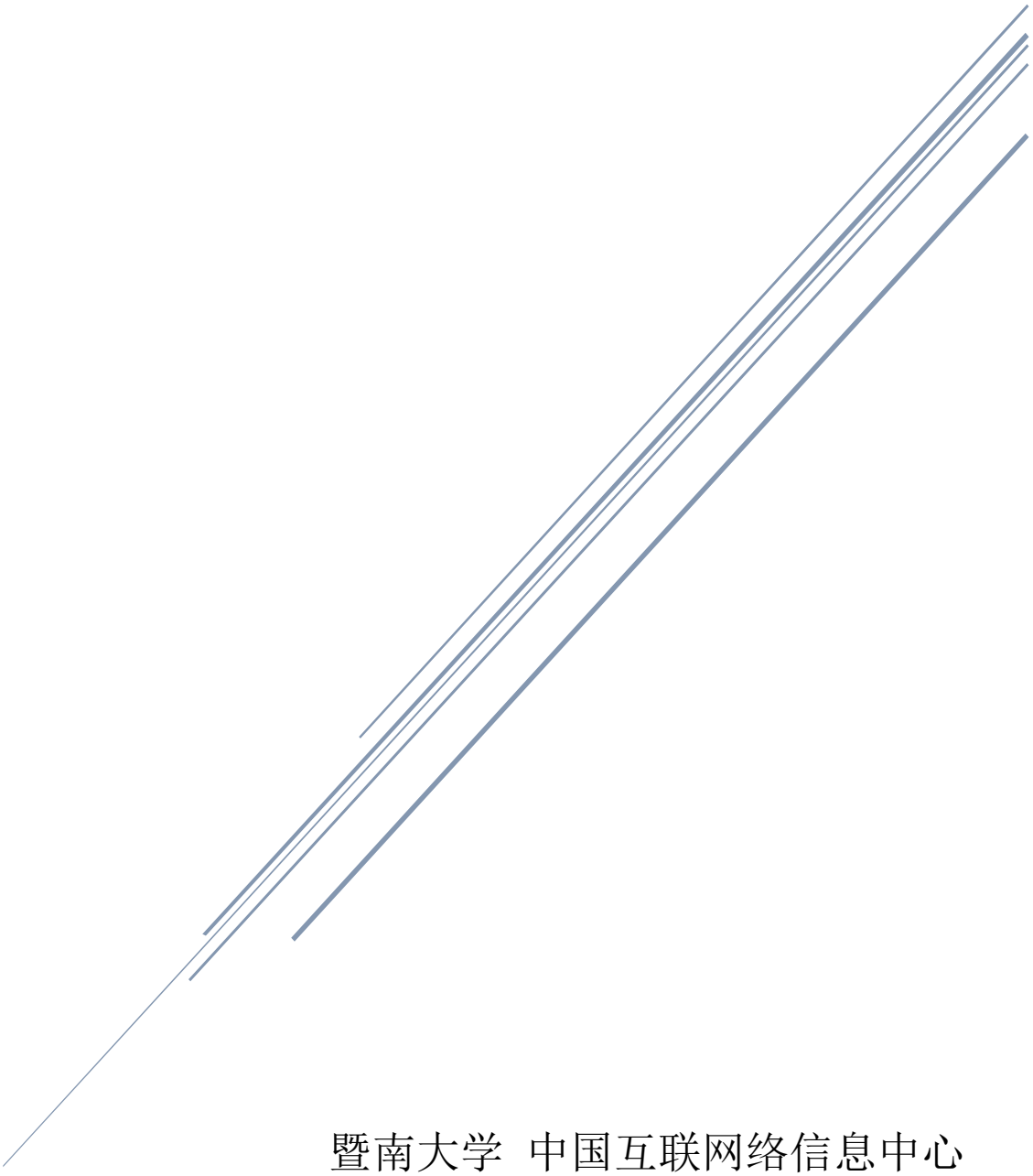
RPKI 测试环境搭建

延志伟 耿光刚 孔凯传 赖泽桐 黄衍铭



暨南大学 中国互联网络信息中心

2021 年 12 月

目录

1 实现方案

1.1 测试环境

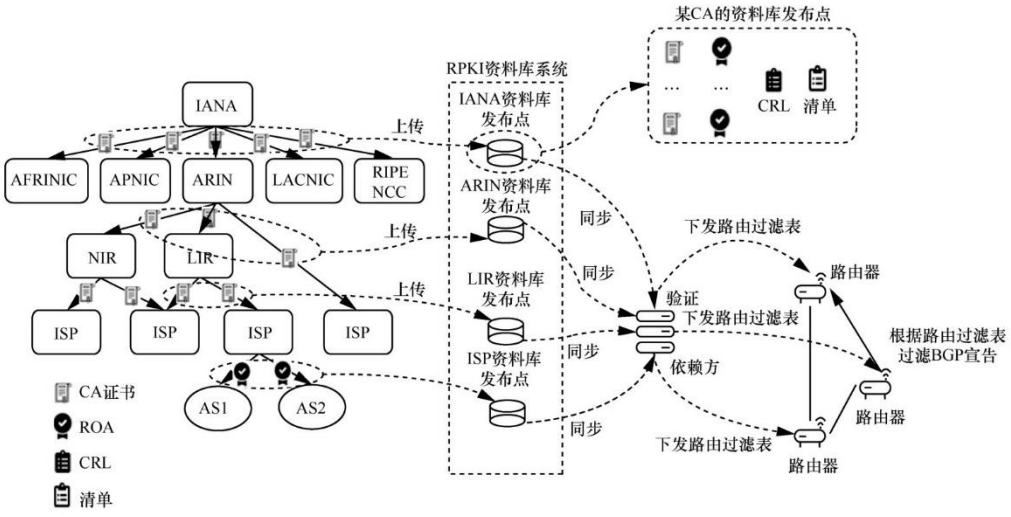
1.1.1 测试实际环境图

如下图所示，RPKI 测试环境由三个子系统组成：

1）CA 系统，负责 RPKI 的证书签发与互联网号码资源由上至下的分配（下图左侧）。

2）RPKI 依赖方(Relying Party)，简称 RP。RPKI 涉及的所有证书都存放至 RPKI 资料库中供 RP 同步（下图中间部分），RP 同步并验证 RPKI 证书和签名对象，而后将验证结果下放。

3）BGP 路由体系，接受 RP 的验证结果作为自己的选路策略依据（下图右侧）。



1.1.2 设备使用情况

实验环境硬件主要有 CA 服务器（二级）、RP 服务器和支持 RPKI 的路由器。设备所用的操作系统类型可分为三种类型：

1）Ubuntu 系统、Debian 系统

2）Freebsd 操作系统

3）其他 Linux 操作系统类型

本报告仅对 Ubuntu16.04 下的安装配置方法进行介绍。

1.1.3 软件使用介绍

CA 系统和 RPKI 依赖方的构建使用的是 rpki.net

在进行 BGP 网络搭建时使用的软件是 GNS3

1.2 CA 系统

1.2.1 安装 CA 系统

1.2.1.1 使用 APT 安装 rpki.net

1）直接使用 APT 无法获得该库的资源，需要先添加该库的 GPG 公钥

wget -q -O /etc/apt/trusted.gpg.d/rpki.gpg https://download.rpki.net/APTng/apt-gpg-key.gpg

2）配置 APT 以使用该库

wget -q -O /etc/apt/sources.list.d/rpki.list https://download.rpki.net/APTng/rpki.xenial.list

3）更新 APT 资源

apt-get update

4）安装 rpki.net

apt install rpki-rp rpki-ca

若在安装时出现以下问题

django.db.utils.OperationalError: could not connect to server: No such file or directory

Is the server running locally and accepting

connections on Unix domain socket "/var/run/postgresql/.s.PGSQL.5432"?

是由于/etc/ssl/certs/ssl-cert-snakeoil.pem 权限不足所导致的，赋予该文件 775 权限，再重启 PostgreSQL 服务，重新安装 rpki.net 即可

若在安装时出现以下问题

/var/lib/dpkg/info/rpki-ca.postinst: 43: /var/lib/dpkg/info/rpki-ca.postinst: hexdump: not found

/var/lib/dpkg/info/rpki-ca.postinst: 43: /var/lib/dpkg/info/rpki-ca.postinst: arithmetic expression: expecting primary: " % 60 "

是由于是缺少 bsdmainutils 模块导致的，安装该模块即可

1.2.2 配置 CA 系统

1.2.2.1 CA 引擎构成

在 RPKI 体系中，CA 引擎主要用于产生证书、ROA、CRL 以及其他 RPKI 对象。CA 引擎主要由以下几部分构成：

1）rpkid：rpki 的守护引擎

2）pubd：发布引擎

3）Irdbd：IR 数据库的实现引擎

4）rpkic：控制 rpkid 和 pubd 的命令行接口

利用上述几种工具可以进行的主要操作包括创建数据库，利用 rpkic 配置父节点和子节点的关系，配置发布客户端同资料库节点的关系，为子节点分配资源和创建 ROA 等。一旦创建过程完成，rpkid 可以自动维护请求数据，包括对父节点发送周期性请求用于查看是否存在变化，根据需求重新签发证书和其他对象等

1.2.2.2 配置 rpki.conf

rpki.conf 是 CA 引擎默认的配置文件，在启动 CA 引擎之前需要对配置文件进行设置。通过以上操作安装 rpki.net，rpki.conf 一般位于/etc/rpki.conf。下面对 rpki.conf 几个关键配置进行介绍

1）handle

每一个资源持有者都需要一个“handle”的名称，handle 的名称不需要唯一性保证，但它是在出现运行问题时排查错误的依据之一，也方便父节点和子节点进行识别。handle 名称可以由 ASCII 字母、数字、连字符、下划线构成。

2）run_rpkid

除非只需要运行 pubd，否则该项的设为 yes。其设置格式如下：

run_rpkid = yes

3）rpkid_server_host

rpkid_server_host 应设置为 rpkid server 的 IP 地址或者是 DNS 主机名

如设置为 rpkid server 的 IP 地址，设置格式如下：

rpkid_server_host = XX.XX.XX.XX

4）irdbd_server_host

一般情形下该值设置为 localhost

irdbd_server_host = localhost

5）run_pubd

对于是否启用 pubd 引擎，一般来说，最佳的选择是在父节点允许的情形下使用父节点的 pubd，这样可以减少发布节点的整体数目，更有利于 RPKI 依赖方获取数据。然而并不是所有的父节点都提供发布服务，或者出于可靠性的考虑需要运行自己的 pubd，又或者因为需要认证私有地址空间或者私有 ASN，这时需要启用 pubd。可设置为

run_pubd = yes

6）pubd_server_host

设置方法同 rpkid_server_host

其余的部分通常不需修改，若要修改参考 RPKI 的 rpki.conf 文档(https://gith

ub.com/dragonresearch/rpki.net/blob/master/doc/manual/12.RPKI.CA.Configuration.md)

对以上文件进行修改后，执行/etc/init.d/xinetd restart 重启服务

1.2.2.3 配置 Rsync

rsync 是类 unix 系统下的数据镜像备份工具，它的主要特性如下：

1）可以镜像保存整个目录树和文件系统

2）可以很容易做到保持原来文件的权限、时间、软硬链接等

3）无须特殊权限即可安装

4）优化的流程，文件传输效率高

5）可使用 rcp、ssh 等方式传输文件，当然也可以通过直接的 socket 连接

6）支持匿名传输，以方便进行网站镜像

在 RPKI 体系中，rsync 主要用于 RPKI 依赖方同 RPKI 资料库之间进行数据同步。其中 rsync 配置文件在/etc/rsyncd.conf，若没有则创建，其内容配置如下：

```
uid     = nobody
gid     = rpki

[rpki]
    use chroot          = no
    read only           = yes
    transfer logging    = yes
    path                = /usr/share/rpki/publication
    comment             = RPKI publication

# 若该服务器是根服务器则设置以下
[tal]
    use chroot          = no
    read only           = yes
```

```
        transfer logging        = yes
        path                    = /usr/share/rpki/rrdp-publication
        comment                 = MyCA TAL
```

接着查看/etc/xinetd.d/rsync，若不存在需要创建

```
service rsync
{
        disable         = no
        socket_type     = stream
        port            = 873
        protocol        = tcp
        wait            = no
        user            = root
        server          = /usr/bin/rsync
        server_args     = --daemon
        log_on_failure  += USERID
}
```

port 是 rsync 服务运行的端口

对以上文件进行修改后，执行/etc/init.d/xinetd restart 重启服务

1.2.2.4 创建 root 节点

执行/etc/init.d/rpki-ca restart 重启 rpki-ca 服务

下面的步骤建议创建一个 CA-data 文件夹储存生成的文件

1）首先获得 root 节点的 handle，也可以直接到/etc/rpki.conf 中查看（下文用 root_handle 代表 root 节点的 handle 的值）

```
fgrep handle /etc/rpki.conf
```

2）接着执行 rpkic 进入 rpkic 的命令行中，初始化 bpki

initialize_server_bpki

3）创建身份，生成 root_handle.identity.xml

create_identity root_handle

4）选择身份

select_identity root_handle

5）配置 root 节点，可以使用--resources 对 AS 号码以及 IPv4 和 IPv6 进行配置，中间逗号分隔，生成 root_handle.root_handle.repository-request.xml

configure_root --resources AS0,AS1,IPv4.0,IPv4.1,IPv4.2,IPv6.0

6）配置发布客户端，生成 root_handle.repository-response.xml

configure_publication_client root_handle.root_handle.repository-request.xml

7）配置储存库

configure_repository root_handle.repository-response.xml

等待一段时间后，/usr/share/rpki/publication/目录下会生成名为 root_handle 的文件夹，证书保存在该目录

8）生成 root 证书和 TAL，然后将证书移动到 https 服务中命名为 rrdp 的目录下，通常是/usr/share/rpki/rrdp-publication/

rpkic extract_root_certificate

rpkic extract_root_tal

9）检查对比 root 证书和 TAL 的密钥

openssl x509 -inform DER -in root_handle.cer -noout -pubkey | openssl pkey -pubin -outform DER -out root_handle.cer.key

sed -n '/^$/,$p' root_handle.tal | openssl enc -d -a -out root_handle.tal.key

diff -qs root_handle.cer.key root_handle.tal.key

1.2.2.5 创建节点

执行/etc/init.d/rpki-ca restart 重启 rpki-ca 服务

下面的步骤建议创建一个 CA-data 文件夹储存生成的文件

1）首先获得节点的 handle，也可以直接到/etc/rpki.conf 中查看

fgrep handle /etc/rpki.conf

2）初始化 bpki

rpkic initialize_server_bpki

3）创建身份，生成 handle.identity.xml

rpkic create_identity handle

4）将 handle.identity.xml 移动到 rsync 的 rpki 目录下

cp handle.identity.xml /usr/share/rpki/publication

1.2.2.6 配置节点父子关系

1）root 节点获取子节点身份

rsync -avz rsync://rpkic@handle_ip/rpki/handle.identity.xml /root/CA-data

2）root 节点执行命令，生成 root_handle.handle.parent-response.xml，并移动到 tal 目录下

rpkic configure_child handle.identity.xml

3）子节点获取 root_handle.handle.parent-response.xml

rsync -avz rsync://rpkic@root_ip/tal/root_handle.handle.parent-response.xml /root/CA-data

4）子节点配置父子关系, 生成 handle.root_handle.repository-request.xml

rpkic configure_parent root_handle.handle.parent-response.xml

5）子节点配置发布客户端，生成 handle.repository-response.xml

rpkic configure_publication_client handle.root_handle.repository-request.xml

5）子节点配置资料库

rpkic configure_repository handle.repository-response.xml

6）为子节点分配 AS 号码资源，root 节点运行

rpkic load_asns asn.csv

asn.csv 的格式为:

handle 42-44

handle 62255

7）为子节点分配 IP 地址资源，root 节点运行

rpkic load_prefixes prefix.csv

prefix.csv 的格式为:

handle 192.0.2.1-192.0.2.33

handle 192.0.2.44-192.0.2.100

8）若需子节点创建 ROA，则子节点运行

rpkic load_roa_requests roa.csv

roa.csv 的格式为:

192.0.2.0/24 62255 24

各字段的含义分别为:IP 地址前缀、为前缀发起路由的 AS 号、前缀长度

每个 AS 号可以为多个 IP 前缀发起路由，但不同 AS 号发起的路由不能写在同一个 csv 文件中

1.3 Relying Party

1.3.1 rcynic

rcynic 是最基础的验证工具，是 RPKI 验证工作的真正执行者，包括检查语法、签名、过期时间、是否与 RPKI 对象的约束文件一致等。其他的 relying party 工具会将 rcynic 的输出结果作为输入。rcynic 的配置参数在 rpki.conf 的【rcynic】部分进行设置

1.3.1.1 启动 rcynic 服务

1）以用户 rpki 打开 crontab

crontab -e -u rpki

2）修改定时任务为

MAILTO=root

xx * * * * /usr/bin/python /usr/bin/rcynic-cron

xx 为邻近的分钟数以尽快启动 rcynic

3）重启 crontab

/etc/init.d/cron restart

1.3.2 rpki-rp

rtr-origin 依靠 rcynic 收集并验证 RPKI 数据，rtr-origin 的功能是为路由验证提供轻量级格式的数据。

其中 rcynic-cron 脚本会自动将 rcynic 输出结果处理成 rtr-origin 可用的数据文件。在 freebsd、Debian、ubuntu 平台会自动的为 rtr-orgin 服务器建立一个 TCP 监听器。当然也可以建立其他协议类型的监听器。

建立的监听器会以“--server”的模式调用 rtr-origin。rtr-origin 可以在 inetd、xinetd、sshd 下运行。RFC 6810 规定了该服务启动在 323 端口上。

rtr-origin 还有其他两种模式可以用来 debug：

1、--client 模式：调用方式$ rtr-origin --client tcp 该模式除了用于 debug 目前没有其他的用途。

2、--show 模式：以文本的方式对验证并加工后的 RPKI 数据进行展示。

1.3.2.1 配置 rpki-rp 服务

查看/etc/xinetd.d/rpki-rtr，若不存在需要创建

service rpki-rtr

{

```
        type            = UNLISTED
        flags           = IPv4
        socket_type     = stream
        protocol        = tcp
        port            = 323
        wait            = no
        user            = rpki
        server          = /usr/bin/rpki-rtr
        server_args     = server  /var/rcynic/rpki-rtr
}
```

设置为以上后执行/etc/init.d/xinetd restart 重启服务

1.4 BGP 体系

1.4.1 搭建工具

本文使用 GNS3 作为 BGP 网络的搭建工具。GNS3 是一款具有图形化界面的可以运行在多平台（包括 Windows, Linux, and MacOS 等）的网络虚拟软件，可以用于虚拟体验 Cisco 网际操作系统 IOS 或者是检验将要在真实的路由器上部署实施的相关配置。

简单说来 GNS3 是 dynamips 的一个图形前端，相比直接使用 dynamips 这样的虚拟软件要更容易上手和更具有可操作性。GNS3 整合了如下的软件：

Dynamips：一款可以让用户直接运行 Cisco 系统(IOS)的模拟器

Dynagen：是 Dynamips 的文字显示前端

Pemu：PIX 防火墙设备模拟器。

Winpcap：windows 平台下一个免费、公共的网络访问系统。增加 winpcap 目的在于为 win32 应用程序提供访问网络底层的能力。利用 GNS3 可以设计优秀的网络拓扑结构，模拟 Cisco 路由设备和 PIX 防火墙，仿真简单的 Ethernet、ATM 和帧中继交换机，同时能够装载和保存为 Dynamips 的配置格式，也就是说对于

使用 dynamips 内核的虚拟软件具有较好的兼容性支持一些文件格式（JPEG、PNG、BMP 和 XPM）的导出。

1.4.2 安装与使用

1.4.2.1 安装 GNS3

到 GNS3 官网(https://www.gns3.com/)下载 GNS3 安装即可

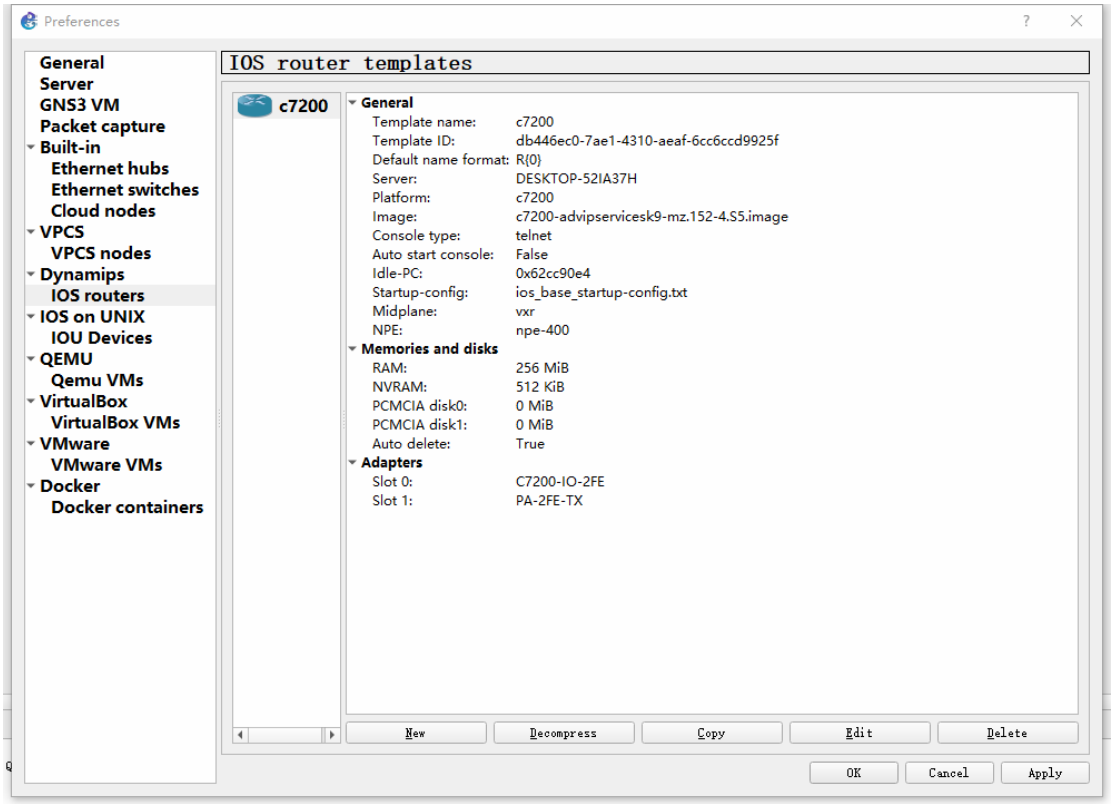
1.4.2.2 支持 RPKI 的路由器类型

支持 RPKI 的路由器类型包括 Cisco、Juniper、Quagga。其中 Juniper 路由器从版本 12.2 开始即可支持 RPKI 的功能，支持 RPKI 配置的 cisco 路由器型号如下表所示：

路由器类型	XR4.2.1	XR5.1.1	XE3.5
具体型号	CRS-1,CRS-3, CRS-x ASR9000 c12000	NCS6000 XRv	c7200,c7600,ASR1000 CSR1000v ASR901,ASR903,ASR907 ME3600,ME3800 ASR1000 & CSR1000v

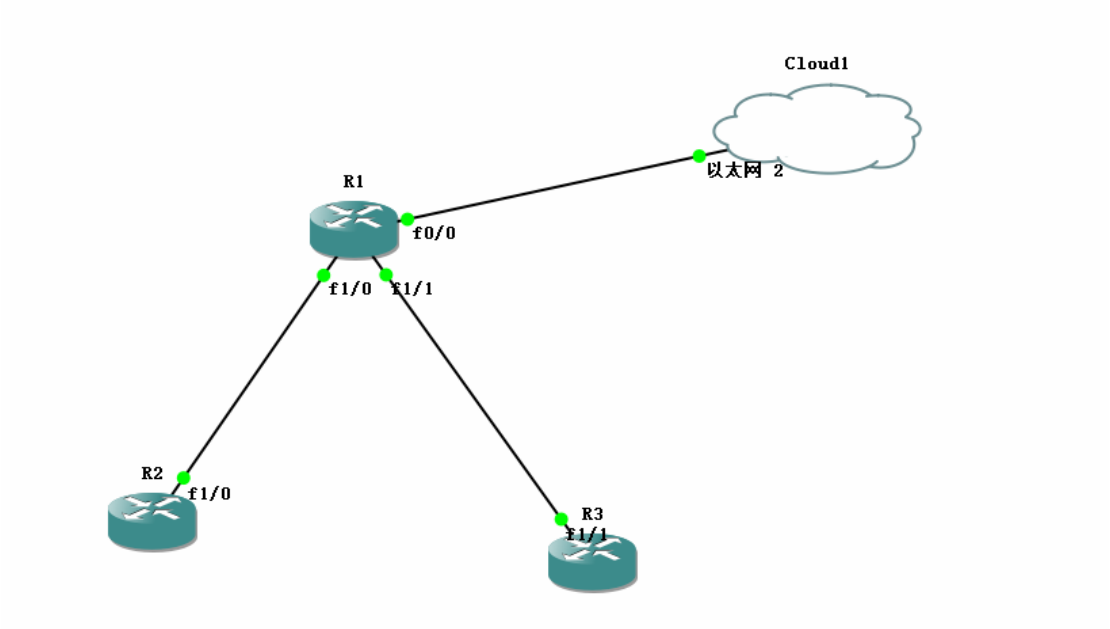
本报告使用的路由器类型是 c7200，可以在此处下载到 c7200 路由器镜像(https://drive.google.com/file/d/1iAt4dN6mpgbKHrYHjwHhvu6sEaQROKNk/view)

路由器镜像的导入在 GNS3 中选择 Edit->Preferences 打开首选项窗口，选择 IOS router，通过 New 导入新镜像。Idle 值靠软件自动计算即可



1.4.3 搭建拓扑环境

最终通过 GNS3 搭建的 BGP 拓扑环境图如下所示：



其搭建主要包含以下几步：

1）配置 Cloud1

R1 在连接时连接到 C1 的以太网接口即可

2）配置 R2

R2 的配置信息如下图所示：

```
interface FastEthernet0/0
 no ip address
 shutdown
 speed auto
 duplex auto
!
interface FastEthernet0/1
 no ip address
 shutdown
 speed auto
 duplex auto
!
interface FastEthernet1/0
 ip address 10.1.1.2 255.255.255.0
 speed auto
 duplex auto
!
interface FastEthernet1/1
 no ip address
 shutdown
 speed auto
 duplex auto
!
router bgp 43532
 synchronization
 bgp log-neighbor-changes
 network 10.1.1.0
 neighbor 10.1.1.1 remote-as 65000
```

3）配置 R3

R3 的配置信息如下图所示：

```
interface FastEthernet0/0
 no ip address
 shutdown
 speed auto
 duplex auto
!
interface FastEthernet0/1
 no ip address
 shutdown
 speed auto
 duplex auto
!
interface FastEthernet1/0
 no ip address
 shutdown
 speed auto
 duplex auto
!
interface FastEthernet1/1
 ip address 10.1.255.2 255.255.255.0
 speed auto
 duplex auto
!
router bgp 64800
 synchronization
 bgp log-neighbor-changes
 network 10.1.255.0
 neighbor 10.1.255.1 remote-as 65000
```

4）配置 R1

（1）首先配置 R1 的各个接口,R1 的各个接口配置信息如图：

```
R1#show ip int br
Interface              IP-Address      OK? Method Status                Protocol
FastEthernet0/0        192.168.1.192   YES DHCP   up                    up
FastEthernet1/0        10.1.1.1        YES NVRAM  up                    up
FastEthernet1/1        10.1.255.1      YES NVRAM  up                    up
R1#
```

连接以太网的接口 f0/0 通过 DHCP 自动配置 ip

（2）配置路由器连接至 RPKI server，其 IP 地址为 47.92.141.139，端口号为 10323

```
router bgp 65000
 bgp log-neighbor-changes
 bgp rpki server tcp 47.92.141.139 port 10323 refresh 3600
```

（3）基于路由信息的验证状态制定路由策略

```
route-map rpki-loc-pref permit 10
 match rpki invalid
 set local-preference 90
!
route-map rpki-loc-pref permit 20
 match rpki not-found
 set local-preference 100
!
route-map rpki-loc-pref permit 30
 match rpki invalid
 set local-preference 110
```

（4）将制定的路由策略应用到 BGP 邻居

```
 neighbor 10.1.1.2 remote-as 43532
 neighbor 10.1.255.2 remote-as 64800
 !
 address-family ipv4
  bgp bestpath prefix-validate allow-invalid
  network 10.1.1.0
  network 10.1.255.0
  network 192.168.1.0
  neighbor 10.1.1.2 activate
  neighbor 10.1.1.2 route-map rpki-loc-pref in
  neighbor 10.1.255.2 activate
  neighbor 10.1.255.2 route-map rpki-loc-pref in
```

（5）测试 bgp 路由器与 RPKI server 的连接是否建立

```
R1#show ip bgp rpki server
BGP SOVC neighbor is 47.92.141.139/10323 connected to port 10323
Flags 64, Refresh time is 3600, Serial number is 1638793396, Session ID is 34371
InQ has 0 messages, OutQ has 0 messages, formatted msg 1
Session IO flags 3, Session flags 4008
 Neighbor Statistics:
  Prefixes 211797
  Connection attempts: 2
  Connection failures: 1
  Errors sent: 0
  Errors received: 0
```

```
R1#show ip bgp rpki table
194636 BGP sovc network entries using 17127968 bytes of memory
211797 BGP sovc record entries using 4235940 bytes of memory

Network            Maxlen  Origin-AS  Source  Neighbor
1.0.0.0/24         24      13335      0       47.92.141.139/10323
1.0.4.0/24         24      38803      0       47.92.141.139/10323
1.0.4.0/22         22      38803      0       47.92.141.139/10323
1.0.5.0/24         24      38803      0       47.92.141.139/10323
1.0.6.0/24         24      38803      0       47.92.141.139/10323
1.0.7.0/24         24      38803      0       47.92.141.139/10323
1.1.1.0/24         24      13335      0       47.92.141.139/10323
1.1.4.0/22         22      4134       0       47.92.141.139/10323
1.1.16.0/20        20      4134       0       47.92.141.139/10323
1.2.9.0/24         24      4134       0       47.92.141.139/10323
1.2.10.0/24        24      4134       0       47.92.141.139/10323
1.2.11.0/24        24      4134       0       47.92.141.139/10323
1.2.12.0/22        22      4134       0       47.92.141.139/10323
1.3.0.0/16         16      4134       0       47.92.141.139/10323
1.6.0.0/22         24      9583       0       47.92.141.139/10323
1.6.4.0/22         24      9583       0       47.92.141.139/10323
1.6.8.0/22         24      9583       0       47.92.141.139/10323
1.6.12.0/24        24      9583       0       47.92.141.139/10323
```

由图可见， BGP 路由器已经通过 10323 端口连接至 RPKI server 47.92.141.139，并且获取了 RPKI 数据（包括 IP 地址前缀、前缀长度、为前缀发起路由的 AS 号等内容）以用于进行路由验证

（6）查看制定的路由策略是否应用到路由中

```
R1#show ip bgp 192.168.1.0
BGP routing table entry for 192.168.1.0/24, version 0
Paths: (1 available, no best path)
  Not advertised to any peer
  Refresh Epoch 1
  Local
    0.0.0.0 (inaccessible) from 0.0.0.0 (10.1.255.1)
      Origin IGP, metric 0, localpref 100, weight 32768, valid, sourced, local
      path 67A1B57C RPKI State valid
      rx pathid: 0, tx pathid: 0
```

由图可见，在 IP 地址前缀 192.168.1.0 的 bgp 路由信息中已经添加了 RPKI 相关内容。

项目支持

北京市科技新星计划项目

Z191100001119113