Update of SLURM
(Simplified Local internet nUmber Resource Management with the RPKI)

IETF 97

Di Ma

ZDNS

madi@zdns.cn

SLURM Review

- ## Motivations
  - Network operators MAY want to selectively override the RPKI hierarchy at its discretion as for private INRs.
  - Network operators MAY wish to make use of a local override capability to protect routes from adverse actions [I-D.ietf-sidr-adverse-actions], until the results of such actions have been addressed.

- ## Methodology
  - A relying party uses both output filtering and locally added assertions to modify validated cache.

Update Overview

- Reorganize the layout of the intended content

- Rewrite the use cases

- Give an overview of SLURM by adding a figure of SLURM's Position in the Relying Party Stack

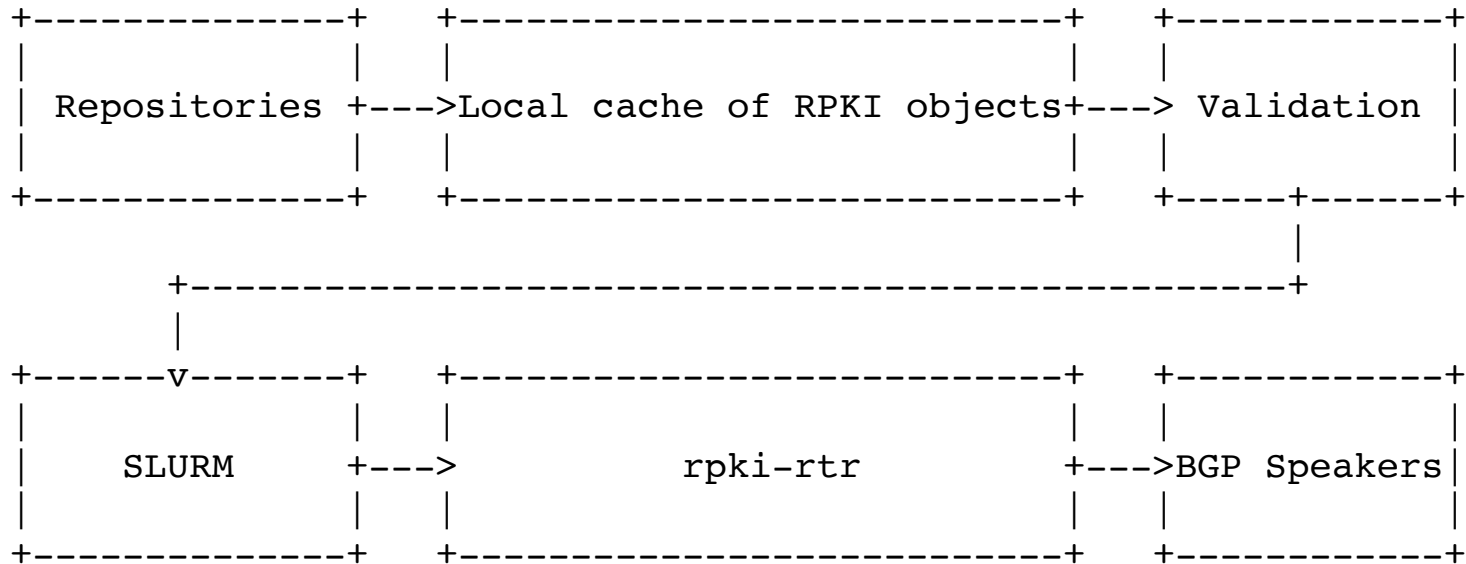- Add more text to Security Considerations

Reorganized Layout

- RPKI RPs with SLURM

- SLURM Mechanisms
  - Validation Output Filtering
  - Locally Adding Assertions
  - Combining Mechanisms

- Format of the SLURM

- SLURM File Configuration
  - SLURM File Atomicity
  - Multiple SLURM Files

Usecase Revision

- Making the motivation unfocused from private INR by changing expressions throughout the I-D

- Referring to draft-ietf-sidr-adverse-actions

SLURM's Position in the Relying Party Stack

```
+-------------+    +---------------------------+    +------------+
|             |    |                           |    |            |
| Repositories+--->Local cache of RPKI objects+---> Validation  |
|             |    |                           |    |            |
+-------------+    +---------------------------+    +-----+------+
                                                          |
      +---------------------------------------------------+
      |
+------v------+    +---------------------------+    +------------+
|             |    |                           |    |            |
|   SLURM     +--->          rpki-rtr          +--->BGP Speakers|
|             |    |                           |    |            |
+-------------+    +---------------------------+    +------------+
```

Security Considerations

- Manipulation on assertions about non-private INRs

- Errors in the SLURM file

- Authenticity and Integrity of the SLURM file

Reconsideration on SLURM file format

- ABNF V.S. JSON/XML/YAML
  - ABNF is used widely to define syntax of program language, which expresses the essential logic.
  - There are wide availability of libraries to parse JSON/XML/YAML.

- Format is different from format instruction.
  - SLURM file is just configuration file. Yet the RP needs to be reinforced with new module to support SLURM.
  - ABNF is employed in this I-D to specify SLURM file format.
  - Implementers are free to choose JSON/XML/YAML mapped from ABNF to generate SLURM file to be configured.

Examples of SLURM File

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<root>
    <head>
        <firstLine>SLURM 1.0</firstLine>
    </head>
    <head>
        <target>hostname=rpki.example.com</target>
    </head>
    <body>
        <add>
            <type>origination</type>
            <IPv4>192.0.2.0/24</IPv4>
            <ASnum>65536</ASnum>
        </add>
    </body>
    <body>
        <add>
            <type>bgpsec</type>
            <ASnum>65535</ASnum>
            <RouterSKI>akhssbndbjshgfjsbfwe+0r=s</RouterSKI>
            <RouterPubKey>lkmdcdjnvhbdsk=</RouterPubKey>
        </add>
    </body>
    <body>
        <del>
            <type>origination;IPv6:2001:778::/48-52</type>
        </del>
    </body>
    <body>
        <del>
            <type>bgpsec</type>
            <ASnum>65536</ASnum>
        </del>
    </body>
</root>
```

```json
{
    "head": [
        {
            "firstLine": "SLURM 1.0"
        },
        {
            "target": "hostname=rpki.example.com"
        }
    ],
    "body": [
        {
            "add": {
                "type": "origination",
                "IPv4": "192.0.2.0/24",
                "ASnum": 65536
            }
        },
        {
            "add": {
                "type": "bgpsec",
                "ASnum": 65535,
                "RouterSKI": "akhssbndbjshgfjsbfwe+0r=s",
                "RouterPubKey": "lkmdcdjnvhbdsk="
            }
        },
        {
            "del": {
                "type": "origination;IPv6:2001:778::/48-52"
            }
        },
        {
            "del": {
                "type": "bgpsec",
                "ASnum": 65536
            }
        }
    ]
}
```

```yaml
├ SLURM 1.0
- target: hostname=rpki.example.com
- add:
    - origination
    - 192.0.2.0/24
    - 65536    #comments
- add:
    - bgpsec
    - 65535
    - akhssbndbjshgfjsbfwe+0r=s
    - lkmdcdjnvhbdsk=
- del:
    - origination
    - 2001:DB8::/48-52
- del:
    - bgpsec
    - 65536
```

9

Implementation

- RPSTIR, as a sort of RPKI RP software, will be supporting SLURM in the coming future.

- Thanks go to Steve Kent for his guidance and detailed reviews in preparing this updated I-D.

- Thanks go to Tim Bruijnzeels and Rob Austein for sharing with me their considerations on SLURM file format.

Thanks!