

Protecting the Integrity of Internet Routing:

Border Gateway Protocol (BGP) Route Origin Validation

Volume C:
How-To Guides

William Haag

Applied Cybersecurity Division
Information Technology Laboratory

Doug Montgomery

Advanced Networks Technology Division
Information Technology Laboratory

Allen Tan

The MITRE Corporation
McLean, VA

William C. Barker

Dakota Consulting
Silver Spring, MD

August 2018

DRAFT

This publication is available free of charge from:

<https://www.nccoe.nist.gov/projects/building-blocks/secure-inter-domain-routing>

DISCLAIMER

Certain commercial entities, equipment, products, or materials may be identified in this document in order to describe an experimental procedure or concept adequately. Such identification is not intended to imply recommendation or endorsement by NIST or NCCoE, nor is it intended to imply that the entities, equipment, products, or materials are necessarily the best available for the purpose.

National Institute of Standards and Technology Special Publication 1800-14C, Natl. Inst. Stand. Technol. Spec. Publ. 1800-14C, 61 pages, (August 2018), CODEN: NSPUE2

FEEDBACK

You can improve this guide by contributing feedback. As you review and adopt this solution for your own organization, we ask you and your colleagues to share your experience and advice with us.

Comments on this publication may be submitted to: sidr-nccoe@nist.gov.

Public comment period: August 30, 2018 through October 15, 2018

All comments are subject to release under the Freedom of Information Act (FOIA).

National Cybersecurity Center of Excellence
National Institute of Standards and Technology
100 Bureau Drive
Mailstop 2002
Gaithersburg, MD 20899
Email: nccoe@nist.gov

NATIONAL CYBERSECURITY CENTER OF EXCELLENCE

The National Cybersecurity Center of Excellence (NCCoE), a part of the National Institute of Standards and Technology (NIST), is a collaborative hub where industry organizations, government agencies, and academic institutions work together to address businesses' most pressing cybersecurity issues. This public-private partnership enables the creation of practical cybersecurity solutions for specific industries, as well as for broad, cross-sector technology challenges. Through consortia under Cooperative Research and Development Agreements (CRADAs), including technology partners—from Fortune 50 market leaders to smaller companies specializing in IT security—the NCCoE applies standards and best practices to develop modular, easily adaptable example cybersecurity solutions using commercially available technology. The NCCoE documents these example solutions in the NIST Special Publication 1800 series, which maps capabilities to the NIST Cybersecurity Framework and details the steps needed for another entity to recreate the example solution. The NCCoE was established in 2012 by NIST in partnership with the State of Maryland and Montgomery County, Md.

To learn more about the NCCoE, visit <https://www.nccoe.nist.gov/>. To learn more about NIST, visit <https://www.nist.gov>.

NIST CYBERSECURITY PRACTICE GUIDES

NIST Cybersecurity Practice Guides (Special Publication Series 1800) target specific cybersecurity challenges in the public and private sectors. They are practical, user-friendly guides that facilitate the adoption of standards-based approaches to cybersecurity. They show members of the information security community how to implement example solutions that help them align more easily with relevant standards and best practices, and provide users with the materials lists, configuration files, and other information they need to implement a similar approach.

The documents in this series describe example implementations of cybersecurity practices that businesses and other organizations may voluntarily adopt. These documents do not describe regulations or mandatory practices, nor do they carry statutory authority.

ABSTRACT

The Border Gateway Protocol (BGP) is the default routing protocol to route traffic among internet domains. While BGP performs adequately in identifying viable paths that reflect local routing policies and preferences to destinations, the lack of built-in security allows the protocol to be exploited by route hijacking. Route hijacking occurs when an entity accidentally or maliciously alters an intended route. Such attacks can (1) deny access to internet services, (2) detour internet traffic to permit eavesdropping and to facilitate on-path attacks on end points (sites), (3) misdeliver internet network traffic to malicious end points, (4) undermine internet protocol (IP) address-based reputation and filtering systems, and (5) cause routing instability in the internet. This document describes a security platform that

demonstrates how to improve the security of inter-domain routing traffic exchange. The platform provides route origin validation (ROV) by using the Resource Public Key Infrastructure (RPKI) in a manner that mitigates some misconfigurations and malicious attacks associated with route hijacking. The example solutions and architectures presented here are based upon standards-based, open-source, and commercially available products.

KEYWORDS

AS, autonomous systems, BGP, Border Gateway Protocol, DDoS, denial-of-service (DoS) attacks, internet service provider, ISP, Regional Internet Registry, Resource Public Key Infrastructure, RIR, ROA, route hijack, route origin authorization, route origin validation, routing domain, ROV, RPKI

ACKNOWLEDGMENTS

We are grateful to the following individuals for their generous contributions of expertise and time.

Name	Organization
Tim Battles	AT&T
Jay Borkenhagen	AT&T
Chris Boyer	AT&T
Nimrod Levy	AT&T
Kathryn Condello	CenturyLink
Christopher Garner	CenturyLink
Peter Romness	Cisco Systems
Tony Tauber	Comcast
Jonathan Morgan	Juniper Networks
Carter Wyant	Juniper Networks
Oliver Borchert	NIST ITL Advanced Networks Technologies Division

Name	Organization
Katikalapudi Sriram	NIST ITL Advanced Networks Technologies Division
Sean Morgan	Palo Alto Networks
Tom Van Meter	Palo Alto Networks
Andrew Gallo	The George Washington University
Sophia Applebaum	The MITRE Corporation
Yemi Fashina	The MITRE Corporation
Susan Prince	The MITRE Corporation
Susan Symington	The MITRE Corporation

○

The Technology Partners/Collaborators who participated in this build submitted their capabilities in response to a notice in the Federal Register. Respondents with relevant capabilities or product components were invited to sign a Cooperative Research and Development Agreement (CRADA) with NIST, allowing them to participate in a consortium to build this example solution. We worked with:

Technology Partner/Collaborator	Build Involvement
AT&T	Subject Matter Expertise
CenturyLink	1 gigabit per second (Gbps) Ethernet Link Subject Matter Expertise
Cisco	7206 VXR Router v15.2 ISR 4331 Router v16.3 2921 Router v15.2 IOS XRv 9000 Router v6.4.1 Subject Matter Expertise
Comcast	Subject Matter Expertise

Technology Partner/Collaborator	Build Involvement
Juniper Networks	MX80 3D Universal Edge Router v15.1R6.7 Subject Matter Expertise
Palo Alto Networks	Palo Alto Networks Next-Generation Firewall PA-5060 v7.1.10 Subject Matter Expertise
The George Washington University	Subject Matter Expertise

1	Contents	
2	1 Introduction.....	1
3	1.1 Practice Guide Structure	1
4	1.2 Build Overview.....	2
5	1.3 Typographic Conventions.....	6
6	2 Product Installation Guides.....	6
7	2.1 RPKI Validators	7
8	2.1.1 RIPE NCC RPKI Validator Configuration/Installation	7
9	2.1.2 Dragon Research RPKI.net Validator Configuration/Installation	8
10	2.2 RPKI CA and Repository	9
11	2.2.1 Dragon Research RPKI.net CA and Repository Configuration/Installation	9
12	2.3 BGP-SRx Software Suite.....	15
13	2.4 Firewalls.....	16
14	2.5 Test Harness Topology Configuration	18
15	2.5.1 RTR 1-1 Configuration – Cisco	18
16	2.5.2 RTR 2-1 Configuration – Cisco	21
17	2.5.3 RTR 2-2 Configuration – Cisco	23
18	2.5.4 RTR 1-1 Configuration – Juniper	24
19	2.5.5 RTR 2-1 Configuration – Juniper	27
20	2.5.6 RTR 2-2 Configuration – Juniper	29
21	2.5.7 Traffic Generator BIO Configuration	31
22	2.6 Live Data Configuration	35
23	2.6.1 CenturyLink Configuration Router AS 65501 – Cisco	35
24	2.6.2 Router AS 65500 Configuration – Cisco.....	37
25	2.6.3 Router 65501 Configuration – Cisco.....	40
26	2.6.4 Router AS 65502 Configuration – Juniper	43
27	2.6.5 Router AS 65503 Configuration – Cisco.....	45
28	2.6.6 Router AS 65504A Configuration – Cisco	48
29	2.6.7 Router AS 65504B Configuration – Cisco	50

30	2.6.8 Router AS 65505 Configuration – Juniper	51
31	2.6.9 Router AS 65507 Configuration – Cisco.....	53
32	2.6.10 Router AS 65508 Configuration – Cisco.....	55
33	2.6.11 Cisco IOS XRv Router Configuration	56

34 **List of Figures**

35	Figure 1-1 Test Harness Environment for SIDR RPKI-Based ROV Solution Testing.....	4
36	Figure 1-2 Live Data Environment for SIDR RPKI-Based ROV Solution Testing.....	5
37	Figure 2-1 Palo Alto Firewall Configuration	17

38 1 Introduction

39 The following guides show information technology (IT) professionals and security engineers how we
40 implemented the example Secure Inter-Domain Routing (SIDR) Project solution for Resource Public Key
41 Infrastructure (RPKI)-based route origin validation (ROV). We cover all of the products employed in this
42 reference design. We do not recreate the product manufacturers' documentation, which is presumed to
43 be widely available. Rather, these guides show how we incorporated the products together in our
44 environment.

45 Note: These are not comprehensive tutorials. There are many possible service and security
46 configurations for these products that are out of scope for this reference design.

47 1.1 Practice Guide Structure

48 This National Institute of Standards and Technology (NIST) Cybersecurity Practice Guide demonstrates a
49 standards-based reference design and provides users with the information they need to replicate the
50 SIDR RPKI-based ROV solution. This reference design is modular and can be deployed in whole or in
51 parts.

52 NIST Special Publication (SP) 1800-14 contains three volumes:

- 53 ▪ NIST SP 1800-14A: *Executive Summary*
- 54 ▪ NIST SP 1800-14B: *Approach, Architecture, and Security Characteristics* – what we built and why
- 55 ▪ NIST SP 1800-14C: *How-To Guides* – instructions for building the example solution (**you are**
56 **here**)

57 Depending on your role in your organization, you might use this guide in different ways:

58 **Business decision makers, including chief security and technology officers**, will be interested in the
59 *Executive Summary* (NIST SP 1800-14A), which describes:

- 60 ▪ The challenges that enterprises face in implementing and maintaining route origin validation
- 61 ▪ An example solution built at the National Cybersecurity Center of Excellence (NCCoE)
- 62 ▪ Benefits of adopting the example solution

63 **Technology or security program managers** who are concerned with how to identify, understand, assess,
64 and mitigate risk will be interested in NIST SP 1800-14B, which describes what we did and why. The
65 following sections will be of particular interest:

- 66 ▪ Section 4.4.3, Risks, provides a description of the risk analysis we performed
- 67 ▪ Section 4.4.4, Cybersecurity Framework Functions, Categories, and Subcategories Addressed by
68 the Secure Inter-Domain Routing Project, maps the security characteristics of this example
69 solution to cybersecurity standards and best practices

70 If you are a technology or security program manager, you might share the *Executive Summary*, NIST SP
71 1800-14A, with your leadership team members to help them understand the importance of adopting
72 the standards-based SIDR RPKI-based ROV solution.

73 IT professionals who want to implement an approach like this can use the How-To portion of the guide,
74 NIST SP 1800-14C, to replicate all or parts of the build created in our lab. The How-To guide provides
75 specific product installation, configuration, and integration instructions for implementing the example
76 solution. We do not recreate the product manufacturers' documentation, which is generally widely
77 available. Rather, we show how we incorporated the products together in our environment to create an
78 example solution.

79 This guide assumes that IT professionals have experience implementing security products within the
80 enterprise. While we have used a suite of commercial products to address this challenge, it is not NIST
81 policy to endorse any particular products. Your organization can adopt this solution or one that adheres
82 to these guidelines in whole, or you can use this guide as a starting point for tailoring and implementing
83 parts of an RPKI-based ROV solution. Your organization's security experts should identify the products
84 that will best integrate with your existing tools and IT system infrastructure. We hope that you will seek
85 products that are congruent with applicable standards and best practices. Section 4.5, Technologies, of
86 NIST SP 1800-14B lists the products that we used and maps them to the cybersecurity controls provided
87 by this reference solution.

88 A NIST Cybersecurity Practice Guide does not describe "the" solution, but a possible solution. This is a
89 draft guide. We seek feedback on its contents and welcome your input. Comments, suggestions, and
90 success stories will improve subsequent versions of this guide. Please contribute your thoughts to [sidr-
91 nccoe@nist.gov](mailto:sidr-nccoe@nist.gov).

92 **1.2 Build Overview**

93 This NIST Cybersecurity Practice Guide addresses the challenge of using existing protocols to improve
94 the security of inter-domain routing traffic exchange in a manner that mitigates accidental and malicious
95 attacks associated with route hijacking. It implements and follows various Internet Engineering Task
96 Force (IETF) Request for Comments (RFC) documents that define RPKI-based Border Gateway Protocol
97 (BGP) ROV, such as [RFC 6480](#), [RFC 6482](#), [RFC 6811](#), and [RFC 7115](#), as well as recommendations of [NIST](#)

98 [SP 800-54](#), *Border Gateway Protocol Security*. To the extent practicable from a system composition point
99 of view, the security platform design, build, and test processes have followed [NIST SP 800-160](#), *Systems*
100 *Security Engineering: Considerations for a Multidisciplinary Approach in the Engineering of Trustworthy*
101 *Secure Systems*.

102 The ROV capabilities demonstrated by the proof-of-concept implementation described in this Practice
103 Guide improve inter-domain routing security by using standards-conformant security protocols to
104 enable an entity that receives a route advertisement to validate whether the autonomous system (AS)
105 that has originated it is in fact authorized to do so.

106 In the NCCoE lab, the team built an environment that resembles portions of the internet. The SIDR lab
107 architecture is depicted in [Figure 1-1](#) and [Figure 1-2](#). It consists of virtual and physical hardware, physical
108 links to ISPs, and access to the Regional Internet Registries (RIRs). The physical hardware mainly consists
109 of the routers performing ROV, workstations providing validator capabilities, and firewalls that protect
110 the lab infrastructure. The virtual environment hosts the RPKI repositories, validators, and caches used
111 for both the hosted and delegated RPKI scenarios. The architecture is organized into separate virtual
112 local area networks (VLANs), each of which is designed to represent a different AS. For example, VLAN 1
113 represents an ISP with AS 64501, VLAN 2 represents the enterprise network of an organization with AS
114 64502, and VLAN 3 represents an ISP with AS 64503.

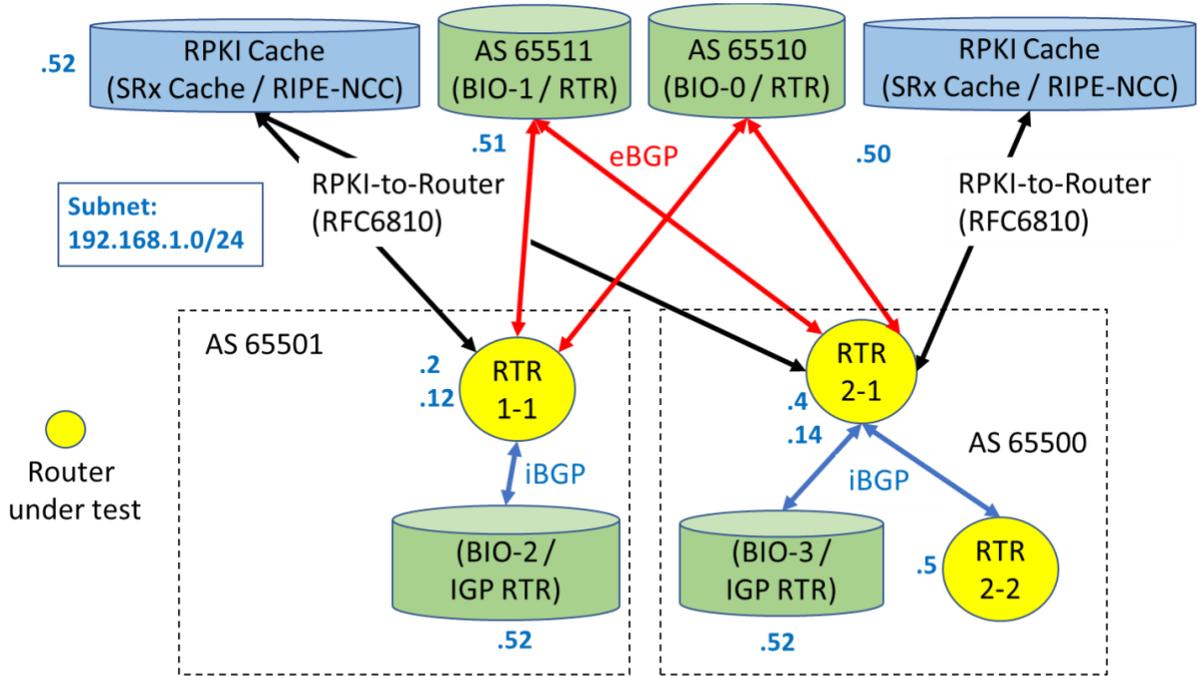
115 The configurations in this document provide a baseline for completing all the test cases that were
116 performed for the project.

117 There are two environments that are used: test harness and live data.

- 118 ▪ The test harness environment consists of physical/virtual routers, a lab RPKI repository, RPKI
119 validators, and simulation tools (or test harness). The physical and virtual routers in this
120 environment are from Cisco and Juniper. The lab RPKI repository is configured using the
121 RPKI.net tool. The RPKI caches in this environment are the Réseaux IP Européens Network
122 Coordination Centre (RIPE NCC) validator and the RPKI.net validator. The test harness simulates
123 BGP routers sending and receiving advertisements and emulates RPKI data being sent from
124 validators/caches. There are two components of the test harness: the BGPSEC-IO (BIO) traffic
125 generator and collector, which produces BGP routing data, and the SRx-RPKI validator cache test
126 harness, which simulates RPKI caches.
- 127 ▪ The live data environment leverages many of the same components from the test harness
128 environment. The difference is that this environment leverages live data from the internet,
129 rather than uses emulated BGP advertisements and RPKI data. The physical and virtual routers
130 in this environment are from Cisco and Juniper. The lab RPKI repository is configured using the
131 RPKI.net tool. Repositories from the RIRs (American Registry for Internet Numbers [ARIN], RIPE
132 NCC, African Network Information Center [AFRINIC], Latin America and Caribbean Network
133 Information Center [LACNIC], and Asia-Pacific Network Information Center [APNIC]) are also
134 used to receive real-world route origin authorization (ROA) data. The RPKI caches in this

135 environment are the RIPE NCC validator and the RPKI.net validator. A physical wide area
 136 network (WAN) link is used to connect to CenturyLink to receive a full BGP table and to connect
 137 to the RIRs.

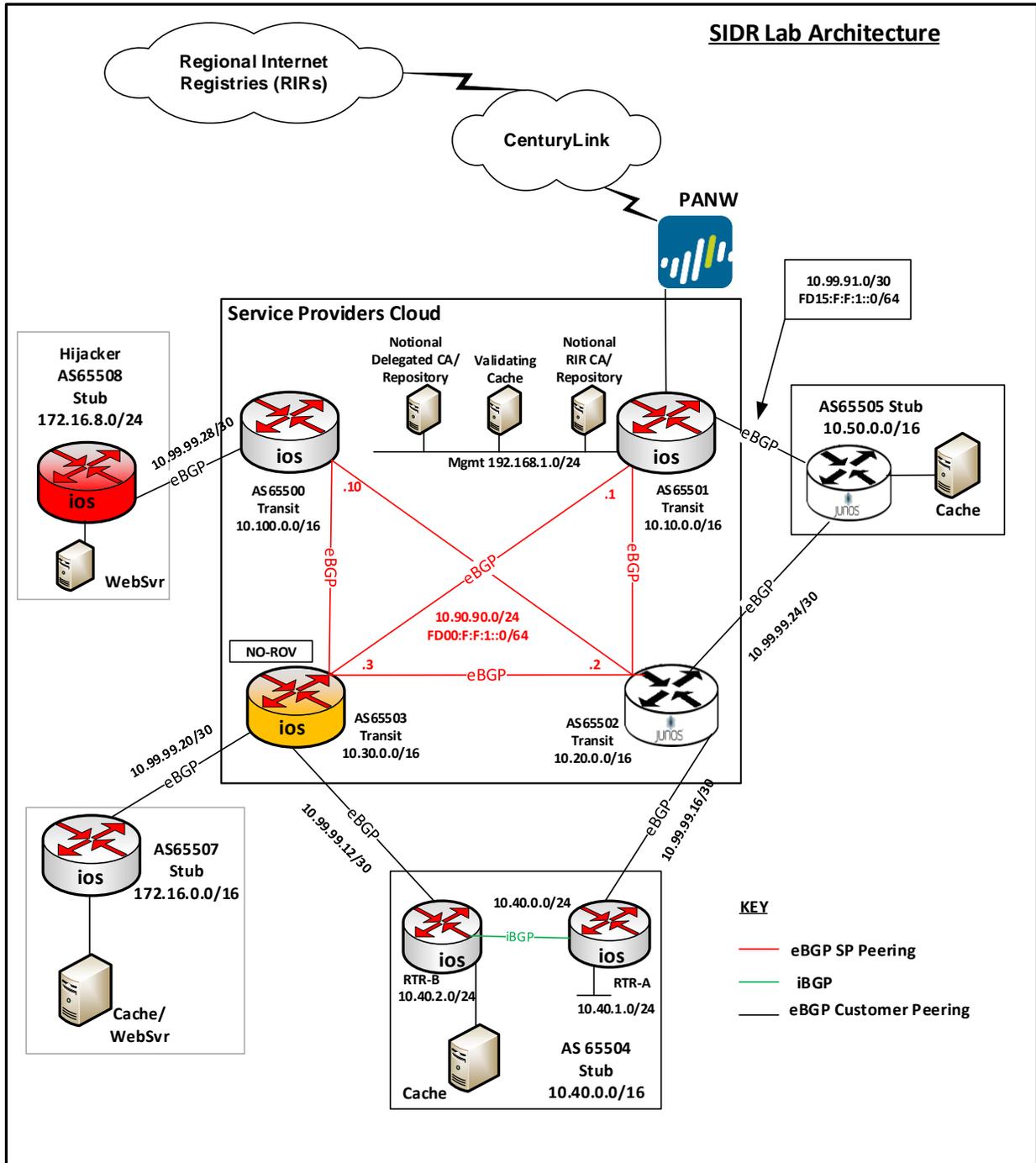
138 **Figure 1-1 Test Harness Environment for SIDR RPKI-Based ROV Solution Testing**



BGPSEC-IO (BIO) – BGP traffic generator & collector / RTR – CISCO or Juniper Router

139

140 Figure 1-2 Live Data Environment for SIDR RPKI-Based ROV Solution Testing



141

142 **1.3 Typographic Conventions**

143 The following table presents typographic conventions used in this volume.

Typeface/Symbol	Meaning	Example
<i>Italics</i>	filenames and pathnames references to documents that are not hyperlinks, new terms, and placeholders	For detailed definitions of terms, see the <i>CSRC.NIST.GOV Glossary</i> .
Bold	names of menus, options, command buttons, and fields	Choose File > Edit .
Monospace	command-line input, on- screen computer output, sample code examples, status codes	Mkdir
Monospace Bold	command-line user input contrasted with computer output	service sshd start
blue text	link to other parts of the document, a web URL, or an email address	All publications from NIST’s National Cybersecurity Center of Excellence are available at http://www.nccoe.nist.gov

144 **2 Product Installation Guides**

145 This section of the Practice Guide contains detailed instructions for installing and configuring all of the
146 products used to build an instance of the SIDR RPKI-based ROV example solution. The main components
147 of the lab build consist of ROV-enabled routers, RPKI repositories, RPKI validators / validating caches
148 (VCs), a live internet circuit, and firewalls.

149 2.1 RPKI Validators

150 The RPKI validator receives and validates ROAs from the RPKI repositories of the trust anchors and
151 delegated repositories. Currently, there are five trust anchors, all of which are managed by the RIRs:
152 AFRINIC, APNIC, ARIN, LACNIC, and the RIPE NCC. A subset of the data from ROAs, called validated ROA
153 payload (VRP), is then retrieved from the local RPKI validator by an RPKI-capable router to perform ROV
154 of BGP routes.

155 In this lab build, two RPKI validators (also referred to as VCs) are tested: the RIPE NCC RPKI validator and
156 the Dragon Research RPKI.net validator.

157 2.1.1 RIPE NCC RPKI Validator Configuration/Installation

158 The RIPE NCC RPKI validator is developed and maintained by RIPE NCC [RIPE Tools]. This validator tool is
159 free and open-source. The version used in the build is 2.24. It is available for download at
160 <https://www.ripe.net/manage-ips-and-asns/resource-management/certification/tools-and-resources>.

161 System requirements: a UNIX-like operating system (OS), Java 7 or 8, rsync, and 2 gigabytes (GB) of free
162 memory.

163 Lab setup: CentOS 7 minimal install, Java 8, rsync, one central processing unit (CPU), 6 GB memory, and
164 running on a virtual machine (VM) on VMware ESXi.

165 For release notes, installation information, and source code, please view [https://github.com/RIPE-](https://github.com/RIPE-NCC/rpki-validator/blob/master/rpki-validator-app/README.txt)
166 [NCC/rpki-validator/blob/master/rpki-validator-app/README.txt](https://github.com/RIPE-NCC/rpki-validator/blob/master/rpki-validator-app/README.txt).

- 167 1. Use the CentOS template to create the VM with the system requirements provided above.
 - 168 a. Put the VM in the proper VLAN.
- 169 2. Install Java (must be Oracle 8) and open firewall to allow rsync.
- 170 3. In the VM, create a folder under home called "RPKI".

171 a. `# mkdir RPKI`

172 b. `# cd RPKI`

- 173 4. Download and install the RIPE NCC RPKI validator software in the VM.

174 a. `# tar -xvf rpki-validator-app-2.24-dist.tar.gz`

- 175 5. Set `JAVA_HOME` (only if the application complains that it does not see the `JAVA_HOME` path).

176 a. `# cd /etc/environment`

177 i. `# nano environment`

- 178 ii. # `JAVA_HOME="/usr"`
- 179 b. Source it and check echo.
- 180 i. # `source /etc/environment`
- 181 ii. # `Echo $JAVA_HOME`
- 182 6. Reboot the server.
- 183 7. Start the RPKI cache.
- 184 a. # `./rpki-validator.sh start`
- 185 8. Using a web browser, connect to the validator software that you just installed, by typing
- 186 `http://ip-address:8080` into the browser search window, replacing “ip-address” with the internet
- 187 protocol (IP) address of the VM that you just created in step 1. (i.e., `http://192.168.1.124:8080`).
- 188 9. Once the validator is up, it receives data from the following RIR repositories: AFRINIC, APNIC,
- 189 LACNIC, and RIPE NCC.
- 190 a. To retrieve ROAs from the ARIN repository, download the Trust Anchor Locator (TAL) file
- 191 from <https://www.arin.net/resources/rpki/tal.html>.
- 192 b. Stop the validator.
- 193 i. # `./rpki-validator.sh stop`
- 194 c. Put the file in the *TAL* sub-directory.
- 195 d. Restart the validator.
- 196 i. # `./rpki-validator.sh start`

197 2.1.2 Dragon Research RPKI.net Validator Configuration/Installation

198 The Dragon Research Labs-developed RPKI.net toolkit contains both a VC and a certificate authority

199 (CA). This section discusses the VC only.

200 System requirements: Ubuntu 16.04 Xenial server, 32 GB of hard disk, 1 GB of random access memory

201 (RAM), and a minimum of one CPU.

202 Lab setup: Ubuntu 16.04 Xenial server, rsync, one CPU, 6 GB memory, and running on a VM on VMware

203 ESXi.

204 For release notes, installation information, and additional information, please view

205 <https://github.com/dragonresearch/rpki.net/blob/master/doc/quickstart/xenial-rp.md>.

```
206 # wget -q -O
207 /etc/apt/sources.list.d/rpki.list https://download.rpki.net/APTng/rpki.xenial.l
208 ist
```

209 You may get a message that says that there were errors (i.e., “the following signatures couldn’t be
210 verified because the public key is not available”). To fix this, use the following command, along with the
211 key that showed up on the error:

```
212 # apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 40976EAF437D05B5
```

213 Note: *40976EAF437D05B5* is an example. Use the exact key that showed up in the error.

214 Reference: [https://chrisjean.com/fix-apt-get-update-the-following-signatures-couldnt-be-verified-
215 because-the-public-key-is-not-available/](https://chrisjean.com/fix-apt-get-update-the-following-signatures-couldnt-be-verified-because-the-public-key-is-not-available/).

```
216 # apt update
```

```
217 # apt install rpki-rp
```

218 This should install the VC. Next, access the VC by opening a browser and typing
219 <http://192.168.2.106/rcynic> into the search window.

220 Note: It takes up to an hour to completely update. The proper Uniform Resource Locator (URL) will not
221 show up until then. Just wait for it. You will see a parent folder directory in the URL during that time.
222 Once it’s ready, charts about the repositories from the different RIRs will show up.

223 Check to see if the VC is running by entering the following command:

```
224 # ps -aux | grep rpki
```

225 2.2 RPKI CA and Repository

226 The delegated model of RPKI for ROA creation and storage requires that two components be set up,
227 operated, and maintained by the address holder: a CA and a repository. Currently, only the Dragon
228 Research RPKI.net toolkit provides the components needed to set up a delegated model.

229 2.2.1 Dragon Research RPKI.net CA and Repository Configuration/Installation

230 The setup for the CA and repository is different from the setup for the relying-party VC.

231 System requirements: Ubuntu 16.04 Xenial server, 32 GB of hard disk, 1 GB of RAM, and a minimum of
232 one CPU.

233 Lab setup: Ubuntu 16.04 Xenial server, rsync, one CPU, 6 GB memory, and running on a VM on VMware
234 ESXi.

235 For release notes, installation information, and additional information, please view
236 <https://github.com/dragonresearch/rpki.net/blob/master/doc/quickstart/xenial-ca.md>.

237 Steps for installing the rpki-ca (the CA software) toolkit for this lab build were different from the
238 instructions provided by the GitHub documentation. Guidance for the lab build is provided below.

239 *2.2.1.1 Assumptions*

240 Prior to installing rpki-ca and rpki-rp (the repository software), ensure that you are working with two
241 hosts running the Ubuntu Xenial server. In our setup, we will call one host *primary_root* (parent) and the
242 other host *remote_child* (child); both are running the Ubuntu Xenial server.

243 *2.2.1.2 Installation Instructions*

244 Run the initial setup to install rpki-ca. Follow the steps in the Xenial guide up to “CA Data initialization”.

245 Execute the steps under rcynic and rsyncd, specifically the “cat” commands that are listed.

246 *2.2.1.3 Getting rcynic to Run*

247 1. It’s important to note that the rcynic software will NOT be installed correctly. You will need to
248 add the following line to */var/spool/cron/crontabs/rcynic*:

249 `*/10 * * * * exec /usr/bin/rcynic-cron`

250 a. This ensures that the rcynic software will be run periodically to update the certificates.
251 This should be done on both hosts. Rcynic is designed to run periodically by default.

252 b. Rcynic will error out when external TAL files are called. Delete all repository files in the
253 trust-anchors folder. To do this, run the following command:

254 `# rm /etc/rpki/trust-anchors/*`

255 i. This should be done on both hosts.

256 2. The next step is to edit the */etc/rpki.conf* file.

257 a. On the host that we will be calling *primary_root*, make the following changes:

258 i. Change the handle to *primary_root*.

259 ii. Change *rpki_server_host* to *0.0.0.0*.

260 iii. Change *irdb_server_host* to *0.0.0.0*.

261 iv. Set *run_pubd* to *yes*.

262 v. Change *pubd_server_host* to *0.0.0.0*.

263 This should be sufficient for the changes on *primary_root*.

264 b. On the host that we will be calling *remote_child*, make the following changes to
265 */etc/rpki.conf*:

266 i. Change the handle to *remote_child*.

267 ii. Change *rpki_server_host* to *localhost*.

268 iii. Change *irdb_server_host* to *localhost*.

269 iv. Set *run_pubd* to *no*.

270 v. Change *pubd_server_host* to *primary_root*.

271 This last change means that *remote_child* will look to *primary_root* as the
272 publication server rather than running its own. To access *primary_root*,
273 *remote_child* will need a Domain Name System entry for *primary_root*.

274 1) To create this, first find *primary_root*'s IP address by running **ifconfig**
275 on *primary_root*. In our setup, this IP address is 192.168.2.115.

276 2) Then, on *remote_child*, we add the following line to the */etc/hosts* file:

```
277                    192.168.2.115: primary_root :(Replacing the IP address with  
278                    whatever IP address is currently assigned to primary_root.)
```

279 At this point, *rcynic*, *rpki*, and *rsyncd* should all be set up.

280 3. On both hosts, run the following commands to reboot the services:

```
281            # systemctl restart xinetd
```

```
282            # systemctl restart rpki-ca
```

283 2.2.1.4 GUI Setup

284 1. Set up the graphical user interface (GUI) on both VMs by running the following command:

```
285            # rpki-manage createsuperuser
```

286 2. Fill in the details appropriately. Verify that each GUI is up by opening a browser and visiting
287 <https://127.0.0.1> on both hosts.

288 2.2.1.5 Root CA Repository Setup

289 1. For simplicity, create a folder named */root/CA-stuff* on both VMs. Change the directory into this
290 folder for both VMs.

291 2. Now, we will set up *primary_root* as a root server for all resources.

292 a. On `primary_root`, run the following command:

293 `# rpki create_identity primary_root`

294 This will produce a file named `primary_root.identity.xml`.

295 b. Next, run the following command:

296 `# rpki configure_root`

297 This will produce a file named `primary_root.primary_root.repository-request.xml`. We
298 will return to this file later.

299 c. Now, run the following command:

300 `# rpki -i primary_root extract_root_certificate`

301 `# rpki -i primary_root extract_root_tal`

302 These commands will respectively produce a `.cer` file and a `.tal` file.

303 d. Copy both of these files into the `/usr/share/rpki/rrdp-publication` folder. (Note: This
304 step may not be necessary.)

305 e. Copy the `.tal` file to `/etc/rpki/trust-anchors`. This step configures `rcynic` to look at this
306 node as a repository.

307 f. Now, we will copy the `.tal` file from `primary_root` to `remote_child`. One way to do this is
308 with `rsync` as follows:

309 i. Copy the `.tal` file to `/usr/share/rpki/publication` on `primary_root`.

310 ii. On `remote_child`, run the following command to verify that `rsync` is working,
311 replacing the IP address as appropriate in the command below:

312 `# rsync rsync://192.168.2.115/rpki`

313 iii. If the above runs correctly, copy the `.tal` file, replacing `<file>` as appropriate in the
314 command below:

315 `# rsync rsync://192.168.2.115/rpki/<file>.tal /etc/rpki/trust-`
316 `anchors`

317 Now, `primary_root`'s `.tal` file should be on both VMs in the `/etc/rpki/trust-anchors`
318 directory.

319 g. We now want to update rcynic. To force it to synchronize, we run the following
320 command on both VMs:

```
321 # sudo -u rpki python /usr/bin/rcynic-cron
```

322 i. To verify that rcynic works, visit <https://127.0.0.1/rcynic> on both VMs.

323 h. We return to setting up `primary_root`.

324 i. On `primary_root`, find the file named `primary_root.primary_root.repository-`
325 `request.xml`. Once in the right directory, run the following command:

```
326 # rpki configure_publication_client  
327 primary_root.primary_root.repository-request.xml
```

328 This should produce a file named `primary_root.repository-response`.

329 ii. With this file, run the following command:

```
330 # rpki configure_repository primary_root.repository-response
```

331 Now, `primary_root` should be set up.

332 i. On `primary_root`, visit <https://127.0.0.1> and log in. You should see `primary_root` as a
333 repository at the bottom of the page.

334 2.2.1.6 Child CA Repository Setup

335 1. Our next step is to set up `remote_child` as a child of `primary_root`. On `remote_child`, run the
336 following command:

```
337 # rpki create_identity remote_child
```

338 This will produce a file named `remote_child.identity.xml`.

339 2. We now want to copy this over to `primary_root` by using `rsync`.

340 a. First, copy the file to `/usr/share/rpki/publication` on `remote_child`.

341 b. Next, on `primary_root`, run the following command:

```
342 # rsync rsync://192.168.2.116/rpki/remote_child.identity.xml ./
```

343 (Replace `192.168.2.116` with `remote_child`'s IP address in the command above.)

344 This command will copy the child's identity file to the current working directory on
345 `primary_root`.

346 c. Now, on `primary_root`, run the following command:

347 `# rpki configure_child remote_child.identity.xml`

348 This will produce a file named `primary_root.remote_child.parent-response.xml`.

349 3. We will copy this file over to `remote_child`.

350 a. To do this, first (on `primary_root`) copy the file to `/usr/share/rpki/publication`.

351 b. Next, on `remote_child`, run the following command:

352 `# rsync rsync://192.168.2.115/rpki/primary_root.remote_child.parent-`
353 `response.xml ./`

354 (Replace the IP address with the appropriate one for `primary_root` in the command
355 above.)

356 This command will copy the response to the current working directory on `remote_child`.

357 c. With this file, we now run the following command on `remote_child`:

358 `# rpki configure_parent primary_root.remote_child.parent-response.xml`

359 This will produce a file named `remote_child.primary_root.repository-request.xml`.

360 4. We will copy this file to `primary_root` with `rsync`.

361 a. To do this, on `remote_child`, copy the file to `/usr/share/rpki/publication`.

362 b. Then, on `primary_root`, run the following command:

363 `# rsync rsync://192.168.2.116/rpki/remote_child.primary_root.repository-`
364 `request.xml ./`

365 (Replace the IP address in the command above with `remote_child`'s IP address).

366 This will copy the file to the current working directory.

367 c. Now, on `primary_root`, we run the following command:

368 `# rpki configure_publication_client`
369 `remote_child.primary_root.repository-request.xml`

370 This will produce a file named `remote_child.repository-response.xml`.

371 5. We will copy this file to the `remote_child` by using `rsync`.

372 a. On `primary_root`, copy the file to `/usr/share/rpki/publication`.

373 b. Then, on remote_child, run the following command:

```
374 # rsync rsync://192.168.2.115/rpki/remote_child.repository-response.xml
375 ./
```

376 (Replace the IP address as necessary in the command above.)

377 This will copy the file to the current working directory.

378 c. Now, on remote_child, we run the following command:

```
379 # rpkic configure_repository remote_child.repository-response.xml
```

380 *2.2.1.7 Run rcynic to Update Root and Child CA Repositories*

381 This will complete the parent-child setup between primary_root and remote_child. Before verifying, we
382 run the following commands on both VMs:

```
383 # rpkic force_publication
384 # rpkic force_run_now
385 # rpkic synchronize
386 # sudo -u rpki python /usr/bin/rcynic-cron
```

387 This should force both VMs to fully update everything, including running rcynic. At this point, you should
388 verify that primary_root shows up as a parent on remote_child's GUI, and that remote_child shows up
389 as a child on primary_root's GUI. Now, we can assign resources. On primary_root's GUI, assign some
390 resources to remote_child. Given enough time, remote_child should update its GUI to reflect that it has
391 been assigned resources under the resources header on the GUI.

392 *2.2.1.8 Adding Resources*

393 When adding resources using the GUI, run the following commands to ensure that rcynic runs to update
394 the repository:

```
395 # rpkic force_run_now
396 # rpkic synchronize
397 # sudo -u rpki python /usr/bin/rcynic-cron
```

398 **2.3 BGP-SRx Software Suite**

399 BGP Secure Routing Extension (BGP-SRx) is an open-source reference implementation and research
400 platform for investigating emerging BGP security extensions and supporting protocols, such as RPKI
401 Origin Validation and Border Gateway Protocol Security (BGPsec) Path Validation [[NIST BGP-SRx](#)].

402 For the latest installation information, please use the Quick Install Guide:
403 <https://bgpsrx.antd.nist.gov/bgpsrx/documents/SRxSoftwareSuite-5.0-QuickInstallGuide.pdf>.

404 **2.4 Firewalls**

405 The firewall used for the lab build is the Palo Alto Next Generation Firewall. The firewall provides
406 protection against known and unknown threats. In this deployment, only ports and connections
407 necessary for the build are configured. All other ports and connections are denied.

408 System requirements: Palo Alto PA-5060 Next Generation Firewall running Version 7.1.10 software.

409 The configuration shown in [Figure 2-1](#) addressed all ports that are allowed by the firewall. Ports that are
410 allowed by the firewall are BGP, rsync, and RPKI Repository Delta Protocol (RRDP). All other ports are
411 denied by the firewall. [Figure 2-1](#) depicts the firewall rules.

412 Figure 2-1 Palo Alto Firewall Configuration

The screenshot displays the Palo Alto Networks firewall configuration interface, specifically the 'Policies' tab. The interface includes a top navigation bar with 'Dashboard', 'ACC', 'Monitor', 'Policies', 'Objects', 'Network', and 'Device'. A search bar and 'Commit', 'Save', and 'Search' buttons are also visible. The main content area shows a table of security rules with columns for Name, Tags, Type, Source (Zone, Address, User, HIP Profile), Destination (Zone, Address), Application, Service, Action, and Profile. The table contains 7 rules, with the last two being default rules for intrazone and interzone traffic.

	Name	Tags	Type	Source				Destination		Application	Service	Action	Profile
				Zone	Address	User	HIP Profile	Zone	Address				
1	BGP_PE_AND_CE	none	interzone	trust untrust	CE_ROUTER	any	any	untrust	PE_ROUTER	bgp	application-d...	Allow	none
2	ICMP-Untrust-Trust	none	universal	trust untrust	any	any	any	trust untrust	any	ping	application-d...	Allow	none
3	RPKI-In-Out	none	universal	trust untrust	any	any	any	trust untrust	CE_ROUTER PE_ROUTER	rsync	application-d...	Allow	none
4	Deny-SSH-Telnet	none	universal	any	any	any	any	any	any	ssh telnet	application-d...	Deny	none
5	RRDP-HTTPS	none	interzone	trust	any	any	any	any	any	any	service-https	Allow	none
6	intrazone-default	none	intrazone	any	any	any	any	(intrazone)	any	any	any	Allow	none
7	interzone-default	none	interzone	any	any	any	any	any	any	any	any	Deny	none

At the bottom of the interface, there is a toolbar with options: Add, Delete, Clone, Override, Revert, Enable, Disable, Move, and Highlight Unused Rules.

413

414 2.5 Test Harness Topology Configuration

415 The configurations provided in this section are the configurations that are used on each of the routers
416 when operating in the test harness environment architecture provided in [Figure 1-1](#) in [Section 1.2](#).
417 Initially, Cisco routers were used as routers RTR 1-1, RTR 2-1, and RTR 2-2 in that architecture to perform
418 the functional tests. The same tests were then repeated, replacing the Cisco routers with Juniper routers
419 as RTR 1-1, RTR 2-1, and RTR 2-2.

420 The systems and operating software used for the Cisco routers are as follows:

- 421 ▪ Cisco 7206 running *c7200p-adventerprisk9-mz.152-4.s7.bin*, with a minimum of 4-gigabit
422 Ethernet (GbE) ports. Routers AS 65500 (RTR 2-1) and AS 65501 (RTR 1-1) use this system and
423 OS.
- 424 ▪ Cisco 4331 running *ISR4300-universalk9.16.03.04.SPA.bin*, with a minimum of 4 GbE ports.
425 Router AS 65504A (RTR 2-2) uses this system and OS.

426 All Juniper routers have the following requirements: Juniper MX80 running on Juniper Operating System
427 (JUNOS) 15.1R6.7, with a minimum of 4 GbE ports. Routers AS 65500 (RTR 2-2), AS 65503-J (RTR 2-1),
428 and AS 65505 (RTR 1-1) use this system and OS.

429 The BGP-SRx Software Suite traffic generators can run on a CentOS Linux system with minimum
430 requirements.

431 2.5.1 RTR 1-1 Configuration – Cisco

432 RTR 1-1 acts as an exterior border gateway protocol (eBGP) router receiving eBGP routes from BIO-1, as
433 depicted in [Figure 1-1](#). It updates its interior border gateway protocol (iBGP) peer, BIO-2, with iBGP
434 updates. VRP data is provided to RTR 1-1 by the RPKI validator.

```
435     hostname AS65501
436     !
437     interface GigabitEthernet0/1
438         ip address 10.90.90.1 255.255.255.0
439         ipv6 address FD00:F:F:1::1/64
440     !
441     interface FastEthernet0/2
442         description VLAN1
443         ip address 192.168.1.2 255.255.255.0
```

```
444      !
445      interface GigabitEthernet0/2
446          ip address x.x.x.x 255.255.255.252 #Actual IP address to CenturyLink removed.
447      !
448      interface GigabitEthernet0/3
449          ip address y.y.y.y 255.255.255.248 #Actual IP address to CenturyLink removed.
450          ipv6 address FD15:F:F:1::1/64
451
452      !
453      router bgp 65501
454          bgp log-neighbor-changes
455          bgp rpki server tcp 192.168.1.52 port 8282 refresh 5
456          neighbor 10.90.90.4 remote-as 65501
457          neighbor 192.168.1.50 remote-as 65510
458          neighbor 192.168.1.51 remote-as 65511
459          neighbor 192.168.1.52 remote-as 65501
460          neighbor 192.168.1.53 remote-as 65512
461          neighbor FD00:F:F:1::3 remote-as 65503
462      !
463      address-family ipv4
464          bgp bestpath prefix-validate allow-invalid
465          no neighbor 10.90.90.4 activate
466          neighbor 192.168.1.50 activate
467          neighbor 192.168.1.51 activate
468          neighbor 192.168.1.52 activate
469          neighbor 192.168.1.52 send-community both
```

```
470     neighbor 192.168.1.52 announce rpki state
471     neighbor 192.168.1.53 activate
472     no neighbor FD00:F:F:1::3 activate
473     exit-address-family
474     !
475     address-family ipv6
476         redistribute connected
477         neighbor FD00:F:F:1::3 activate
478     exit-address-family
479     !
480     ip prefix-list WAN-OUT seq 10 permit 65.118.221.8/29
481     !
482     route-map rpki permit 10
483         match rpki invalid
484         set local-preference 100
485     !
486     route-map RPKI-TEST permit 10
487         match ip address prefix-list WAN-OUT
488         set community 13698023
489     !
490     end
```

491 2.5.2 RTR 2-1 Configuration – Cisco

492 RTR 2-1 acts as an eBGP router receiving eBGP routes from BIO-0, and as an iBGP peer providing updates
493 to RTR 2-2, as depicted in [Figure 1-1](#). RTR 2-1 updates another iBGP peer, BIO-2, with iBGP updates. VRP
494 data is provided to RTR 1-1 by the RPKI validator.

```
495     hostname AS65500
496     !
497     interface Loopback1
498         ip address 10.100.0.1 255.255.0.0
499         ipv6 address 2010:10:10:10::1/64
500     !
501     interface GigabitEthernet0/1
502         ip address 10.90.90.10 255.255.255.0
503         ipv6 address FD00:F:F:1::10/64
504     !
505     interface FastEthernet0/2
506         ip address 192.168.1.4 255.255.255.0
507     !
508     interface GigabitEthernet0/2
509         ip address 10.99.99.21 255.255.255.252
510     !
511     interface GigabitEthernet0/3
512         description VLAN8
513     !
514     router bgp 65500
515         bgp log-neighbor-changes
516         bgp rpki server tcp 192.168.1.52 port 8282 refresh 5
```

```
517      bgp rpki server tcp 192.168.1.53 port 8282 refresh 5
518      neighbor 192.168.1.5 remote-as 65500
519      neighbor 192.168.1.50 remote-as 65510
520      neighbor 192.168.1.51 remote-as 65511
521      neighbor 192.168.1.52 remote-as 65500
522      neighbor 192.168.1.53 remote-as 65513
523      !
524      address-family ipv4
525          bgp bestpath prefix-validate allow-invalid
526          redistribute connected
527          neighbor 192.168.1.5 activate
528          neighbor 192.168.1.5 send-community both
529          neighbor 192.168.1.5 announce rpki state
530          neighbor 192.168.1.50 activate
531          neighbor 192.168.1.51 activate
532          neighbor 192.168.1.52 activate
533          neighbor 192.168.1.52 send-community both
534          neighbor 192.168.1.52 announce rpki state
535          neighbor 192.168.1.53 activate
536      exit-address-family
537      !
538      route-map 10 permit 10
539      !
540      end
```

541 2.5.3 RTR 2-2 Configuration – Cisco

542 RTR 2-2 acts as an iBGP router receiving iBGP routes from RTR 2-1, and as an eBGP peer providing
543 updates to BIO-6, as depicted in [Figure 1-1](#).

```
544     version 16.3
545     !
546     hostname AS65504A
547     !
548     interface GigabitEthernet0/0/0
549         description VLNA5
550         ip address 10.40.0.1 255.255.255.0
551         ipv6 address FD34:F:F:1::4/64
552     !
553     interface GigabitEthernet0/0/1
554         description VLN6
555         ip address 10.99.99.18 255.255.255.252
556         ipv6 address FD24:F:F:1::4/64
557     !
558     interface GigabitEthernet0/0/2
559         ip address 192.168.1.5 255.255.255.0
560         ipv6 address 2004:4444:4444:4444::4/64
561     !
562     router bgp 65500
563         bgp log-neighbor-changes
564         bgp rpki server tcp 192.168.1.53 port 8282 refresh 5
565         bgp rpki server tcp 192.168.1.52 port 8282 refresh 5
566         neighbor 192.168.1.4 remote-as 65500
```

```

567     neighbor 192.168.1.53 remote-as 65513
568     !
569     address-family ipv4
570         neighbor 192.168.1.4 activate
571         neighbor 192.168.1.4 send-community both
572         neighbor 192.168.1.4 announce rpki state
573         neighbor 192.168.1.53 activate
574     exit-address-family
575     !
576     route-map NO-EXPORT permit 10
577         set community no-export
578     !
579     end

```

580 2.5.4 RTR 1-1 Configuration – Juniper

581 RTR 1-1 acts as an eBGP router receiving eBGP routes from BIO-1, as depicted in [Figure 1-1](#). RTR 1-1
582 updates its iBGP peer, BIO-2, with iBGP updates. VRP data is provided to it by the RPKI validator.

```

583     set system host-name AS65501
584     set system login user nccoe uid 2000
585     set system login user nccoe class read-only
586     set system login user nccoe authentication encrypted-password
587     "$5$8.Yu28ng$LbcoMQ9uqDO3.U4VaiG4bg5fWMeaMYAJjr09Aniu8c7"
588     set interfaces ge-1/3/0 unit 0 family inet address 192.168.1.12/24
589     set interfaces ge-1/3/1 unit 0 family inet
590     set interfaces ge-1/3/2 unit 0 family inet
591     set interfaces ge-1/3/3 unit 0 family inet
592     set interfaces lo0 unit 0 family inet address 127.0.0.1/32
593     set routing-options autonomous-system 65501

```

```
594      set routing-options validation group cache session 192.168.1.52 refresh-time 5
595      set routing-options validation group cache session 192.168.1.52 port 8282
596      set protocols bgp group external-as65511 type external
597      set protocols bgp group external-as65511 import validation
598      set protocols bgp group external-as65511 export allow-direct
599      set protocols bgp group external-as65511 peer-as 65511
600      set protocols bgp group external-as65511 neighbor 192.168.1.51
601      set protocols bgp group external-as65510 type external
602      set protocols bgp group external-as65510 import validation
603      set protocols bgp group external-as65510 export allow-direct
604      set protocols bgp group external-as65510 peer-as 65510
605      set protocols bgp group external-as65510 neighbor 192.168.1.50
606      set protocols bgp group internal-as65501 type internal
607      set protocols bgp group internal-as65501 neighbor 192.168.1.52
608      set protocols bgp group external-as65512 type external
609      set protocols bgp group external-as65512 import validation
610      set protocols bgp group external-as65512 export allow-direct
611      set protocols bgp group external-as65512 peer-as 65512
612      set protocols bgp group external-as65512 neighbor 192.168.1.53
613      set policy-options policy-statement allow-all from route-filter 0.0.0.0/0
614      orlonger
615      set policy-options policy-statement allow-all then accept
616      set policy-options policy-statement allow-direct term default from protocol
617      direct
618      set policy-options policy-statement allow-direct term default then accept
619      set policy-options policy-statement validation term valid from protocol bgp
620      set policy-options policy-statement validation term valid from validation-
621      database valid
```

622 set policy-options policy-statement validation term valid then local-preference
623 110

624 set policy-options policy-statement validation term valid then validation-state
625 valid

626 set policy-options policy-statement validation term valid then community add
627 origin-validation-state-valid

628 set policy-options policy-statement validation term valid then accept

629 set policy-options policy-statement validation term invalid from protocol bgp

630 set policy-options policy-statement validation term invalid from validation-
631 database invalid

632 set policy-options policy-statement validation term invalid then local-
633 preference 90

634 set policy-options policy-statement validation term invalid then validation-
635 state invalid

636 set policy-options policy-statement validation term invalid then community add
637 origin-validation-state-invalid

638 set policy-options policy-statement validation term invalid then accept

639 set policy-options policy-statement validation term unknown from protocol bgp

640 set policy-options policy-statement validation term unknown then validation-
641 state unknown

642 set policy-options policy-statement validation term unknown then community add
643 origin-validation-state-unknown

644 set policy-options policy-statement validation term unknown then accept

645 set policy-options community origin-validation-state-invalid members 0x4300:2

646 set policy-options community origin-validation-state-unknown members 0x4300:1

647 set policy-options community origin-validation-state-valid members 0x4300:0

648 2.5.5 RTR 2-1 Configuration – Juniper

649 RTR 2-1 acts as an eBGP router receiving eBGP routes from BIO-0, and as an iBGP peer providing updates
650 to RTR 2-2, as depicted in [Figure 1-1](#). It updates another iBGP peer, BIO-2, with iBGP updates. VRRP data
651 is provided to RTR 2-1 by the RPKI validator.

```
652     set system host-name AS65500-J
653     set interfaces ge-1/3/0 unit 0 family inet
654     set interfaces ge-1/3/1 unit 0 family inet address 192.168.1.14/24
655     set interfaces lo0 unit 0 family inet address 127.0.0.1/32
656     set routing-options autonomous-system 65500
657     set routing-options validation traceoptions file rpki-trace
658     set routing-options validation traceoptions flag all
659     deactivate routing-options validation traceoptions
660     set routing-options validation group cache session 192.168.1.52 refresh-time 5
661     set routing-options validation group cache session 192.168.1.52 port 8282
662     set protocols bgp group external-as65511 type external
663     set protocols bgp group external-as65511 import validation
664     set protocols bgp group external-as65511 export allow-direct
665     set protocols bgp group external-as65511 peer-as 65511
666     set protocols bgp group external-as65511 neighbor 192.168.1.51
667     set protocols bgp group external-as65510 type external
668     set protocols bgp group external-as65510 import validation
669     set protocols bgp group external-as65510 export allow-direct
670     set protocols bgp group external-as65510 peer-as 65510
671     set protocols bgp group external-as65510 neighbor 192.168.1.50
672     set protocols bgp group internal-as65500 type internal
673     set protocols bgp group internal-as65500 neighbor 192.168.1.52
```

674 set policy-options policy-statement allow-all from route-filter 0.0.0.0/0
675 orlonger

676 set policy-options policy-statement allow-all then accept

677 set policy-options policy-statement allow-direct term default from protocol
678 direct

679 set policy-options policy-statement allow-direct term default then accept

680 set policy-options policy-statement validation term valid from protocol bgp

681 set policy-options policy-statement validation term valid from validation-
682 database valid

683 set policy-options policy-statement validation term valid then local-preference
684 110

685 set policy-options policy-statement validation term valid then validation-state
686 valid

687 set policy-options policy-statement validation term valid then community add
688 origin-validation-state-valid

689 set policy-options policy-statement validation term valid then accept

690 set policy-options policy-statement validation term invalid from protocol bgp

691 set policy-options policy-statement validation term invalid from validation-
692 database invalid

693 set policy-options policy-statement validation term invalid then local-
694 preference 90

695 set policy-options policy-statement validation term invalid then validation-
696 state invalid

697 set policy-options policy-statement validation term invalid then community add
698 origin-validation-state-invalid

699 set policy-options policy-statement validation term invalid then accept

700 set policy-options policy-statement validation term unknown from protocol bgp

701 set policy-options policy-statement validation term unknown then validation-
702 state unknown

703 set policy-options policy-statement validation term unknown then community add
704 origin-validation-state-unknown

705 set policy-options policy-statement validation term unknown then accept

```
706     set policy-options community origin-validation-state-invalid members 0x4300:0:2
707     set policy-options community origin-validation-state-unknown members 0x4300:0:1
708     set policy-options community origin-validation-state-valid members 0x4300:0:0
```

709 2.5.6 RTR 2-2 Configuration – Juniper

710 RTR 2-2 acts as an iBGP router receiving iBGP routes from RTR 2-1, and as an eBGP peer providing
711 updates to BIO-6, as depicted in [Figure 1-1](#).

```
712     set system host-name AS65500
713     set interfaces ge-1/3/0 unit 0 family inet address 192.168.1.15/24
714     set interfaces ge-1/3/1 unit 0
715     set interfaces ge-1/3/2 unit 0
716     set interfaces ge-1/3/3 unit 0
717     set interfaces lo0 unit 0 family inet
718     set routing-options autonomous-system 65500
719     set routing-options validation group cache session 192.168.1.52 refresh-time 5
720     set routing-options validation group cache session 192.168.1.52 port 8282
721     set routing-options validation group cache session 192.168.1.53 refresh-time 5
722     set routing-options validation group cache session 192.168.1.53 port 8282
723     set protocols bgp group internal-as65500 type internal
724     set protocols bgp group internal-as65500 neighbor 192.168.1.14
725     set protocols bgp group external-as65513 type external
726     set protocols bgp group external-as65513 import validation
727     set protocols bgp group external-as65513 export allow-direct
728     set protocols bgp group external-as65513 peer-as 65513
729     set protocols bgp group external-as65513 neighbor 192.168.1.53
730     set policy-options policy-statement allow-all from route-filter 0.0.0.0/0
731     orlonger
732     set policy-options policy-statement allow-all then accept
```

733 set policy-options policy-statement allow-direct term default from protocol
734 direct

735 set policy-options policy-statement allow-direct term default then accept

736 set policy-options policy-statement validation term valid from protocol bgp

737 set policy-options policy-statement validation term valid from validation-
738 database valid

739 set policy-options policy-statement validation term valid then local-preference
740 110

741 set policy-options policy-statement validation term valid then validation-state
742 valid

743 set policy-options policy-statement validation term valid then community add
744 origin-validation-state-valid

745 set policy-options policy-statement validation term valid then accept

746 set policy-options policy-statement validation term invalid from protocol bgp

747 set policy-options policy-statement validation term invalid from validation-
748 database invalid

749 set policy-options policy-statement validation term invalid then local-
750 preference 90

751 set policy-options policy-statement validation term invalid then validation-
752 state invalid

753 set policy-options policy-statement validation term invalid then community add
754 origin-validation-state-invalid

755 set policy-options policy-statement validation term invalid then accept

756 set policy-options policy-statement validation term unknown from protocol bgp

757 set policy-options policy-statement validation term unknown then validation-
758 state unknown

759 set policy-options policy-statement validation term unknown then community add
760 origin-validation-state-unknown

761 set policy-options policy-statement validation term unknown then accept

762 set policy-options community origin-validation-state-invalid members 0x4300:2

763 set policy-options community origin-validation-state-invalid members 0x43:100:2

764 set policy-options community origin-validation-state-unknown members 0x4300:1

```
765     set policy-options community origin-validation-state-valid members 0x4300:0
```

766 2.5.7 Traffic Generator BIO Configuration

```
767     ski_file      = "/var/lib/key-volt/ski-list.txt";
768     ski_key_loc  = "/var/lib/key-volt/";
769     preload_eckey = false;
770     mode         = "BGP";
771     max          = 0;
772     only_extended_length = true;
773     session = (
774     {
775         disconnect = 0;
776         ext_msg_cap      = true;
777         ext_msg_liberal = true;
778         bgpsec_v4_snd   = false;
779         bgpsec_v4_rcv   = false;
780         bgpsec_v6_snd   = false;
781         bgpsec_v6_rcv   = false;    update = (
782             );
783         incl_global_updates = true;
784         algo_id = 1;
785         signature_generation = "BIO";
786         null_signature_mode = "FAKE";
787         fake_signature      = "1BADBEEFDEADFEED" "2BADBEEFDEADFEED"
788                               "3BADBEEFDEADFEED" "4BADBEEFDEADFEED"
789                               "5BADBEEFDEADFEED" "6BADBEEFDEADFEED"
790                               "7BADBEEFDEADFEED" "8BADBEEFDEADFEED"
791                               "ABADBEEFFACE";
792         fake_ski            = "0102030405060708" "090A0B0C0D0E0F10"
793                               "11121314";
794         printOnSend = {
```

```
795         update          = true;
796     };
797
798     printOnReceive = {
799         update.          = true;
800         notification = true;
801         unknown       = true;
802     };
803     printSimple      = true;
804     printPollLoop   = false;
805     printOnInvalid  = false;
806 }
807 );
808 update = (
809     );
```

810 *2.5.7.1 AS – Peer Configuration: BIO-0 (AS 65510) – RTR-1-1 (AS 65501)*

```
811     asn                = 65510;
812     bgp_ident         = "192.168.1.50";
813     hold_timer        = 180;
814
815     peer_asn          = 65501;
816     # For CISCO replace x with 2, For JUNIPER replace x with 12
817     peer_ip           = "192.168.1.x";
818     peer_port         = 179;
```

819 *2.5.7.2 AS – Peer Configuration: BIO-0 (AS 65510) – RTR-2-1 (AS 65500)*

```
820     asn                = 65510;
821     bgp_ident         = "192.168.1.50";
822     hold_timer        = 180;
823
824     peer_asn          = 65500;
```

```
825         # For CISCO replace x with 4, For JUNIPER replace x with 14
826         peer_ip      = "192.168.1.x";
827         peer_port    = 179;
```

828 *2.5.7.3 AS – Peer Configuration: BIO-1 (AS 65511) – RTR-1-1 (AS 65501)*

```
829         asn          = 65511;
830         bgp_ident    = "192.168.1.51";
831         hold_timer   = 180;
832
833         peer_asn     = 65500;
834         # For CISCO replace x with 2, For JUNIPER replace x with 12
835         peer_ip      = "192.168.1.x";
836         peer_port    = 179;
```

837 *2.5.7.4 AS – Peer Configuration: BIO-1 (AS 65511) – RTR-2-1 (AS 65500)*

```
838         asn          = 65511;
839         bgp_ident    = "192.168.1.51";
840         hold_timer   = 180;
841
842         peer_asn     = 65500;
843         # For CISCO replace x with 4, For JUNIPER replace x with 14
844         peer_ip      = "192.168.1.x";
845         peer_port    = 179;
```

846 *2.5.7.5 AS – Peer Configuration: BIO-2 (AS 65501) – RTR-1-1 (AS 65501)*

```
847         asn          = 65501;
848         bgp_ident    = "192.168.1.52";
849         hold_timer   = 180;
850
851         peer_asn     = 65501;
852         # For CISCO replace x with 2, For JUNIPER replace x with 12
853         peer_ip      = "192.168.1.x";
854         peer_port    = 179;
```

855 **2.5.7.6 AS – Peer Configuration: BIO-3 (AS 65500) – RTR-2-1 (AS 65500)**

```
856     asn          = 65500;
857     bgp_ident   = "192.168.1.52";
858     hold_timer  = 180;
859
860     peer_asn    = 65500;
861     # For CISCO replace x with 4, For JUNIPER replace x with 14
862     peer_ip     = "192.168.1.x";
863     peer_port   = 179;
```

864 **2.5.7.7 AS – Peer Configuration: BIO-5 (AS 65512) – RTR-1-1 (AS 65500)**

```
865     asn          = 65512;
866     bgp_ident   = "192.168.1.53";
867     hold_timer  = 180;
868
869     peer_asn    = 65501;
870     # For CISCO replace x with 2, For JUNIPER replace x with 12
871     peer_ip     = "192.168.1.x";
872     peer_port   = 179;
```

873 **2.5.7.8 AS – Peer Configuration: BIO-6 (AS 65513) – RTR-1-1 (AS 65513)**

```
874     asn          = 65513;
875     bgp_ident   = "192.168.1.53";
876     hold_timer  = 180;
877
878     peer_asn    = 65500;
879     # For CISCO replace x with 4, For JUNIPER replace x with 14
880     peer_ip     = "192.168.1.x";
881     peer_port   = 179;
```

882 2.6 Live Data Configuration

883 The configurations provided in this section are the configurations that are used on each of the routers
884 when operating in the live data environment architecture shown in [Figure 1-2](#). Live BGP data and RPKI
885 data can be retrieved in this environment. The architecture is organized into eight separate networks,
886 each of which is designed to represent a different AS.

887 The systems and operating software used for the Cisco routers are as follows:

- 888 ▪ Cisco 7206 running *c7200p-adventerprsrk9-mz.152-4.s7.bin*, with a minimum of 4 GbE ports.
889 Routers AS 65500, AS 65501, and AS 65503 use this system and OS.
- 890 ▪ Cisco 4331 running *ISR4300-universalk9.16.03.04.SPA.bin*, with a minimum of 4 GbE ports.
891 Routers AS 65504A and AS 65504B use this system and OS.
- 892 ▪ Cisco 2921 running *c2900-universalk9-mz-SPA.152-4.M6.bin*, with a minimum of 4 GbE ports.
893 Routers AS 65507 and AS 65508 use this system and OS.
- 894 ▪ Cisco Internetwork Operating System (IOS) XRv 9000 router Version 6.4.1 running on VMware
895 ESXi using the *xrv9k-fullk9-x.vrr-6.4.1.ova* file.

896 All Juniper routers have the following requirements: Juniper MX80 running on JUNOS 15.1R6.7, with a
897 minimum of 4 GbE ports. Routers AS 65502 and AS 65505 use this system and OS.

898 RPKI validators and repositories are configured based on [Section 2.1](#) and [Section 2.2](#). Live ROV data is
899 retrieved from the five trust anchors, and lab ROA data is retrieved from the lab delegated model of the
900 local RPKI repository.

901 Note: Real IP addresses and AS numbers were removed from the configuration.

902 2.6.1 CenturyLink Configuration Router AS 65501 – Cisco

903 To receive a full BGP route table, CenturyLink provided a physical link connecting the NCCoE lab with an
904 eBGP peering. The configuration below illustrates the eBGP peering. An additional configuration for this
905 router, related to the lab build, is provided in [Section 2.5.3](#).

```
906     version 15.2
907     !
908     hostname AS65501
909     !
910     ipv6 unicast-routing
911     ipv6 cef
```

```
912      !
913      interface GigabitEthernet0/1
914          ip address 10.90.90.1 255.255.255.0
915      ipv6 address FD00:F:F:1::1/64
916      !
917      interface FastEthernet0/2
918          description VLAN1
919          ip address 192.168.1.2 255.255.255.0
920      !
921      interface GigabitEthernet0/2
922          ip address a.a.a.a 255.255.255.252
923      !
924      interface GigabitEthernet0/3
925          ip address c.c.c.c 255.255.255.248
926
927      ipv6 address FD15:F:F:1::1/64
928      !
929      router bgp aaa
930          bgp log-neighbor-changes
931          neighbor a.a.a.b remote-as bbb
932      !
933      address-family ipv4
934          network c.c.c.d mask 255.255.255.248
935          neighbor a.a.a.b activate
936          neighbor a.a.a.b send-community
937          neighbor a.a.a.b soft-reconfiguration inbound
```

```
938     neighbor a.a.a.b route-map RPKI-TEST out
939     exit-address-family
940     !
941     ip prefix-list WAN-OUT seq 10 permit c.c.c.d/29
942     ipv6 router rip procl
943     !
944     route-map rpki permit 10
945     match rpki invalid
946     set local-preference 100
947     !
948     route-map RPKI-TEST permit 10
949     match ip address prefix-list WAN-OUT
950     set community 13698023
951     !
952     end
```

953 2.6.2 Router AS 65500 Configuration – Cisco

954 Router AS 65500 represents an ISP. For the lab build, this router originates BGP updates from its own AS
955 and receives and sends routes to and from its eBGP peers.

```
956     hostname AS65500
957     !
958     ip cef
959     ipv6 unicast-routing
960     ipv6 cef
961     !
962     interface Loopback1
963     ip address 10.10.0.1 255.255.0.0
```

```
964     ipv6 address FD10:10:10:10::1/64
965     ipv6 rip procl enable
966     !
967     interface GigabitEthernet0/1
968     ipv6 address FD00:F:F:1::1/64
969     ipv6 rip procl enable
970     !
971     interface FastEthernet0/2
972     description VLAN1
973     ip address 192.168.1.2 255.255.255.0
974     ipv6 address FD01:F:F:1::2/64
975     ipv6 rip procl enable
976     !
977     interface GigabitEthernet0/2
978     ip address a.a.a.a 255.255.255.252
979     !
980     interface GigabitEthernet0/3
981     ip address c.c.c.c 255.255.255.248
982     ipv6 address FD15:F:F:1::1/64
983     !
984     router rip
985     version 2
986     network 10.0.0.0
987     network 192.168.1.0
988     no auto-summary
989     !
```

```
990      router bgp aaa
991      bgp log-neighbor-changes
992      neighbor a.a.a.b remote-as bbb
993      !
994      address-family ipv4
995          network c.c.c.d mask 255.255.255.248
996      neighbor a.a.a.b activate
997      neighbor a.a.a.b send-community
998      neighbor a.a.a.b soft-reconfiguration inbound
999      neighbor a.a.a.b route-map RPKI-TEST out
1000     exit-address-family
1001     !
1002     ip route 10.20.0.0 255.255.0.0 192.168.1.3
1003     ip route 10.30.0.0 255.255.0.0 192.168.1.3
1004     ip route 10.40.0.0 255.255.0.0 192.168.1.3
1005     ip route 10.50.0.0 255.255.0.0 192.168.1.3
1006     ip route 10.70.0.0 255.255.0.0 192.168.1.3
1007     ip route 10.80.0.0 255.255.0.0 192.168.1.3
1008     ip route 10.90.90.0 255.255.255.0 192.168.1.3
1009     ip route 10.97.74.0 255.255.255.0 192.178.1.1
1010     ip route 10.99.99.0 255.255.255.0 192.168.1.3
1011     !
1012     ip prefix-list WAN-OUT seq 10 permit c.c.c.d /29
1013     ipv6 router rip procl
1014     !
1015     route-map rpki permit 10
```

```
1016     match rpki invalid
1017     set local-preference 100
1018     !
1019     route-map RPKI-TEST permit 10
1020     match ip address prefix-list WAN-OUT
1021     set community 13698023
1022     !
1023     end
```

1024 2.6.3 Router 65501 Configuration – Cisco

1025 Router AS 65501 represents an ISP. As indicated in [Section 2.5.1](#), this router peers with the CenturyLink
1026 router to receive a full BGP routing table. For the lab build, this router originates BGP updates from its
1027 own AS and receives and sends routes to and from its eBGP peers. It is the gateway for all devices in the
1028 lab, allowing ROAs from RIRs to be retrieved by RPKI validators. It also peers with stub AS A65505.

```
1029     hostname AS65501
1030     !
1031     ip cef
1032     ipv6 unicast-routing
1033     ipv6 cef
1034     !
1035     interface Loopback1
1036     ip address 10.10.0.1 255.255.0.0
1037     ipv6 address FD10:10:10:10::1/64
1038     ipv6 rip procl enable
1039     !
1040     interface GigabitEthernet0/1
1041     ipv6 address FD00:F:F:1::1/64
1042     ipv6 rip procl enable
```

```
1043      !
1044      interface FastEthernet0/2
1045          ip address 192.168.1.2 255.255.255.0
1046          ipv6 address FD01:F:F:1::2/64
1047          ipv6 rip procl enable
1048      !
1049      interface GigabitEthernet0/2
1050          ip address a.a.a.a 255.255.255.252
1051      !
1052      interface GigabitEthernet0/3
1053          ip address c.c.c.c 255.255.255.248
1054          ipv6 address FD15:F:F:1::1/64
1055      !
1056      router rip
1057          version 2
1058          network 10.0.0.0
1059          network 192.168.1.0
1060          no auto-summary
1061      !
1062      router bgp aaa
1063          bgp log-neighbor-changes
1064          neighbor a.a.a.b remote-as bbb
1065      !
1066          address-family ipv4
1067              network c.c.c.d mask 255.255.255.248
1068              neighbor a.a.a.b activate
```

```
1069     neighbor a.a.a.b send-community
1070     neighbor a.a.a.b soft-reconfiguration inbound
1071     neighbor a.a.a.b route-map RPKI-TEST out
1072     exit-address-family
1073     !
1074     ip route 10.20.0.0 255.255.0.0 192.168.1.3
1075     ip route 10.30.0.0 255.255.0.0 192.168.1.3
1076     ip route 10.40.0.0 255.255.0.0 192.168.1.3
1077     ip route 10.50.0.0 255.255.0.0 192.168.1.3
1078     ip route 10.70.0.0 255.255.0.0 192.168.1.3
1079     ip route 10.80.0.0 255.255.0.0 192.168.1.3
1080     ip route 10.90.90.0 255.255.255.0 192.168.1.3
1081     ip route 10.97.74.0 255.255.255.0 192.178.1.1
1082     ip route 10.99.99.0 255.255.255.0 192.168.1.3
1083     !
1084     ip prefix-list WAN-OUT seq 10 permit c.c.c.d /29
1085     ipv6 router rip procl
1086     !
1087     route-map rpki permit 10
1088     match rpki invalid
1089     set local-preference 100
1090     !
1091     route-map RPKI-TEST permit 10
1092     match ip address prefix-list WAN-OUT
1093     set community 13698023
1094     !
```

1095 end

1096 2.6.4 Router AS 65502 Configuration – Juniper

1097 Router AS 65502 represents an ISP using a Juniper router. For the lab build, this router originates BGP
1098 updates from its own AS and receives and sends routes to and from its eBGP peers. It also provides
1099 eBGP routes to stub AS 65504.

```
1100           set system host-name AS65502
1101           set interfaces ge-1/3/0 unit 0 family inet address 10.90.90.2/24
1102           set interfaces ge-1/3/0 unit 0 family inet6 address fd00:f:f:1::2/64
1103           set interfaces ge-1/3/1 unit 0 family inet address 10.99.99.17/30
1104           set interfaces ge-1/3/1 unit 0 family inet6 address fd24:f:f:1::2/64
1105           set interfaces ge-1/3/2 unit 0 family inet address 10.99.99.25/30
1106           set interfaces ge-1/3/2 unit 0 family inet6 address fd25:f:f:1::2/64
1107           set interfaces ge-1/3/3 unit 0 family inet address 10.20.0.1/16
1108           set interfaces ge-1/3/3 unit 0 family inet6 address 2020:2020:2020:1::2/64
1109           set interfaces lo0 unit 0 family inet address 127.0.0.1/32
1110           set routing-options validation group cache session 192.168.1.146 port 8282
1111           set policy-options policy-statement allow-all from route-filter 0.0.0.0/0
1112           orlonger
1113           set policy-options policy-statement allow-all then accept
1114           set routing-instances rpki instance-type virtual-router
1115           set routing-instances rpki interface ge-1/3/0.0
1116           set routing-instances rpki interface ge-1/3/1.0
1117           set routing-instances rpki interface ge-1/3/2.0
1118           set routing-instances rpki interface ge-1/3/3.0
1119           set routing-instances rpki interface lo0.1
1120           set routing-instances rpki routing-options router-id 2.2.2.2
1121           set routing-instances rpki routing-options autonomous-system 65502
```

1122 set routing-instances rpki protocols bgp group external-as65500 type external
1123 set routing-instances rpki protocols bgp group external-as65500 import allow-
1124 all
1125 set routing-instances rpki protocols bgp group external-as65500 export allow-
1126 all
1127 set routing-instances rpki protocols bgp group external-as65500 peer-as 65500
1128 set routing-instances rpki protocols bgp group external-as65500 neighbor
1129 10.90.90.10
1130 set routing-instances rpki protocols bgp group external-as65500 neighbor
1131 fd00:f:f:1::10
1132 set routing-instances rpki protocols bgp group external-as65501 type external
1133 set routing-instances rpki protocols bgp group external-as65501 import allow-
1134 all
1135 set routing-instances rpki protocols bgp group external-as65501 export allow-
1136 all
1137 set routing-instances rpki protocols bgp group external-as65501 peer-as 65501
1138 set routing-instances rpki protocols bgp group external-as65501 neighbor
1139 10.90.90.1
1140 set routing-instances rpki protocols bgp group external-as65501 neighbor
1141 fd00:f:f:1::1
1142 set routing-instances rpki protocols bgp group external-as65503 type external
1143 set routing-instances rpki protocols bgp group external-as65503 import allow-
1144 all
1145 set routing-instances rpki protocols bgp group external-as65503 export allow-
1146 all
1147 set routing-instances rpki protocols bgp group external-as65503 peer-as 65503
1148 set routing-instances rpki protocols bgp group external-as65503 neighbor
1149 10.90.90.3
1150 set routing-instances rpki protocols bgp group external-as65503 neighbor
1151 fd00:f:f:1::3
1152 set routing-instances rpki protocols bgp group external-as65505 type external
1153 set routing-instances rpki protocols bgp group external-as65505 import allow-
1154 all

```

1155     set routing-instances rpki protocols bgp group external-as65505 export allow-
1156     all
1157
1157     set routing-instances rpki protocols bgp group external-as65505 peer-as 65505
1158
1158     set routing-instances rpki protocols bgp group external-as65505 neighbor
1159     fd25:f:f:1::5
1160
1160     set routing-instances rpki protocols bgp group external-as65505 neighbor
1161     10.99.99.26
1162
1162     set routing-instances rpki protocols bgp group external-as65504 type external
1163
1163     set routing-instances rpki protocols bgp group external-as65504 import allow-
1164     all
1165
1165     set routing-instances rpki protocols bgp group external-as65504 export allow-
1166     all
1167
1167     set routing-instances rpki protocols bgp group external-as65504 peer-as 65504
1168
1168     set routing-instances rpki protocols bgp group external-as65504 neighbor
1169     10.99.99.18
1170
1170     set routing-instances rpki protocols bgp group external-as65504 neighbor
1171     fd24:f:f:1::4

```

1172 2.6.5 Router AS 65503 Configuration – Cisco

1173 Router AS 65503 represents an ISP without ROV capabilities. For the lab build, this router originates BGP
1174 updates from its own AS and receives and sends routes to and from its eBGP peers without performing
1175 BGP origin validation. This router peers with two transit routers, AS 65500 and AS 65502, as well as two
1176 stub ASes, AS 65504 and AS 65507.

```

1177     hostname AS65503
1178
1178     !
1179
1179     ip cef
1180
1180     ipv6 unicast-routing
1181
1181     ipv6 cef
1182
1182     !
1183
1183     interface Loopback1
1184
1184         ip address 10.30.0.1 255.255.0.0
1185
1185         ipv6 address 2003:3333:3333:3333::1/64

```

```
1186      !
1187      interface GigabitEthernet0/1
1188          ip address 10.90.90.3 255.255.255.0
1189          ipv6 address FD00:F:F:1::3/64
1190      !
1191      interface FastEthernet0/2
1192          ip address 192.168.1.251 255.255.255.0
1193      !
1194      interface GigabitEthernet0/2
1195          ip address 10.99.99.13 255.255.255.252
1196      !
1197      interface GigabitEthernet0/3
1198          description VLAN7
1199          ip address 10.99.99.21 255.255.255.252
1200          ipv6 address FD37:F:F:1::1/64
1201      !
1202      router bgp 65503
1203          bgp log-neighbor-changes
1204          bgp rpki server tcp 192.168.1.146 port 8282 refresh 10
1205          neighbor 10.90.90.1 remote-as 65501
1206          neighbor 10.90.90.2 remote-as 65502
1207          neighbor 10.90.90.10 remote-as 65500
1208          neighbor 10.99.99.14 remote-as 65504
1209          neighbor 10.99.99.22 remote-as 65507
1210          neighbor FD00:F:F:1::1 remote-as 65501
1211          neighbor FD00:F:F:1::2 remote-as 65502
```

1212 neighbor FD00:F:F:1::10 remote-as 65500
1213 neighbor FD34:F:F:1::4 remote-as 65504
1214 neighbor FD34:F:F:1::7 remote-as 65507
1215 !
1216 address-family ipv4
1217 redistribute connected
1218 redistribute static
1219 neighbor 10.90.90.1 activate
1220 neighbor 10.90.90.2 activate
1221 neighbor 10.90.90.10 activate
1222 neighbor 10.99.99.14 activate
1223 neighbor 10.99.99.22 activate
1224 no neighbor FD00:F:F:1::1 activate
1225 no neighbor FD00:F:F:1::2 activate
1226 no neighbor FD00:F:F:1::10 activate
1227 no neighbor FD34:F:F:1::4 activate
1228 no neighbor FD34:F:F:1::7 activate
1229 exit-address-family
1230 !
1231 address-family ipv6
1232 redistribute connected
1233 neighbor FD00:F:F:1::1 activate
1234 neighbor FD00:F:F:1::2 activate
1235 neighbor FD00:F:F:1::10 activate
1236 neighbor FD34:F:F:1::4 activate
1237 exit-address-family

```
1238      !
1239      ipv6 router rip procl
1240      !
1241      end
```

1242 2.6.6 Router AS 65504A Configuration – Cisco

1243 Router AS 65504A represents an enterprise edge router for AS 65504. For the lab build, this router
1244 originates BGP updates from its own AS and receives and sends routes to and from its eBGP peer, AS
1245 65502. It peers with Router AS 65504B to exchange iBGP routes.

```
1246      hostname AS65504A
1247      !
1248      ipv6 unicast-routing
1249      !
1250      interface Loopback1
1251          ip address 10.40.1.1 255.255.255.0
1252      !
1253      interface GigabitEthernet0/0/0
1254          ip address 10.40.0.1 255.255.255.0
1255          ipv6 address FD00:F:F:1::40/64
1256          ipv6 address FD34:F:F:1::4/64
1257      !
1258      interface GigabitEthernet0/0/1
1259          ip address 10.99.99.18 255.255.255.252
1260          ipv6 address FD24:F:F:1::4/64
1261      !
1262      interface GigabitEthernet0/0/2
1263          ip address 10.40.4.1 255.255.255.0
```

```
1264     ipv6 address 2004:4444:4444:4444::4/64
1265     !
1266     router bgp 65504
1267         bgp log-neighbor-changes
1268         neighbor 10.40.0.2 remote-as 65504
1269         neighbor 10.99.99.17 remote-as 65502
1270         neighbor FD24:F:F:1::2 remote-as 65502
1271     !
1272     address-family ipv4
1273         redistribute connected
1274         redistribute static
1275         no neighbor 10.40.0.2 activate
1276         neighbor 10.99.99.17 activate
1277         no neighbor FD24:F:F:1::2 activate
1278     exit-address-family
1279     !
1280     address-family ipv6
1281         redistribute connected
1282         neighbor FD24:F:F:1::2 activate
1283     exit-address-family
1284     !
1285     ip route 10.40.2.0 255.255.255.0 10.40.0.2
1286     !
1287     route-map NO-EXPORT permit 10
1288         set community no-export
1289     !
```

1290 end

1291 2.6.7 Router AS 65504B Configuration – Cisco

1292 Router AS 65504B represents an enterprise edge router for AS 65504. For the lab build, this router
1293 originates BGP updates from its own AS and receives and sends routes to and from its eBGP peer, AS
1294 65503. It peers with Router AS 65504A to exchange iBGP routes.

```
1295           hostname AS65504B
1296           !
1297           ipv6 unicast-routing
1298           !
1299           interface Loopback1
1300               ip address 10.40.2.1 255.255.255.0
1301               ipv6 address 4040:4040:4040:4242::1/64
1302           !
1303           interface GigabitEthernet0/0/0
1304               ip address 10.99.99.14 255.255.255.252
1305               ipv6 address FD34:F:F:1::4/64
1306           !
1307           interface GigabitEthernet0/0/1
1308               ip address 10.40.0.2 255.255.255.0
1309               ipv6 address FD40:F:F:1::2/64
1310           !
1311           router bgp 65504
1312               bgp log-neighbor-changes
1313               neighbor 10.40.0.1 remote-as 65504
1314               neighbor 10.99.99.13 remote-as 65503
1315               neighbor FD34:F:F:1::2 remote-as 65503
```

```
1316     neighbor FD40:F:F:1::1 remote-as 65504
1317     !
1318     address-family ipv4
1319         redistribute connected
1320         no neighbor 10.40.0.1 activate
1321         neighbor 10.99.99.13 activate
1322         no neighbor FD34:F:F:1::2 activate
1323         no neighbor FD40:F:F:1::1 activate
1324     exit-address-family
1325     !
1326     address-family ipv6
1327         redistribute connected
1328         neighbor FD34:F:F:1::2 activate
1329         neighbor FD40:F:F:1::1 activate
1330     exit-address-family
1331     !
1332     route-map NO-EXPORT permit 10
1333         set community no-export
1334     !
1335     end
```

1336 2.6.8 Router AS 65505 Configuration – Juniper

1337 Router AS 65505 represents an enterprise edge router. For the lab build, this router originates BGP
1338 updates from its own AS and receives and sends routes to and from its eBGP peers, AS 65501 and AS
1339 65502.

```
1340     set system host-name AS65505
1341     set interfaces ge-1/3/0 unit 0 family inet
```

1342 set interfaces ge-1/3/0 unit 0 family inet6
1343 set interfaces ge-1/3/1 unit 0 family inet address 10.99.99.2/30
1344 set interfaces ge-1/3/1 unit 0 family inet6 address fd15:f:f:1::5/64
1345 set interfaces ge-1/3/2 unit 0 family inet address 10.99.99.26/30
1346 set interfaces ge-1/3/2 unit 0 family inet6 address fd25:f:f:1::5/64
1347 set interfaces ge-1/3/3 unit 0 family inet address 10.50.0.1/16
1348 set interfaces ge-1/3/3 unit 0 family inet6 address 5050:5050:5050:1::5/64
1349 set interfaces lo0 unit 0 family inet address 127.0.0.1/32
1350 set routing-options autonomous-system 65505
1351 set routing-options validation group cache session 192.168.1.146 port 8282
1352 set protocols bgp group external-as65501 type external
1353 set protocols bgp group external-as65501 import validation
1354 set protocols bgp group external-as65501 export allow-direct
1355 set protocols bgp group external-as65501 peer-as 65501
1356 set protocols bgp group external-as65501 neighbor 10.99.99.1
1357 set protocols bgp group external-as65501 neighbor fd15:f:f:1::1
1358 set protocols bgp group external-as65502 type external
1359 set protocols bgp group external-as65502 import validation
1360 set protocols bgp group external-as65502 export allow-direct
1361 set protocols bgp group external-as65502 peer-as 65502
1362 set protocols bgp group external-as65502 neighbor 10.99.99.25
1363 set protocols bgp group external-as65502 neighbor fd25:f:f:1::2
1364 set policy-options policy-statement allow-all from route-filter 0.0.0.0/0
1365 orlonger
1366 set policy-options policy-statement allow-all then accept
1367 set policy-options policy-statement allow-direct term default from protocol
1368 direct

```

1369      set policy-options policy-statement allow-direct term default then accept
1370
1371      set policy-options policy-statement validation term valid from protocol bgp
1372
1373      set policy-options policy-statement validation term valid then local-preference
1374      110
1375
1376      set policy-options policy-statement validation term valid then validation-state
1377      valid
1378
1379      set policy-options policy-statement validation term valid then accept
1380
1381      set policy-options policy-statement validation term invalid from protocol bgp
1382
1383      set policy-options policy-statement validation term invalid from validation-
1384      database invalid
1385
1386      set policy-options policy-statement validation term invalid then local-
1387      preference 90
1388
1389      set policy-options policy-statement validation term invalid then validation-
1390      state invalid
1391
1392      set policy-options policy-statement validation term invalid then reject
1393
1394      set policy-options policy-statement validation term unknown from protocol bgp
1395
1396      set policy-options policy-statement validation term unknown then validation-
1397      state unknown
1398
1399      set policy-options policy-statement validation term unknown then accept

```

1390 2.6.9 Router AS 65507 Configuration – Cisco

1391 Router AS 65507 represents an enterprise edge router for AS 65507. For the lab build, this router
1392 originates BGP updates from its own AS and receives and sends routes to and from its eBGP peer, AS
1393 65503.

```

1394      hostname AS65507
1395
1396      !
1397
1398      interface Loopback1
1399
1400      ip address 10.70.0.1 255.255.0.0
1401
1402      ipv6 address 7070:7070:7070:7070::1/64

```

```
1399      !
1400      interface GigabitEthernet0/0
1401          ip address 10.99.99.22 255.255.255.252
1402          ipv6 address FD37:F:F:1::7/64
1403      !
1404      interface GigabitEthernet0/1
1405          ip address 172.16.0.1 255.255.0.0
1406      !
1407      router bgp 65507
1408          bgp log-neighbor-changes
1409          neighbor 10.99.99.21 remote-as 65503
1410          neighbor FD37:F:F:1::3 remote-as 65503
1411      !
1412      address-family ipv4
1413          redistribute connected
1414          neighbor 10.99.99.21 activate
1415          no neighbor FD37:F:F:1::3 activate
1416      exit-address-family
1417      !
1418      address-family ipv6
1419          redistribute connected
1420          neighbor FD37:F:F:1::3 activate
1421      exit-address-family
1422      !
1423      access-list 23 permit 10.10.10.0 0.0.0.7
1424      ipv6 router rip procl
```

```
1425      !
1426      end
```

1427 2.6.10 Router AS 65508 Configuration – Cisco

1428 Router AS 65508 represents a hijacker masquerading as an enterprise edge router. For the lab build, this
1429 router originates BGP updates for routes that are held by other ASes (i.e., for routes for which it is not
1430 authorized to originate updates), in order to demonstrate route hijacks.

```
1431      hostname AS65508
1432      !
1433      ipv6 unicast-routing
1434      ipv6 cef
1435      !
1436      interface Loopback1
1437          ip address 10.80.0.1 255.255.0.0
1438          ipv6 address 8080:8080:8080:8080::1/64
1439      !
1440      interface GigabitEthernet0/0
1441          ip address 10.99.99.30 255.255.255.252
1442          ipv6 address FD00:F:F:1::61/64
1443          ipv6 address FD08:F:F:1::8/64
1444      !
1445      interface GigabitEthernet0/1
1446          ip address 172.16.8.1 255.255.255.0
1447      !
1448      router bgp 65508
1449          bgp log-neighbor-changes
1450          neighbor 10.99.99.29 remote-as 65500
```

```
1451     neighbor FD08:F:F:1::10 remote-as 65500
1452     !
1453     address-family ipv4
1454         redistribute connected
1455         neighbor 10.99.99.29 activate
1456         no neighbor FD08:F:F:1::10 activate
1457     exit-address-family
1458     !
1459     address-family ipv6
1460         redistribute connected
1461         neighbor FD08:F:F:1::10 activate
1462     exit-address-family
1463     !
1464     ipv6 router rip procl
1465     !
1466     end
```

1467 2.6.11 Cisco IOS XRv Router Configuration

1468 The Cisco IOS XRv software was also used to perform many of the functional tests, as many ISPs
1469 currently use it in their network environment. The baseline configuration is provided below. Depending
1470 on the test case, this router can replace any other router shown in [Figure 1-2](#), in order to properly
1471 perform the test.

```
1472     RP/0/RP0/CPU0:ios#sho run
1473     !! IOS XR Configuration version = 6.4.1
1474     !
1475     interface MgmtEth0/RP0/CPU0/0
1476         ipv4 address 192.168.1.201 255.255.255.0
1477         ipv6 address fd00:f:f:1::201/64
```

```
1478      !
1479      route-policy pass-all
1480          pass
1481      end-policy
1482      !
1483      router bgp 65501
1484          bgp router-id 1.1.1.1
1485          rpki server 192.168.1.146
1486          transport tcp port 8282
1487          refresh-time 15
1488      !
1489      address-family ipv4 unicast
1490          bgp bestpath origin-as allow invalid
1491      !
1492      address-family ipv6 unicast
1493          bgp bestpath origin-as allow invalid
1494      !
1495      neighbor 192.168.1.62
1496          remote-as 65501
1497          address-family ipv4 unicast
1498              route-policy pass-all in
1499              route-policy pass-all out
1500      !
1501      !
1502      neighbor fd00:f:f:1::62
1503          remote-as 65501
```

```
1504         address-family ipv6 unicast
1505             route-policy pass-all in
1506             route-policy pass-all out
1507         !
1508     !
1509 !
1510 end
```

Appendix A List of Acronyms

AFRINIC	African Network Information Center
APNIC	Asia-Pacific Network Information Center
ARIN	American Registry for Internet Numbers
AS	Autonomous System
BGP	Border Gateway Protocol
BGPsec	Border Gateway Protocol Security
BGP-SRx	BGP Secure Routing Extension
BIO	BGPSEC-IO
CA	Certificate Authority
CPU	Central Processing Unit
eBGP	Exterior Border Gateway Protocol
Gb	Gigabyte(s)
GbE	Gigabit(s) Ethernet
GUI	Graphical User Interface
iBGP	Interior Border Gateway Protocol
IETF	Internet Engineering Task Force
IOS	Internetwork Operating System
IP	Internet Protocol
ISP	Internet Service Provider
IT	Information Technology
JUNOS	Juniper Operating System
LACNIC	Latin America and Caribbean Network Information Center
NCCoE	National Cybersecurity Center of Excellence
NIST	National Institute of Standards and Technology
OS	Operating System

RFC	Request for Comments
RIPE NCC	Réseaux IP Européens Network Coordination Centre
RIR	Regional Internet Registry
ROA	Route Origin Authorization
ROV	Route Origin Validation
RPKI	Resource Public Key Infrastructure
RRDP	RPKI Repository Delta Protocol
RTR	Router
SIDR	Secure Inter-Domain Routing
SP	Special Publication
TAL	Trust Anchor Locator
URL	Uniform Resource Locator
VLAN	Virtual Local Area Network
VM	Virtual Machine
VRP	Validated ROA Payload
WAN	Wide Area Network

1512

Appendix B References

[NIST BGP-SRx]	<i>BGP Secure Routing Extension (BGP SRx) Prototype</i> , National Institute of Standards and Technology, [website]. https://www.nist.gov/services-resources/software/bgp-secure-routing-extension-bgp-srx-prototype
[NIST SP 800-54]	D. R. Kuhn, K. Sriram, and D. Montgomery, <i>Border Gateway Protocol Security</i> , NIST SP 800-54, July 2007. http://csrc.nist.gov/publications/nistpubs/800-54/SP800-54.pdf
[NIST SP 800-160]	<i>Systems Security Engineering: An Integrated Approach to Building Trustworthy Resilient Systems</i> , NIST SP 800-160 Second Public Draft, National Institute of Standards and Technology, November 2016. http://csrc.nist.gov/publications/drafts/800-160/sp800_160_second-draft.pdf
[RFC 6480]	M. Lepinski and S. Kent, <i>An Infrastructure to Support Secure Internet Routing</i> , RFC 6480, February 2012. https://tools.ietf.org/html/rfc6480
[RFC 6482]	M. Lepinski, S. Kent, and D. Kong, <i>A Profile for Route Origin Authorizations (ROAs)</i> , RFC 6482, February 2012. https://tools.ietf.org/html/rfc6482
[RFC 6811]	P. Mohapatra, J. Scudder, D. Ward, R. Bush, and R. Austein, <i>BGP Prefix Origin Validation</i> , RFC 6811, January 2013. https://tools.ietf.org/pdf/rfc6811.pdf
[RFC 7115]	R. Bush, <i>Origin Validation Operation Based on the Resource Public Key Infrastructure (RPKI)</i> , RFC 7115, January 2014. https://tools.ietf.org/html/rfc7115
[RIPE Tools]	<i>Tools and Resources</i> , RIPE Network Coordination Centre (NCC), [website]. https://www.ripe.net/manage-ips-and-asns/resource-management/certification/tools-and-resources