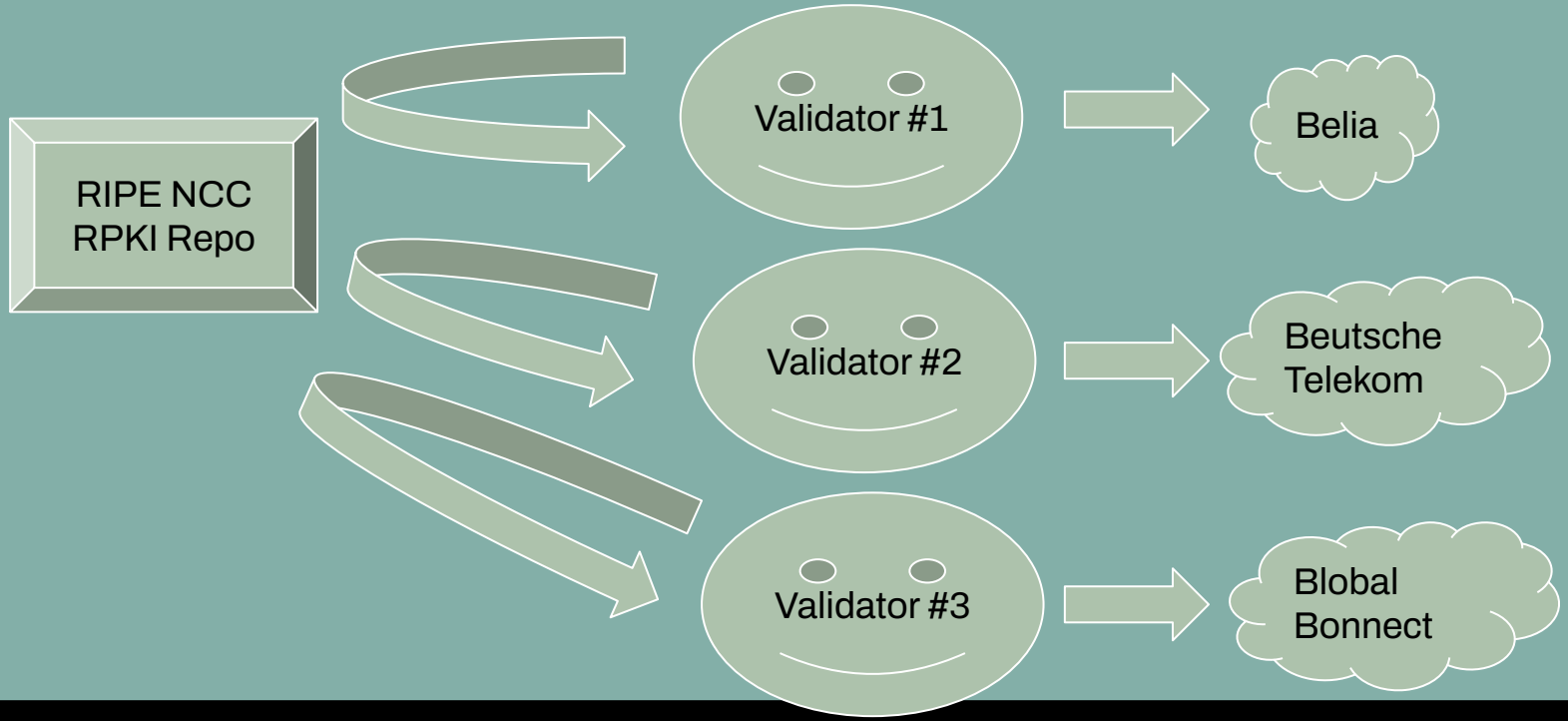
Unethical Hackers



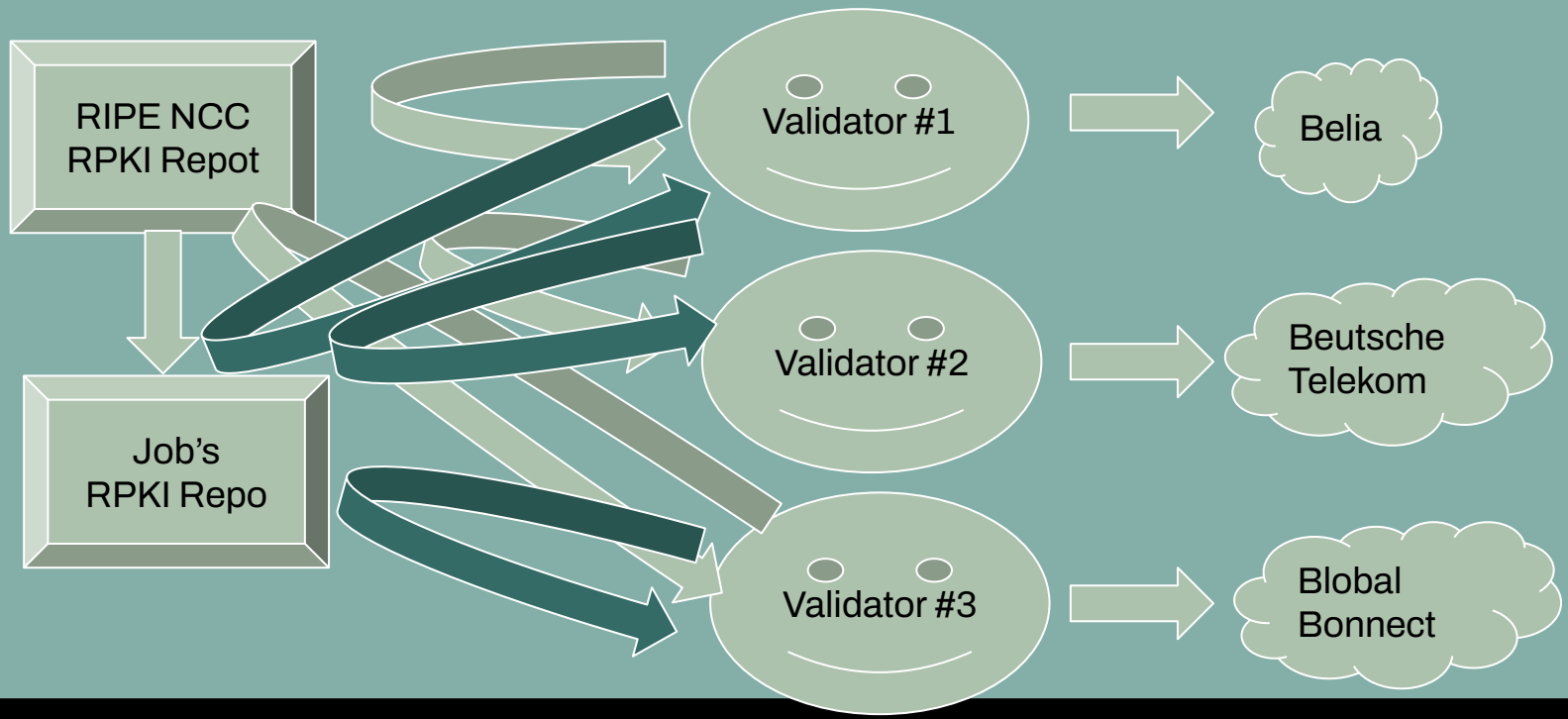
Job Snijders

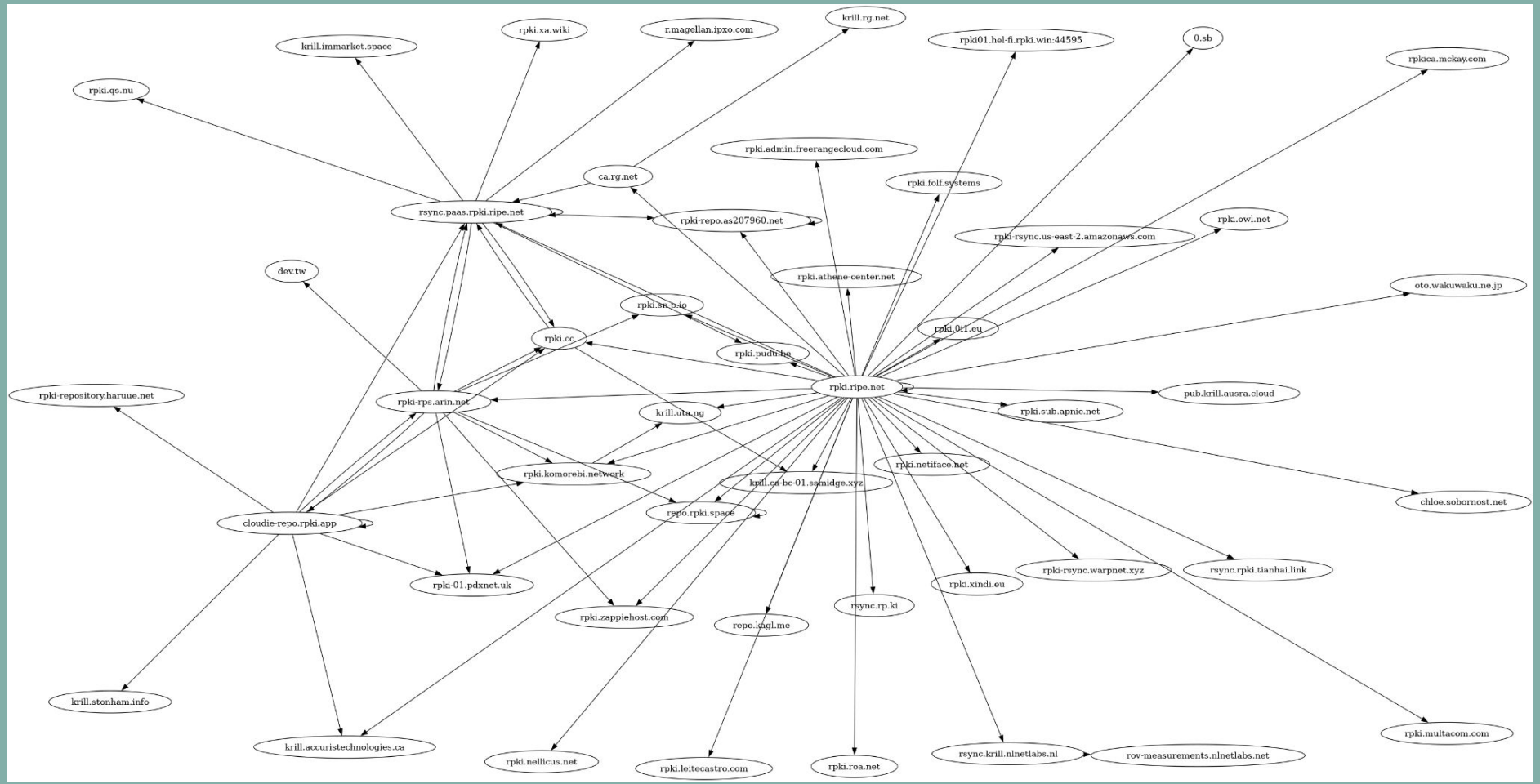
Well meaning Operators

RPKI Architecture: Every validator connects to every publication point



RPKI Architecture: Every validator connects to every publication point




krill.immarket.space
rpki.xa.wiki
r.magellan.ipxo.com
krill.rg.net
rpki01.hel-fi.rpki.win:44595
0.sb
rpkica.mckay.com
rpki.qs.nu
rpki.admin.freerangecloud.com
ca.rg.net
rpki.folf.systems
rsync.paas.rpki.ripe.net
rpki-repo.as207960.net
rpki-rsync.us-east-2.amazonaws.com
rpki.owl.net
dev.tw
rpki.athene-center.net
oto.wakuwaku.ne.jp
rpki.sn.p.io
rpki.0t1.eu
rpki.cc
rpki.pudu.ng
rpki.ripe.net
pub.krill.ausra.cloud
rpki-repository.haruue.net
rpki-rps.arin.net
krill.uta.ng
rpki.sub.apnic.net
rpki.komorebi.network
rpki.netiface.net
krill.ca-bc-01.ssmidge.xyz
chloe.sobornost.net
repo.rpki.space
cloudie-repo.rpki.app
rpki-rsync.warpnet.xyz
rsync.rpki.tianhai.link
rpki-01.pdxnet.uk
rpki.xindi.eu
rsync.rp.ki
rpki.zappiehost.com
repo.kagl.me
krill.stonham.info
rpki.multacom.com
krill.accuristechnologies.ca
rsync.krill.nlnetlabs.nl
rov-measurements.nlnetlabs.net
rpki.nellicus.net
rpki.leitecastro.com
rpki.roa.net

Repositories can serve poison pills

Poster: From Fort to Foe: The Threat of RCE in RPKI

Oliver Jacobsen
ATHENE, Darmstadt, Germany
Goethe-Universität Frankfurt, Frankfurt, Germany

Haya Schulmann
ATHENE, Darmstadt, Germany
Goethe-Universität Frankfurt, Frankfurt, Germany

Niklas Vogel
ATHENE, Darmstadt, Germany
Goethe-Universität Frankfurt, Frankfurt, Germany

Michael Waidner
ATHENE, Darmstadt, Germany
TU Darmstadt, Darmstadt, Germany

Abstract

In this work, we present a novel severe buffer-overflow vulnerability in the RPKI validator Fort, that allows an attacker to achieve Remote Code Execution (RCE) on the machine running the software. We discuss the unique impact of this RCE on networks that use RPKI, illustrating that RCE vulnerabilities are especially severe in the context of RPKI. The design of RPKI makes RCE easy to exploit on a large scale, allows compromise of RPKI validation integrity, and enables a powerful vector for additional attacks on other critical components of the network, like the border routers.

We analyze the vulnerability exposing to this RCE and identify

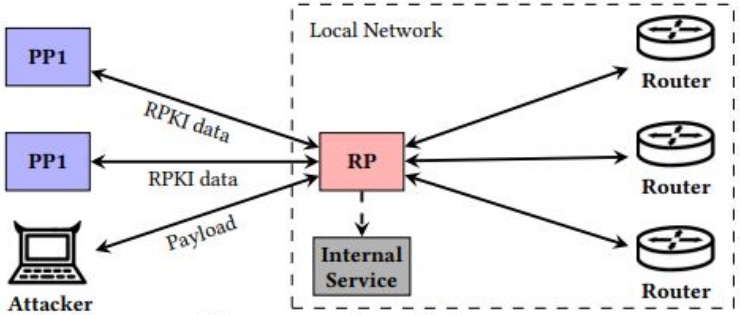
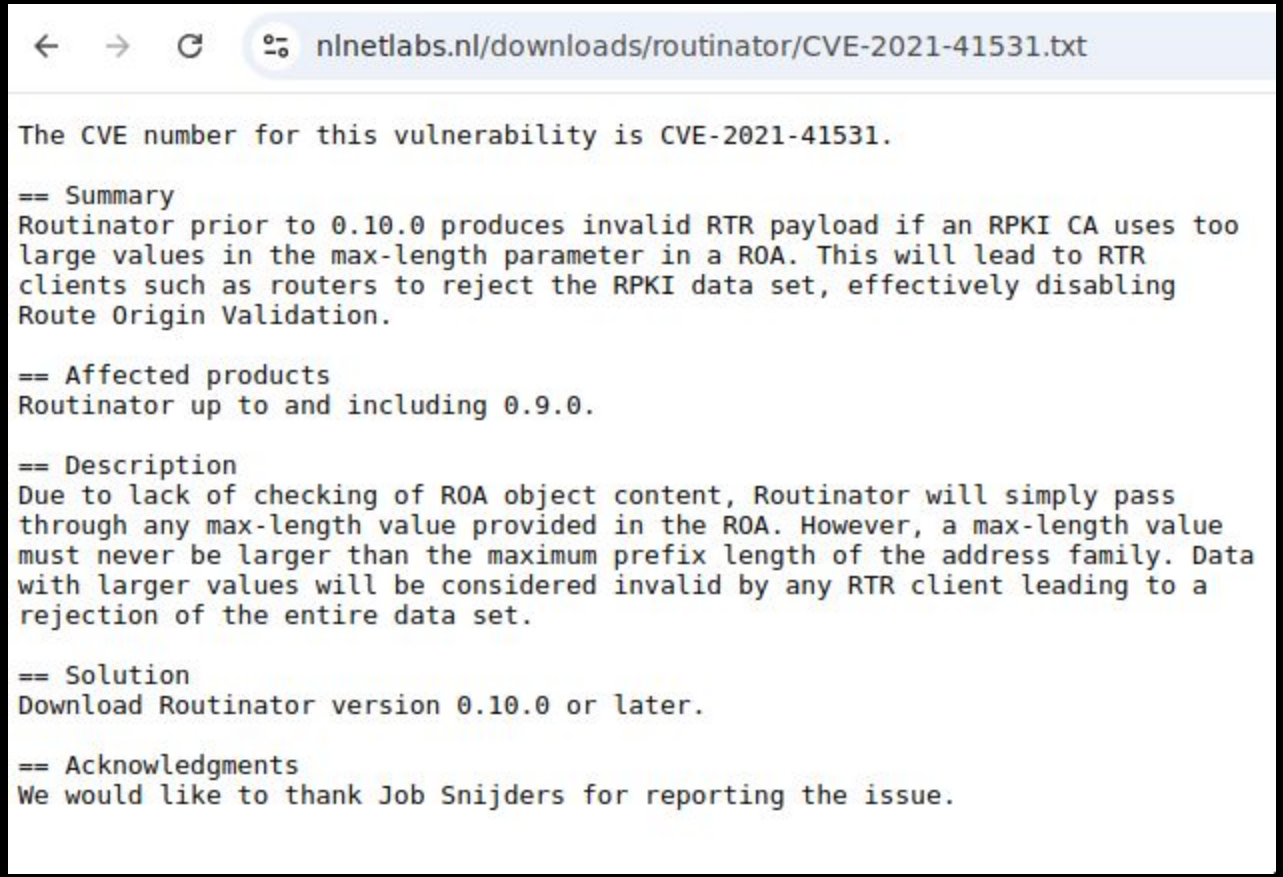
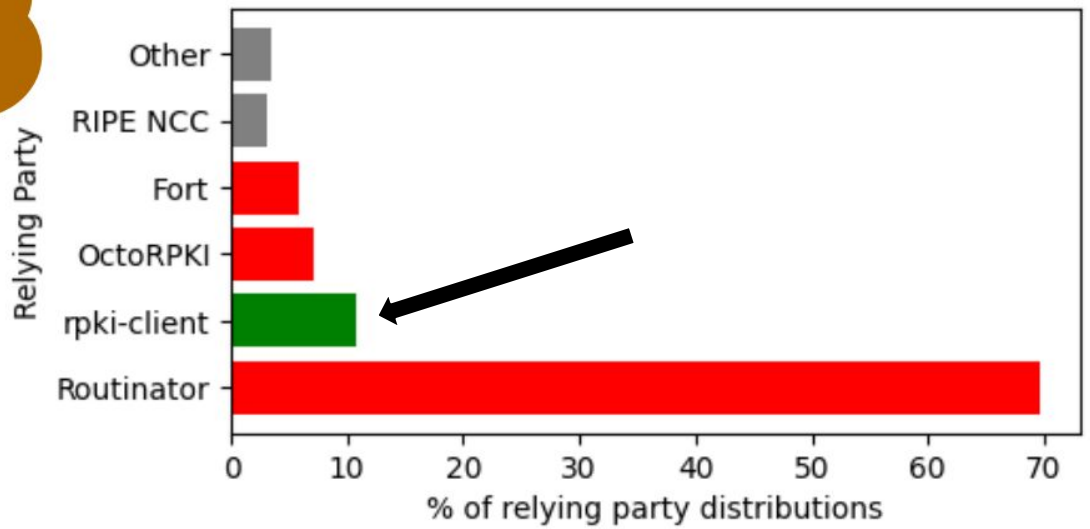


Figure 1: Attack Setup.


nlnetlabs.nl/downloads/routinator/CVE-2021-41531.txt
The CVE number for this vulnerability is CVE-2021-41531.
== Summary
Routinator prior to 0.10.0 produces invalid RTR payload if an RPKI CA uses too large values in the max-length parameter in a ROA. This will lead to RTR clients such as routers to reject the RPKI data set, effectively disabling Route Origin Validation.
== Affected products
Routinator up to and including 0.9.0.
== Description
Due to lack of checking of ROA object content, Routinator will simply pass through any max-length value provided in the ROA. However, a max-length value must never be larger than the maximum prefix length of the address family. Data with larger values will be considered invalid by any RTR client leading to a rejection of the entire data set.
== Solution
Download Routinator version 0.10.0 or later.
== Acknowledgments
We would like to thank Job Snijders for reporting the issue.

Summary of Results

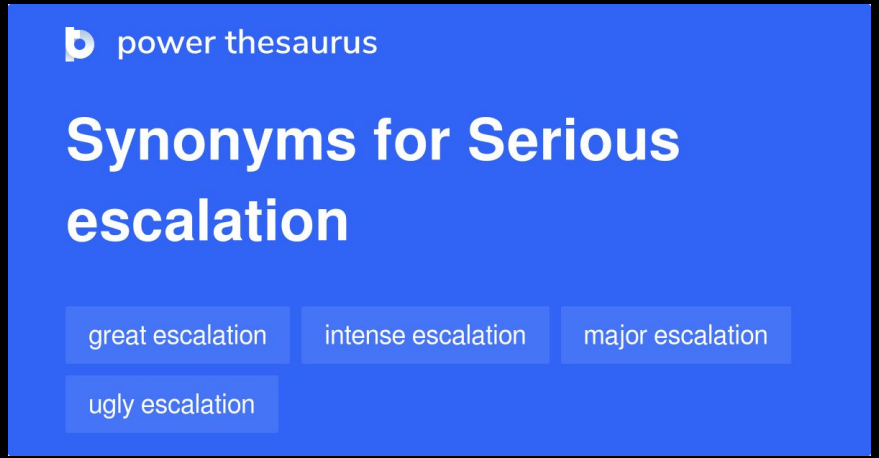
We found issues on **3 out of 4** maintained RPs



18 total vulnerabilities & 5 CVEs

THINGS TO WORRY ABOUT

1. Crashing validator instances
2. Operating System compromise via RPKI validators entrypoint
3. Operators disabling RPKI “*because RPKI itself seems a risk*”
4. The clock on Internet routing security turning back



For the sake of this discussion:

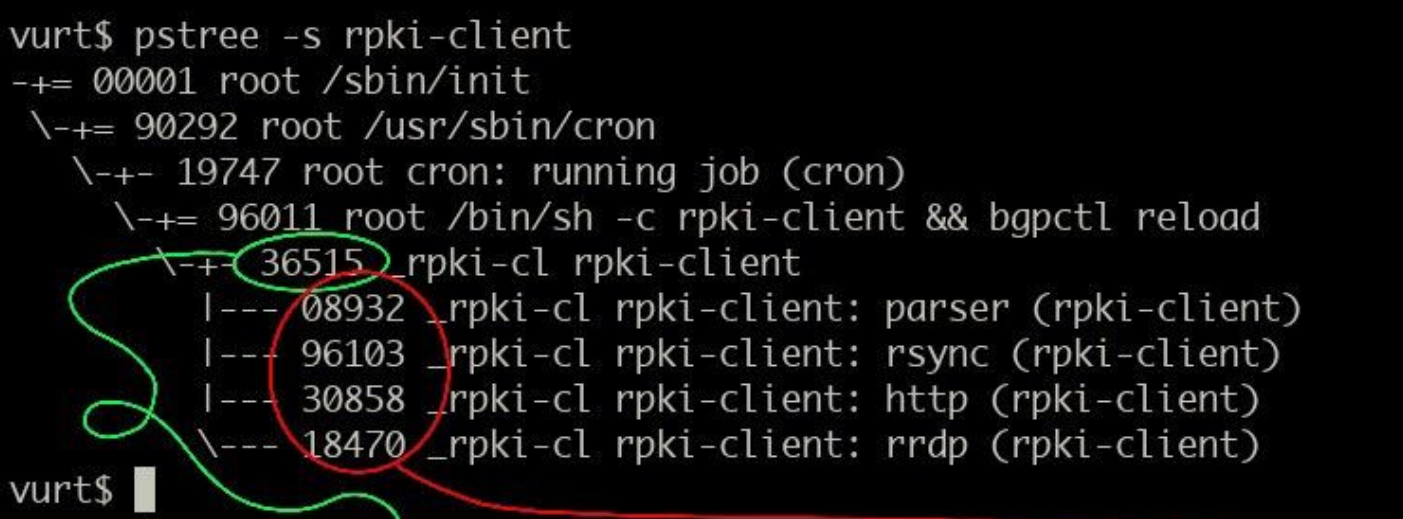
- Every year, ***vulnerabilities will be found*** in validator implementations
- Knowing which validator instances influence which ASes is ***leverage***
- Targeted attacks require ***targeting precision***
- Select (validator <> repository) paths are persistently broken

WHAT CAN BE DONE?

2

A Robust Validator: rpki-client

- Don’t reinvent the wheel: use battle-tested libraries (libcrypto)
- Sandboxing (Linux landlock, OpenBSD unveil & pledge)
- Randomize what can be randomized: unpredictability is king
- Box in resource consumption:
  - Maximum download size
  - Maximum file sizes / minimum file sizes
  - Maximum time spent on a single repository
  - Maximum time spent on all repositories
  - Limit chain length, limit the number of repositories
  - etc



The privileged parent and unprivileged children communicate via simple, well-defined interfaces ("pipes").

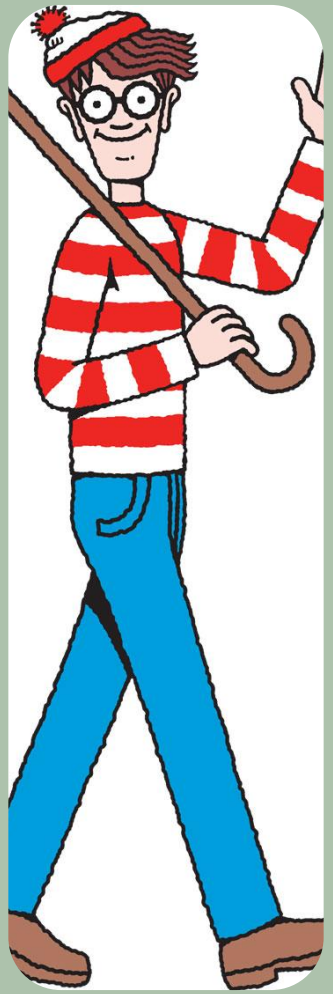
Each child process handles untrusted and potentially hostile data inside its own restricted environment.

Accidental corruption of a child does not lead to a compromise of the parent, keeping the network safe.


What Else Can I Do?
COVERED BY ANNA
ARTWORK BY IKIMARUART

HIDING IN THE CROWD

3



HIDING RPKI VALIDATORS

- Publication point operators don’t need to know the source IP addresses of validators, do they?
- Knowing what instance at what IP address influences what ISPs is ***leverage***
- Conceptually, Internet-wide Multicast would’ve been great for RPKI , but … that’s in an alternate universe

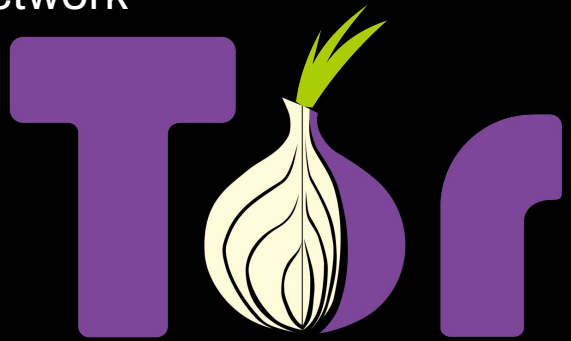

STRENGTH
IN NUMBERS
State Farm
State Farm
State Farm
ORACLE
ARENA

Anonymizing validators

- The RPKI protocols only require RPs to posses the data from Publication Points
- The RPKI protocols do ***not*** require publication points to know the source IP addresses of Validators

Therefore, obviously:

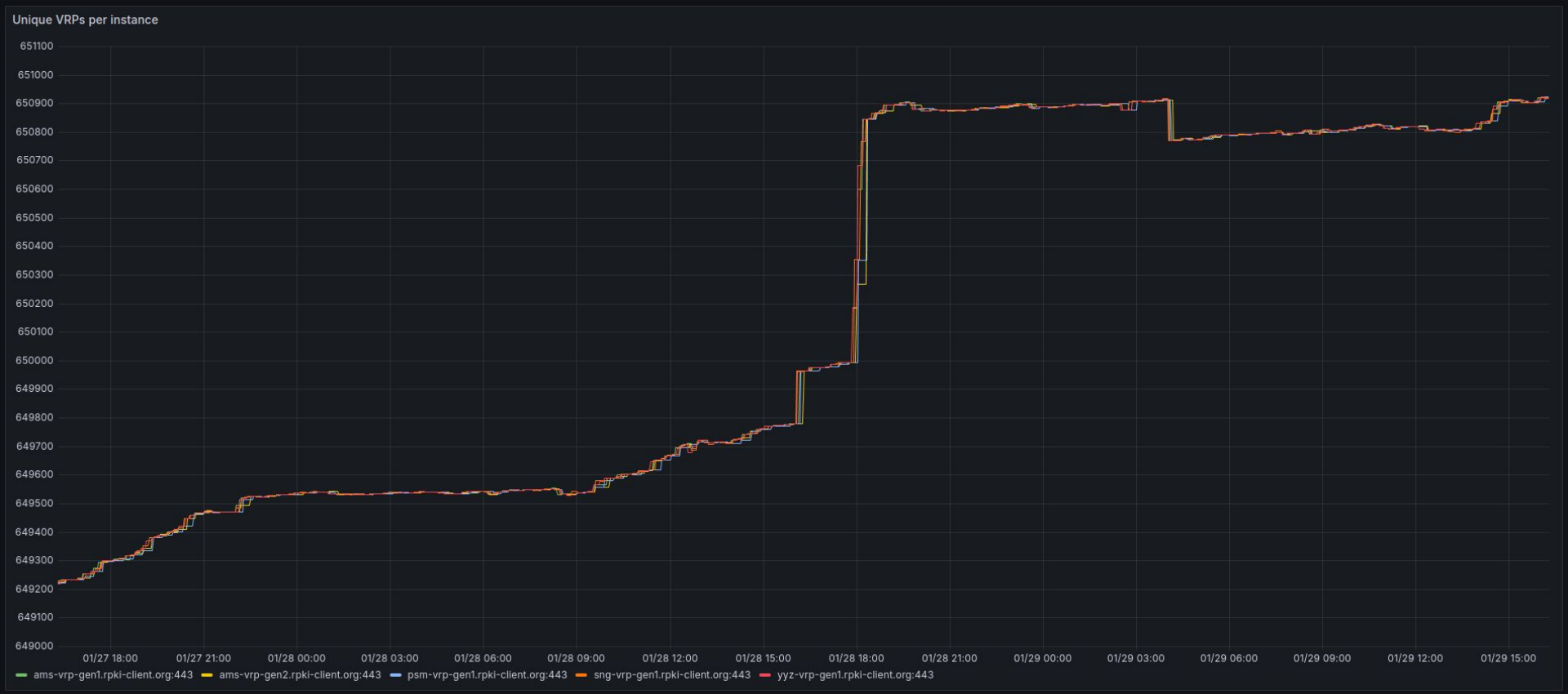
- Validators should use a globally distributed network of forward proxies
- Validators should use the Tor Onion VPN network



PRELIMINARY RESULTS

4

Is It Me You're Looking For?


Unique VRPs per instance
651100
651000
650900
650800
650700
650600
650500
650400
650300
650200
650100
650000
649900
649800
649700
649600
649500
649400
649300
649200
649100
649000
01/27 18:00
01/27 21:00
01/28 00:00
01/28 03:00
01/28 06:00
01/28 09:00
01/28 12:00
01/28 15:00
01/28 18:00
01/28 21:00
01/29 00:00
01/29 03:00
01/29 06:00
01/29 09:00
01/29 12:00
01/29 15:00
ams-vrp-gen1.rpki-client.org:443
ams-vrp-gen2.rpki-client.org:443
psm-vrp-gen1.rpki-client.org:443
sng-vrp-gen1.rpki-client.org:443
yyz-vrp-gen1.rpki-client.org:443

PRELIMINARY RESULTS

- No significant difference between “anonymized” and “normal” validator instances!
- RRDP-via-overlay not as reliable as “direct”, but…
  - As long as “direct” is used as fallback, no difference
  - The overlay also helps overcome broken connectivity!

THE PLAN

4

PLAN: AREA OF STUDY

- Does use of *forward proxies* at scale work well for the RPKI?
  - Could forward proxies work well inside tor? (inside .onion)

- How to handle transport switchovers?
  - RRDP to RSYNC
  - RSYNC to RRDP

- Set up more experiments: find out reliability numbers

REQUEST TO YOU

- I need … Compute & Storage resources to run experiments

QUESTIONS?

5

THANK YOU

Job Snijders

job@sobornost.net