# ROAキャッシュサーバハンズオン

## ～RPKI/ROVの普及を目指して～

長崎県立大学
講師 岡田 雅之
アシスタント 齋藤脩愉・後藤汰珠

Internet Week 2023

# 自己紹介

# 講師−自己紹介

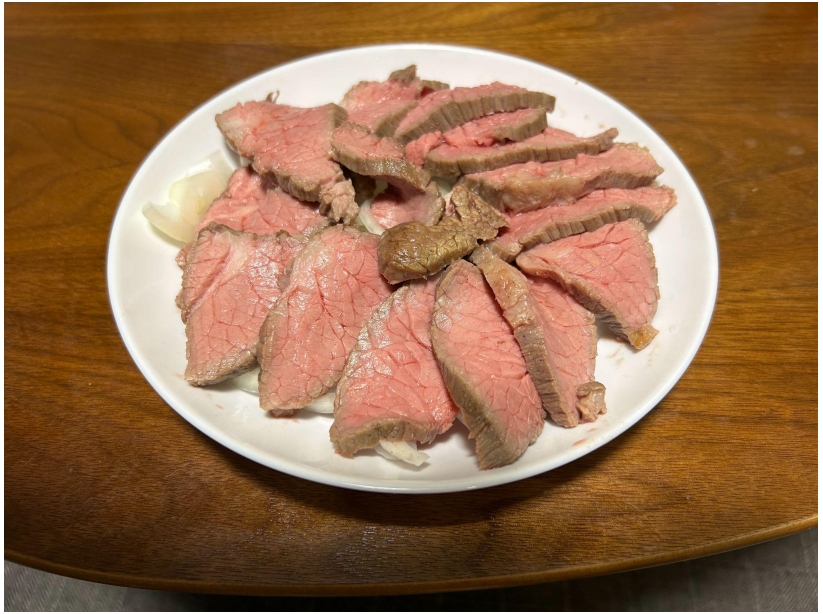
- 名前：岡田雅之
- 所属：長崎県立大学 情報セキュリティ学科
  - 教授
  - （2020年までJPNIC）
- 出身：茨城県生まれ 千葉育ち
- 趣味：
  - AS運用

# TA-自己紹介

- 名前：齋藤 脩愉
- 所属：長崎県立大学 情報セキュリティ学科 B3
- 出身：香川県高松市
  - 今は転勤で実家が大阪府八尾市に
- 趣味：小室哲哉の音楽を聞くこと
  - よくBillboard LIVE OSAKAに遠征しております

# TA-自己紹介

- 名前：後藤汰珠
- 所属：長崎県立大学 情報セキュリティ学科 B3
- 出身：大分県大分市
  - とり天やカボスが有名です
- 趣味：料理
  - 最近作ったものはローストビーフです



# ハンズオンの流れ

# ハンズオンの流れ

1. 接続準備
2. libreTLSのインストール
3. rpki-clientのインストール
4. Stay-RTRのインストール
5. BGPルーターの設定
6. 構築環境の確認

# 接続準備

# 接続準備（接続先確認）

本ハンズオンで利用する
各種クレデンシャルやIP,AS番号の割り振りリスト
InternetWeek2023の来場者証Noで割り振ってますのでご注意を。

https://docs.google.com/spreadsheets/d/1rJvN7GhK3azwuZeFkyQr-v17FmqS9TzZchkkxYsJ12A/edit?usp=drive_link

# 接続準備

踏み台にSSHします

```
ssh [踏み台user名]@[踏み台IP]

password:Intern3tWeek@202e#!
```

# 接続準備

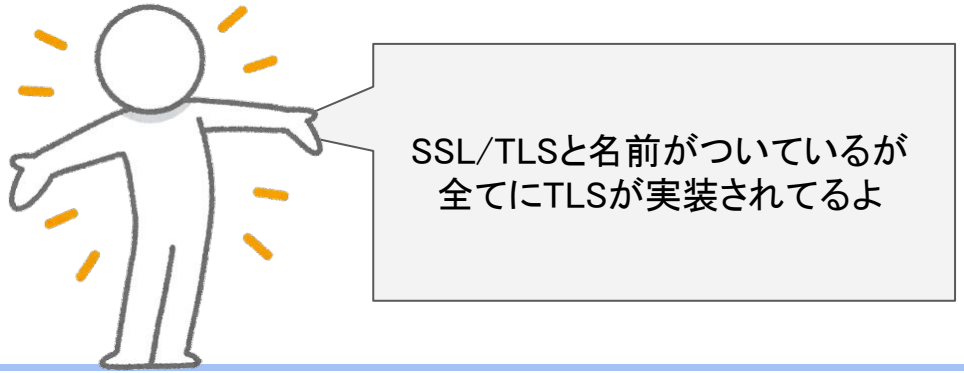
踏み台から演習環境にSSHします

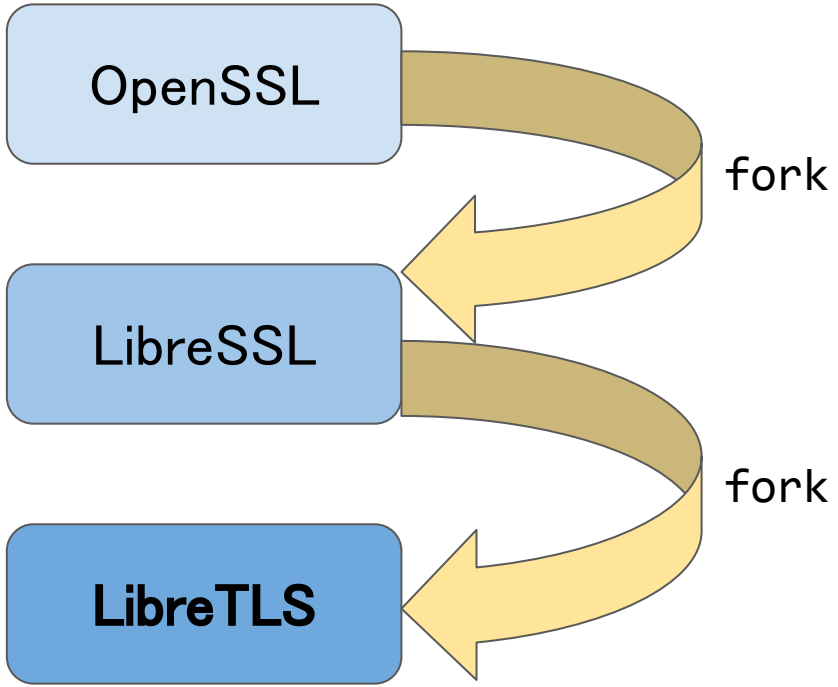
```
(Ubuntu) ssh user01@[server IP]
(Cat8000V)ssh admin@[router IP]

password:iw23#!
```

# LibreTLSのインストール
# （Ubuntuでの操作）

# LibreTLSとは

OpenSSLのフォークの
LibreSSLのフォークが
**LibreTLS**
**OpenSSLやLibreSSLと比較し、**
**RPKIに関する機能拡張が豊富**



# LibreTLSのインストール

- buildツールのインストール

```
sudo apt install -y wget openssl rsync build-essential gcc git autoconf \
libtool automake libssl-dev libbsd-dev gawk libexpat1-dev libtlsh-dev \
zlib1g zlib1g-dev
```

# LibreTLSのインストール

- libreTLSのインストール

```
cd ~

wget https://ftp.openbsd.org/pub/OpenBSD/LibreSSL/libressl-3.8.2.tar.gz

tar -xzvf libressl-3.8.2.tar.gz && rm libressl-3.8.2.tar.gz
```

# LibreTLSのインストール

- libreTLSのmake

```
cd libretls-3.8.2

./configure --libdir=/lib/x86_64-linux-gnu

make check

sudo make install
```

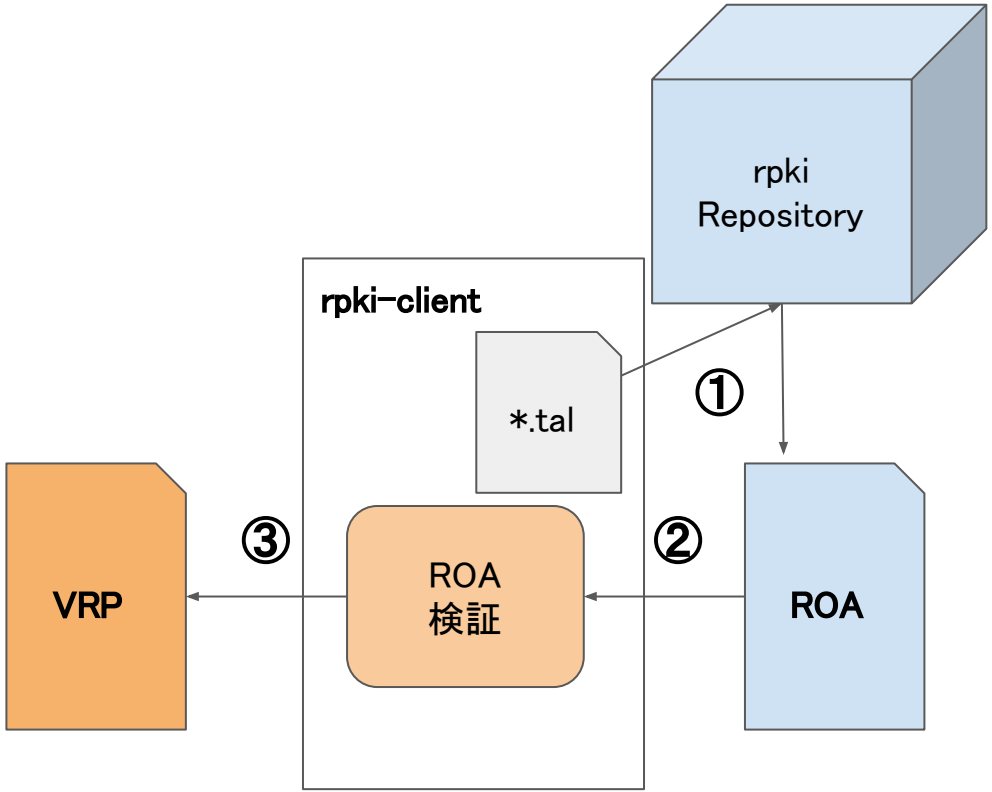
# rpki-clientのインストール
# （Ubuntuでの操作）

# rpki-clientとは

①TAL(Trust Anchor Locator)を元に
ROA(Route origin authorization)を取得

②ROAの検証

③検証済みROAを
VRP(Validate ROA Payload)
として出力



# rpki-clientのインストール

- グループの作成とユーザーの追加

```
cd ~

sudo groupadd _rpki-client

sudo useradd -g _rpki-client -s /sbin/nologin -d /noneexistent _rpki-client
```

# rpki-clientのインストール

- rpki-clientのインストール

```
wget https://ftp.openbsd.org/pub/OpenBSD/rpki-client/rpki-client-8.6.tar.gz \
&& tar xzvf rpki* && rm rpki-client-8.6.tar.gz
```

# rpki-clientのインストール

- rpki-clientのビルド

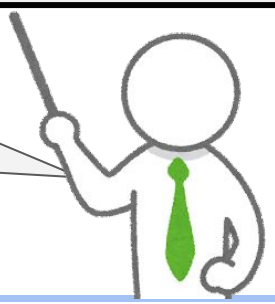
```
cd rpki-client-8.6

./configure

make

sudo make install
```

# rpki-clientのインストール

- arin.talのダウンロード

```
sudo wget https://www.arin.net/resources/manage/rpki/arin.tal -O \
/usr/local/etc/rpki/arin.tal
```

少し前まで、ARINのRPKI利用には制限があったため、手動でarin.talをインストールする必要がありました。他のRIRsのTALは初めからインストールされています。



# rpki-clientのインストール

- rpki-clientの実行（少し時間がかかります）

```
sudo rpki-client
```

# rpki-clientのインストール

- 各種確認

```
which rpki-client     (rpki-clientの場所確認)


rpki-client -c -v     (結果をCSVで出力)
ls -lash /usr/local/var/db/rpki-client/
head /usr/local/var/db/rpki-client/csv
Defaultでは、JSON, OpenBGPD, BIRD, CSV and OpenMetricにて出力
```

CSV出力はオプション無しと
同程度に時間がかかるので
時間に余裕があれば試してみてください

# rpki-client

- 全世界のリポジトリからROAに関するファイルを収穫
- 電子署名を検証し、正しいもの（Valid）をファイルへ出力
- 電子署名の有効期限やファイルが壊れているものはInvalidとして出力されない
- 実運用ではcronにて定期実行を想定

# 収穫したファイルの格納場所

```
root@iw23-saito-ubuntu01:/usr/local/var/cache/rpki-client# pwd
/usr/local/var/cache/rpki-client
root@iw23-saito-ubuntu01:/usr/local/var/cache/rpki-client# ls
0.sb                          r.magellan.ipxo.com           rpki.folf.systems             rpki.sub.apnic.net
ca.nat.moe                    rov-measurements.nlnetlabs.net rpki.luys.cloud              rpki.tools.westconnect.ca
ca.rg.net                     rpki-01.pdxnet.uk             rpki.owl.net                  rpki.xindi.eu
chloe.sobornost.net           rpki.0i1.eu                   rpki.pedjoeang.group          rpki.zappiehost.com
cloudie-repo.rpki.app         rpki.admin.freerangecloud.com rpki.qs.nu                    rsync.krill.cloud
dev.tw                        rpki.afrinic.net              rpki.rand.apnic.net           rsync.paas.rpki.ripe.net
krill.accuristechnologies.ca  rpki.akrn.net                rpki-repo.registro.br         rsync.roa.tohunet.com:3873
krill.ca-bc-01.ssmidge.xyz    rpki.apernet.io               rpki-repository.nic.ad.jp     rsync.rp.ki
krill.rayhaan.net             rpki.apnic.net                rpki.ripe.net                 rsync.rpki.co
krill.stonham.info            rpki.arin.net                 rpki.roa.net                  rsync.rpki.tianhai.link
pub.krill.ausra.cloud         rpkica.mckay.com              rpki-rps.arin.net             sakuya.nat.moe
pub.rpki.win                  rpkica.twnic.tw               rpki-rsync.mnihyc.com         ta
repo.kagl.me                  rpki.cc                       rpki-rsync.us-east-2.amazonaws.com
repo-rpki.idnic.net           rpki.cnnic.cn                 rpki.sailx.co
repository.lacnic.net         rpki.co                       rpki.services.vm.nl.i.bm-x0.w420.net
root@iw23-saito-ubuntu01:/usr/local/var/cache/rpki-client#
```

リポジトリ毎にディレクトリ分割

# rpki-clientのエラーメッセージ

- expire系
  - 証明書等の有効期限切れ
- not valid系
  - 電子署名が検証できない
- network Unreach系
  - リポジトリにアクセスできない
- ASPA系
  - （技術が検討中の）ASPAに関するエラー
- 日本国内を考慮すると
  - “rpki-repository.nic.ad.jp”にエラーが出ていないことが重要
  - 自NWの状態などもチェック

# 正常に終了すると？

```
rpki-client: all files parsed: generating output
Processing time 790 seconds (151 seconds user, 20 seconds system)
Skiplist entries: 0
Route Origin Authorizations: 178733 (17 failed parse, 0 invalid)
AS Provider Attestations: 77 (43 failed parse, 0 invalid)
BGPsec Router Certificates: 3
Certificates: 40423 (1 invalid)
Trust Anchor Locators: 5 (0 invalid)
Manifests: 40422 (91 failed parse, 0 stale)
Certificate revocation lists: 40331
Ghostbuster records: 3
Trust Anchor Keys: 0
Repositories: 89
Cleanup: removed 111 files, 58921 directories
Repository cleanup: kept 4528 and removed 334 superfluous files
VRP Entries: 492292 (480723 unique)
VAP Entries: 34 (34 unique)
```

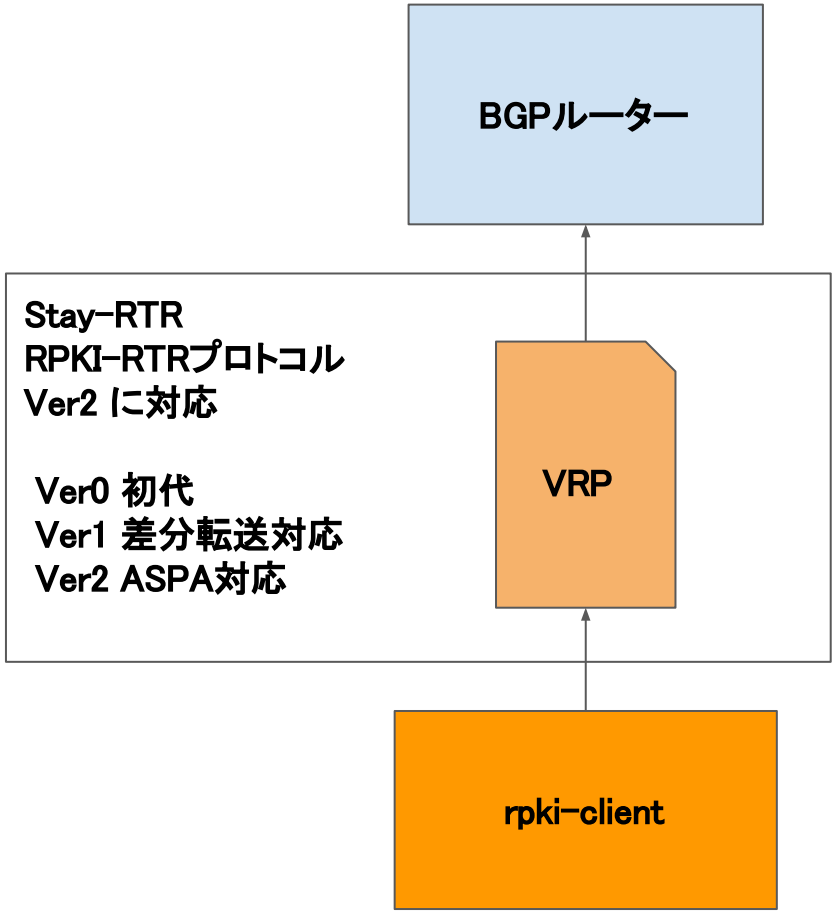
# Stay-RTRのインストール
# （Ubuntuでの操作）

# Stay-RTRとは

Go言語により開発された
ROAキャッシュサーバーの
機能を提供する
オープンソースのライブラリ
(Goに対して”Stay”と言いたいらしい)

今回はrpki-clientで出力された
VRPをBGPルーター(Cat8kv)に
提供する機能を利用する

VRP ＝Validated RPKI Payload
(検証した中身の情報のこと)



# Stay-RTRのインストール

- go環境のインストール

```
cd ~

sudo apt install -y golang-go
```

# Stay-RTRのインストール

- StayRTRのインストール

```
git clone https://github.com/bgp/stayrtr.git
```

# Stay-RTRのインストール

- StayRTRのmake

```
cd stayrtr

make build-stayrt

sudo cp -p dist/stayrtr-* /usr/local/bin/stayrtrr
```

# Stay-RTRのインストール

- stayrtrの起動

```
stayrtr -bind :3323 -cache /usr/local/var/db/rpki-client/json &
（rpki-clientがjsonファイルを生成している）
```

# デバッグツール rtrdump

寡黙のため何も表示されていないが、ファイルに保存される

```
$ pwd
/home/user01/stayrtr
$ go run cmd/rtrdump/rtrdump.go –connect 127.0.0.1:3323
$ cat output.json | more
```

# Stay-RTR 補足

- 差分や更新
  - -cache /usr/local/var/db/rpki-client/jsonに指定したファイルが更新されると更新される。
- rpki-rtrの更新頻度
  - rpki-rtrプロトコルはプロトコルとしては重め
    - 1日1回の更新では反映が遅い？
    - 1時間おきでは？
  - 更新頻度の議論は継続中

# BGPルーターの設定
# （Cat8000Vでの操作）

# ルーターの設定

- rpkiserverとセッションを張る

```
$ ssh admin@IPv4 address
>
> enable

# conf t

(config) router bgp <MyAS-Number>
```

```
(config-router) bgp rpki server tcp <UbuntuIP> port 3323 refresh 3600
```

**enable-Password=Cisco123**

# ルーターの設定

- ピア(経路取得先)の指定

```
(config-router) neighbor 172.24.10.154 remote-as 65041
```

# 構築環境の確認
# （Cat8000Vでの操作）

# BGPルーターの状態確認

- コネクションの確認

```
# show ip bgp rpki servers
```

受け取ったROA数に注目！

```
iw23-goto-c8kv1#show ip bgp rpki servers
BGP SOVC neighbor is 172.24.10.199/3323 connected to port 3323
Flags 64, Refresh time is 3600, Serial number is 68, Session ID is 49689
InQ has 0 messages, OutQ has 0 messages, formatted msg 23
Session IO flags 3, Session flags 4008
 Neighbor Statistics:
  Prefixes 386261
  Connection attempts: 1
  Connection failures: 0
  Errors sent: 0
  Errors received: 0
 --More--
```

# BGPルーターの状態確認

- VRPsの確認

```
# show ip bgp rpki table
```

```
iw23-goto-c8kv1#show ip bgp rpki table
348778 BGP sovc network entries using 55804480 bytes of memory
386261 BGP sovc record entries using 12360352 bytes of memory

Network              Maxlen  Origin-AS  Source  Neighbor
1.0.0.0/24           24      13335      0       172.24.10.199/3323
1.0.4.0/24           24      38803      0       172.24.10.199/3323
1.0.4.0/22           22      38803      0       172.24.10.199/3323
1.0.5.0/24           24      38803      0       172.24.10.199/3323
1.0.6.0/24           24      38803      0       172.24.10.199/3323
1.0.7.0/24           24      38803      0       172.24.10.199/3323
1.0.64.0/18          18      18144      0       172.24.10.199/3323
1.1.1.0/24           24      13335      0       172.24.10.199/3323
1.1.4.0/22           22      4134       0       172.24.10.199/3323
1.1.16.0/20          20      4134       0       172.24.10.199/3323
1.2.9.0/24           24      4134       0       172.24.10.199/3323
1.2.10.0/24          24      4134       0       172.24.10.199/3323
1.2.11.0/24          24      4134       0       172.24.10.199/3323
1.2.12.0/22          22      4134       0       172.24.10.199/3323
1.3.0.0/16           16      4134       0       172.24.10.199/3323
1.6.0.0/22           24      9583       0       172.24.10.199/3323
1.6.4.0/22           24      9583       0       172.24.10.199/3323
1.6.8.0/22           24      9583       0       172.24.10.199/3323
1.6.12.0/24          24      9583       0       172.24.10.199/3323
 --More--
```

先ほど設定した
キャッシュサーバー(Ubuntu)から
受信していることがわかる

# BGPルーターの状態確認

- 経路情報の確認

```
# show ip bgp
# show ip bgp | include ^V (詳細)
先頭がV=Valid、I=Invalid、N=NotFound
```

dstIPに対応する
NextHopや
経由するASなどが
閲覧できる。

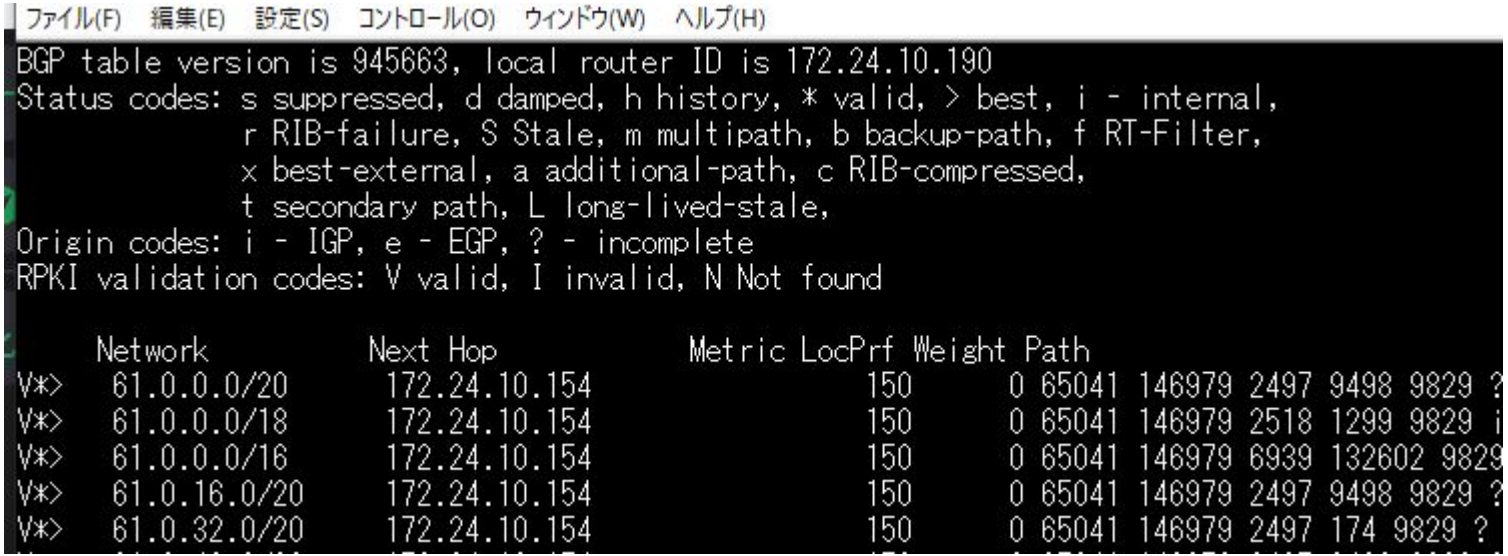
65041はローカルAS
146979はどこの組織？

```
iw23-goto-c8kv1#show ip bgp
BGP table version is 240902, local router ID is 172.24.10.196
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
              t secondary path, L long-lived-stale,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
V*>   61.0.0.0/16      172.24.10.154                          0 65041 146979 6939 132602 9829 i
V*>   61.0.32.0/20     172.24.10.154                          0 65041 146979 2497 174 9829 ?
V*>   61.0.176.0/20    172.24.10.154                          0 65041 146979 2497 174 9829 ?
V*>   61.0.240.0/20    172.24.10.154                          0 65041 146979 2497 1299 64049 55836 9829 i
V*>   61.1.0.0/20      172.24.10.154                          0 65041 146979 2497 174 9829 ?
V*>   61.1.32.0/20     172.24.10.154                          0 65041 146979 2497 174 9829 ?
V*>   61.1.64.0/20     172.24.10.154                          0 65041 146979 2497 174 9829 ?
 --More--
```

# Route-mapによる優先処理

```
route-map RPKI permit 10
 match rpki invalid
 set local-preference 50
!
route-map RPKI permit 20
 match rpki not-found
 set local-preference 100
!
route-map RPKI permit 30
 match rpki valid
 set local-preference 150
!
router bgp 650XX
 bgp log-neighbor-changes
 bgp rpki server tcp 172.24.10.100 port 3323 refresh 3600
 neighbor 172.24.10.154 remote-as 65041
 neighbor 172.24.10.154 route-map RPKI in
```

# show ip bgp




経路情報の先頭のV I NにりLocPrf(ローカルプリファレンス値)が変わっている。(Control PlaneのPollingまで若干時間がかかる場合があり。

# キャッシュ運用のTips

# リポジトリのi-node枯渇に注意

```
user01@iw23-saito-ubuntu01:/$ sudo du -s ./*/ --inodes
1080    ./bin/
311     ./boot/
442     ./dev/
1671    ./etc/
9637    ./home/
31922   ./lib/
790     ./run/
402     ./sbin/
26192   ./snap/
1       ./srv/
54220   ./sys/
20      ./tmp/
479686  ./usr/ (/usr/local/var/にRPKI保存)
4010    ./var/
```

# stayrtrのリセット

```
root@iw23-saito-ubuntu01:/home/user01# ps aux | grep stay
user01     16305  0.6  5.6 1608032 460684 ?      Sl   Nov13  14:30 stayrtr
-bind :3323 -cache /usr/local/var/db/rpki-client/json
root       19792  0.0  0.0   6476  2484 pts/4    S+   16:50   0:00 grep
--color=auto stay
root@iw23-saito-ubuntu01:/home/user01# kill -TERM 16305
root@iw23-saito-ubuntu01:/home/user01# ps aux | grep stay
root       19795  0.0  0.0   6476  2376 pts/4    S+   16:50   0:00 grep
--color=auto stay
root@iw23-saito-ubuntu01:/home/user01#
```

# 付記

## ROVの無効化
## （Cat8000Vでの操作）

# ROVの無効化

- ROVの無効/有効化

```
# conf t
(config) router bgp [MyAs-Number]
(config-router) bgp bestpath prefix-validate disable



# 元に戻す時は
(config-router) no bgp bestpath prefix-validate disable
```

検証情報を利用しない設定です。
ルータは引き続き、キャッシュサーバからVRPをダウンロードします


# ハンズオン終了です。
# みなさまお疲れ様でした!