RRDP in OpenBSD’s RPKI validator rpki-client

Job Snijders
job@fastly.com

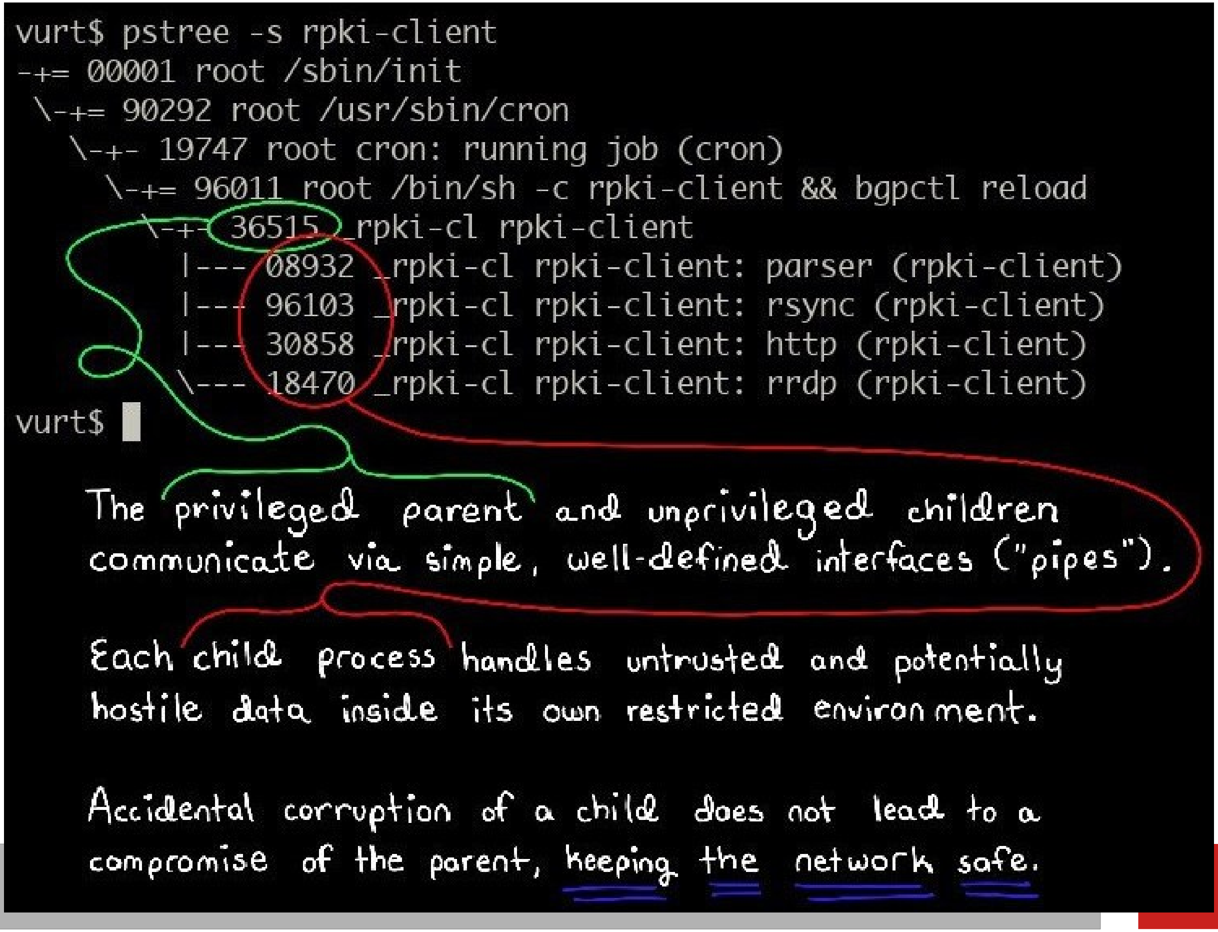
ARIN 48
September 2021

What is RRDP? A HTTP based photocopier for Signed RPKI objects!

Challenging aspects about RRDP for RPKI implementers

- Mapping “RRDP places” to “RSYNC places” is not straightforward, the RRDP RFC could’ve done more to guide implementers. OpenBSD hashes the *NotifyURL*, and uses that as part of the filename. (A cool trick other validators later on copied!)
- RRDP is quite bloated because of the needless use of XML! XML is only used as a very expensive field delimiter…
- Data inconsistencies can also exist in RRDP publications, just like they can in RSYNC!
- RRDP contains a dangerous ‘withdraw’ instruction, with is unauthenticated and unsigned.

Rpki-client: enhanced security through *process privilege separation*



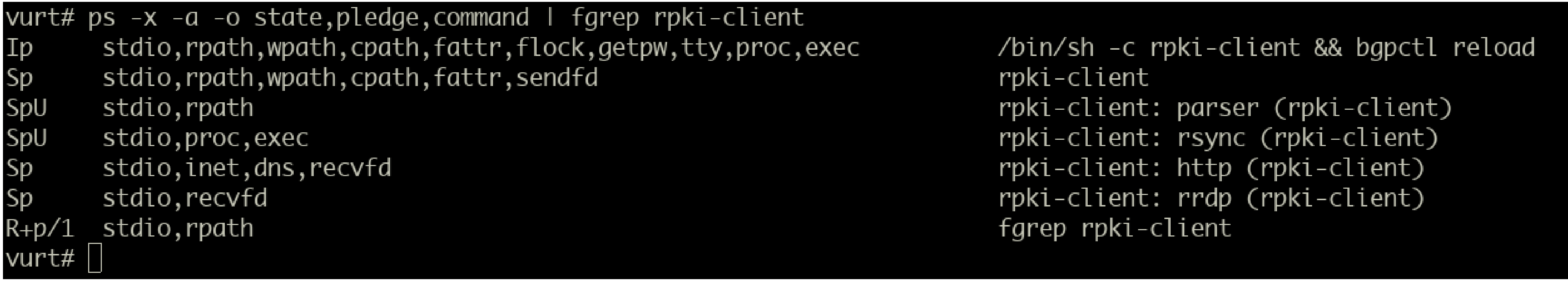
```
vurt$ pstree -s rpki-client
-+= 00001 root /sbin/init
 \-+= 90292 root /usr/sbin/cron
   \-+- 19747 root cron: running job (cron)
     \-+= 96011 root /bin/sh -c rpki-client && bgpctl reload
       \-+- 36515 _rpki-cl rpki-client
         |--- 08932 _rpki-cl rpki-client: parser (rpki-client)
         |--- 96103 _rpki-cl rpki-client: rsync (rpki-client)
         |--- 30858 _rpki-cl rpki-client: http (rpki-client)
         \--- 18470 _rpki-cl rpki-client: rrdp (rpki-client)
vurt$ 
```

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.

Pledge() to constrain which subprocess can make what system calls



```
vurt# ps -x -a -o state,pledge,command | fgrep rpki-client
Ip     stdio,rpath,wpath,cpath,fattr,flock,getpw,tty,proc,exec          /bin/sh -c rpki-client && bgpctl reload
Sp     stdio,rpath,wpath,cpath,fattr,sendfd                             rpki-client
SpU    stdio,rpath                                                      rpki-client: parser (rpki-client)
SpU    stdio,proc,exec                                                  rpki-client: rsync (rpki-client)
Sp     stdio,inet,dns,recvfd                                            rpki-client: http (rpki-client)
Sp     stdio,recvfd                                                     rpki-client: rrdp (rpki-client)
R+p/1  stdio,rpath                                                      fgrep rpki-client
vurt# 
```

Final thoughts

- **The OpenBSD RRDP implementation helped uncover issues in almost CA Repo:**
  - Load-balancers serving inconsistent data for related requests
  - Junk being encoded in RRDP XML
  - <*Withdraw/*> instructions for cryptographically valid objects that should not be withdrawn

CA Operators benefit from a diverse set of tools to test their RPKI service. The community as a whole benefits from increased protocol design scrutiny. RRDP has some nice features, but also some fundamental design issues. RSYNC is great for debugging, RRDP is nice for bulk transport.

Our industry needs both RSYNC and RRDP! … ***It's not a contest...***