PEC 3 Security Analyses - Part 2 Securing the BGPv4

Máster Interuniversitario de Seguridad de las TIC

Advisor: Joan Borrell Viader

Student: José María Foces Vivancos

June 2018

Contents

1. Introduction

This document contains the analyses of Prefix Origin Validation defense, inside the scope of Secure Inter-Domain Routing.

The environment used is exactly the environment defined at (Foces Vivancos, Securing The BGPv4: Working Environment) and it inherits all security measures from the previous document (Foces Vivancos, Securing The BGPv4: PEC2 Security Analyses - Part 1).

1.1. Objectives

The main objective is to illustrate the security improvement provided by protections implemented in routing stacks nowadays and configurations explained at (RFC7454 - BGP Operations and Security) on section SIDR - Secure Inter-Domain Routing, in front of some attacks. Therefore, this document focuses on RPKI based Prefix Origin Validation. Note that BGPSEC cannot be deployed currently yet over Frrouting. But there (NIST) offers BGP-SRx testing suite that implements it.

While doing it, the automated generation of this environment is improved to integrate explained protections. For each security measure a new environment is generated. This way the security of the whole network is improved step by step.

The document preceding this one showed the performance of defenses applied on BGPv4 speakers, sessions and routing. This one, extends the latter to show the benefits of implementing RPKI, part of SIDR Infrastructure, against Prefix and Sub-prefix Hijack attack mainly while introducing new ones while shows the performance of NextHop Hijack against this security measure.

2. Defeating Prefix and Sub-prefix hijacks through RPKI

This section gives a brief introduction to RPKI, explains the configuration applied over the environment and analyzes the performance against Prefix and Sub-prefix hijack attacks.

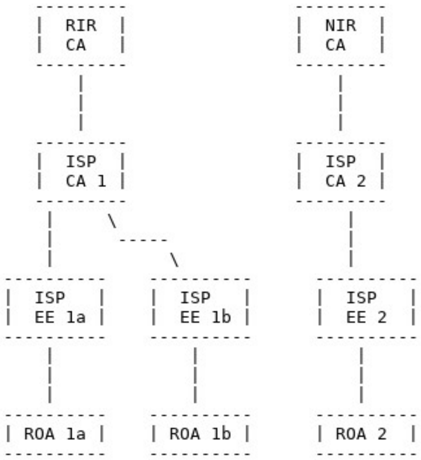
2.1. RPKI Basis

Resource Public Key Infrastructure is a part of SIDR (RFC6480 - An Infrastructure to Support Secure Internet Routing). RPKI defines the application of public key infrastructure to the arborescent structure of IP networks. But it's bigger than that. The following standards define critical parts such as X509 certificate extensions or ROAs' structure (note that only the most important ones are shown):

- (RFC6841 - A Profile for Resource Certificate Repository Structure),
- (RFC6842 - A Profile for Route Origin Authorizations (ROAs))
- (RFC6487 - A Profile for X.509 PKIX Resource Certificates)
- (RFC6488 - Signed Object Template for the Resource Public Key Infrastructure (RPKI))
- (RFC6493 - The Resource Public Key Infrastructure (RPKI) Ghostbusters Record)
- (RFC6810 - The Resource Public Key Infrastructure (RPKI) to Router Protocol)
- (RFC6811 - BGP Prefix Origin Validation)

Certificates are commonly used to identify a subject, but this is not the case here, certificates are referred as Resource Certificates and they attest the allocation by the issuer of IP addresses or AS numbers to the subject.

IANA is defined as the ROOT Certification Authority, RIRs as subordinate CAs, NIRs/LIRs/ISPs as subordinates of the previous ones and so on. The picture below, from (RFC6480 - An Infrastructure to Support Secure Internet Routing), shows the concrete application:



In a nutshell we can say that each prefix – asn pair has its' own certificate and it's signed by the CA that owns this prefix, the issuer.

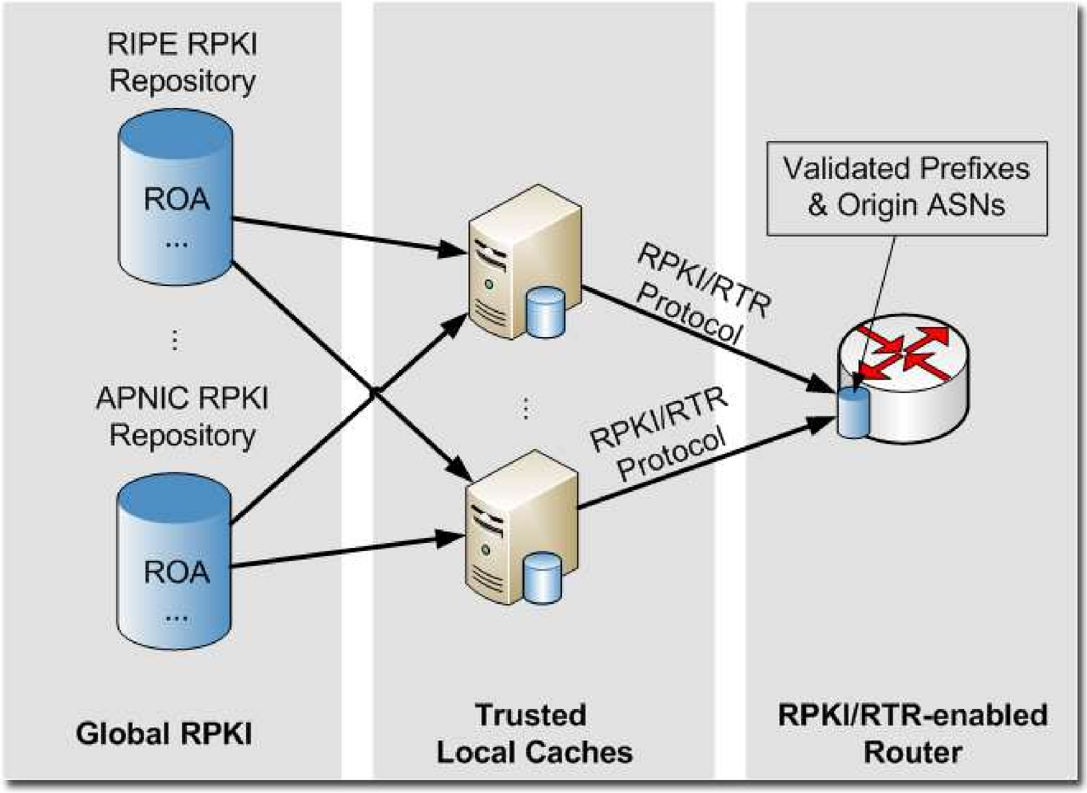
The trust anchor, the CAs that some network operator trusts, are ISPs, LIRs, NIRs, RIRs and IANA. They can sign and update repositories with signed resource certificates to authorize a third party ASN to advertise a certain prefix inside them networks.

But the question now is:

how do we take this information to the routers, so they can validate against the RPKI repository ?

Obviously, make the routers to perform cryptographic validations and to have the full repository stored inside is not currently affordable, considering the size of the Internet, it won't be scalable enough to adapt to the needs. Because of that, the certificates are stored in repositories. Each network operator should clone and keep up to date them. This is done by Validators, that perform the hard work of checking resource certificates' signatures, pulled from the repositories. Now, a new element arises to answer the question: the intermediate caches.

These intermediate caches speak both with local or remote repository clones and routers.



These intermediate caches store the validator output to offer them to the routers. Therefore, the cache stores Validated ROA Payloads, VRPs.

2.2. Practical application

As explained at (Foces Vivancos, Securing The BGPv4: Working Environment) 2.3.2 the RPKI-RTR speaker, the cache, receives a JSON file defining the routing policy and this way, it acts as trusted cache for routers. Regarding environment definition, the cache server is connected to routers through the management network and RPKI/RTR travels in plain text through this assumed secure network.

To accomplish defined objectives and show the performance of this security measure in front of Prefix and Sub-prefix hijacks, all routers have been configured to check advertisements and act as stated on (RFC7454 - BGP Operations and Security):

- *If a corresponding ROA (Route Origin Authorization) is found and is valid, then the prefix SHOULD be accepted.*
- *If the ROA is found and is INVALID, then the prefix SHOULD be discarded.*
- *If a ROA is not found, then the prefix SHOULD be accepted, but the corresponding route SHOULD be given a low preference.*

That is a very smart tactic, because it discards invalid advertisements while keeping legacy operation untouched.

On the testing environment, all routers have been updated with the following configuration:

```
route-map rpki deny 10
 match rpki invalid
 set local-preference 10
!
route-map rpki permit 20
 match rpki notfound
 set local-preference 20
!
route-map rpki permit 30
 match rpki valid
 set local-preference 30
!
route-map rpki permit 40
!
route-map AntiSpoofNextHop permit 10
 call rpki
 set ip next-hop peer-address
!
```

These route maps refer to the following result of the validation of a prefix – asn pair against the RPKI/RTR Table: invalid, not found or valid. As we can find on (RFC6810 -

The Resource Public Key Infrastructure (RPKI) to Router Protocol), these prefix origin validation results are assigned in the following cases:

- *NotFound: No VRP Covers the Route Prefix.*
- *Valid: At least one VRP Matches the Route Prefix.*
- *Invalid: At least one VRP Covers the Route Prefix, but no VRP Matches it.*

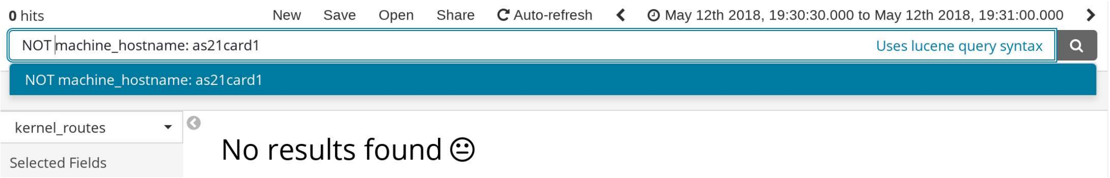
As the standard defines:

Covered: A Route Prefix is said to be Covered by a VRP when the VRP prefix length is less than or equal to the Route prefix length, and the VRP prefix address and the Route prefix address are identical for all bits specified by the VRP prefix length. (That is, the Route prefix is either identical to the VRP prefix or more specific than the VRP prefix.)

Matched: A Route Prefix is said to be Matched by a VRP when the Route Prefix is covered by that VRP, the Route prefix length is less than or equal to the VRP maximum length, and the Route Origin ASN is equal to the VRP ASN.

2.3. Re-executing Prefix and Sub-prefix hijacks

After the environment is deployed. The steps to perform the prefix a sub-prefix hijacks are repeated. There were no changes on the kernel routing tables. All routers discarded the advertisements, since the AS21 is not authorized by the ROAs to advertise these prefixes:



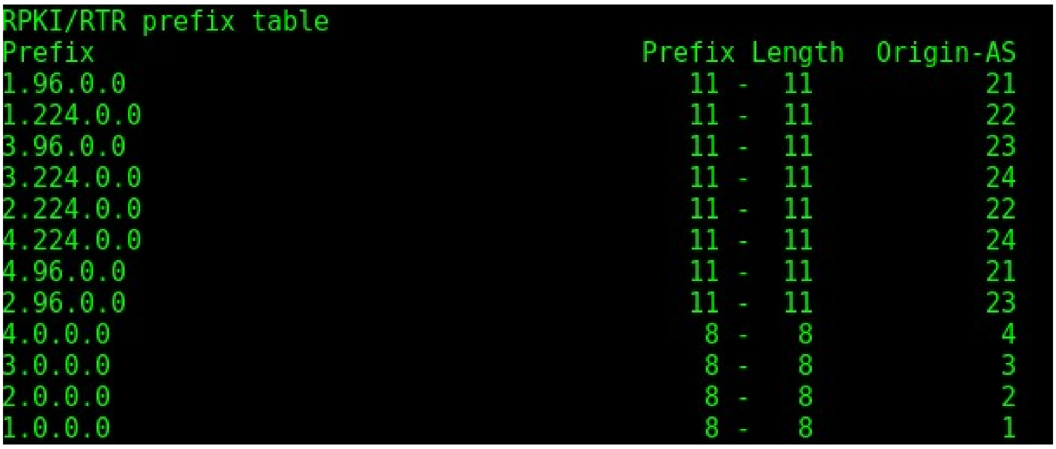
NOT machine_hostname: as21card1

NOT machine_hostname: as21card1

No results found 😐

Obviously, AS21Card1 had updates on its routing table due to the declaration of the virtual interfaces and IP addresses.

The RPKI/RTR prefix table on the routers contain the values shown below. Therefore, the rpki validation returns invalid for the advertisements made by AS21:



```
RPKI/RTR prefix table
Prefix                                  Prefix Length  Origin-AS
1.96.0.0                                   11 -  11           21
1.224.0.0                                  11 -  11           22
3.96.0.0                                   11 -  11           23
3.224.0.0                                  11 -  11           24
2.224.0.0                                  11 -  11           22
4.224.0.0                                  11 -  11           24
4.96.0.0                                   11 -  11           21
2.96.0.0                                   11 -  11           23
4.0.0.0                                     8 -   8            4
3.0.0.0                                     8 -   8            3
2.0.0.0                                     8 -   8            2
1.0.0.0                                     8 -   8            1
```

Combining that with the call to the route map upon receiving advertisements, routers are able to decide that the routes are not valid and therefore, drop them. Note that in this case the whole network checks advertisements against the RPKI-RTR Cache. In case that one of the routers is not checking against it, the effect spreads around until it reaches a router that is validating against the cache.

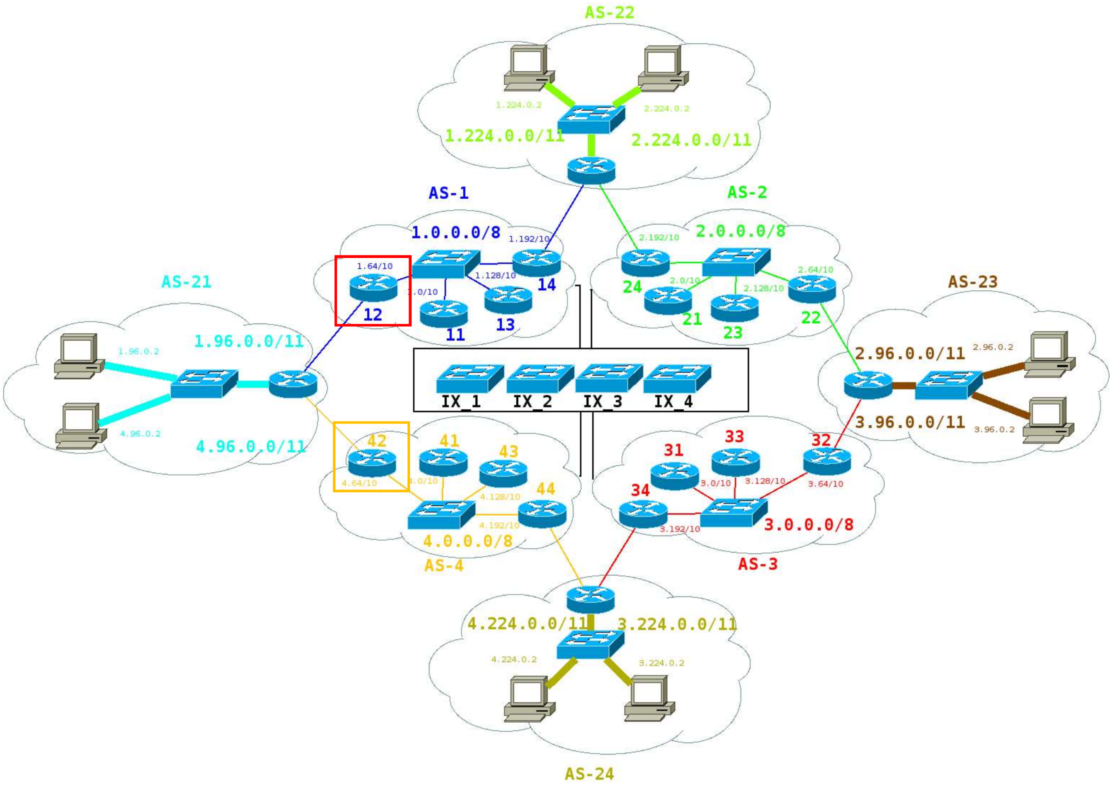
3. Next Hop Hijack

Both prefix and sub-prefix hijacks are not possible if RPKI is deployed and routers are working with it. But, that does not provide protection in terms of the AS path. The only restriction it provides is that the last AS on the path is the owner of the advertised prefix.

Therefore, the attacker can advertise any route that ends on the victim AS.

Over the current simulation, AS21 keeps willing to hijack traffic from AS23. To simulate that, let’s imagine that AS21 offers internet service to his clients and therefore, AS1 and AS4 cannot filter advertisements with more than one AS on the path made by AS21.

Regarding the structure:



Remember that we had protected Tier1 routers offering services to clients with the following access-list:

```
ip as-path access-list StrictProviderAS21 permit ^21$
ip as-path access-list StrictProviderAS21 deny .*
```

In order to simulate that case, where AS21 offers service to its’ virtual clients, we disable this protection at As1Card2 and As4Card2.

To illustrate the effect of VRPs, stored in the cache, two cases of NextHop Hijack are analyzed. On the first one, we misconfigure the maxLength attribute of each ROA on the cache server by increasing the maxLength attribute to 32, therefore, that makes valid any route advertisement that ends at AS23 for any prefix behind 3.96.0.0/11. On the second one it takes the prefix length and therefore, it invalidates any announcement for prefix 3.96.0.0/11 that do not end at AS23. Note that any sub-prefix is also invalid in this case.

3.1. Wide Open Max Prefix Length VRPs

```
roas" : [ {
…
      "asn" : "AS23",
      "prefix" : "3.96.0.0/11",
      "maxLength" : 32,
      "links" : {}
    },
…
```

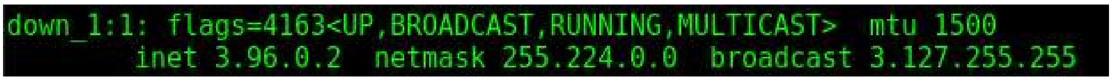
Over this context, the attacker AS21 has to change its configuration to allow advertisements for networks 2.96.0.0/1[1|2] and 3.96.0.0/1[1|2]:

```
ip prefix-list t2-external-adv seq 15 permit 3.96.0.0/12 le 32
ip prefix-list t2-external-adv seq 20 permit 2.96.0.0/12 le 32
ip prefix-list t2-external-adv seq 25 permit 3.96.0.0/11 le 32
ip prefix-list t2-external-adv seq 30 permit 2.96.0.0/11 le 32
```

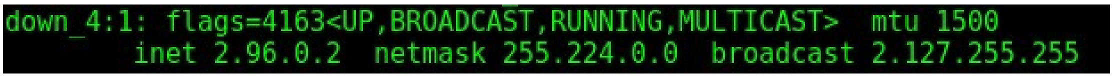
After that, it defines a route-map to add AS23 to advertisements made for certain networks (2.96.0.0/11 and 3.96.0.0/11).

```
route-map NextHopHijack permit 30
 set as-path prepend 23
```

It configures the interfaces as it did previously, on the Prefix and Sub-Prefix hijack attacks:



```
down_1:1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 3.96.0.2  netmask 255.224.0.0  broadcast 3.127.255.255
```



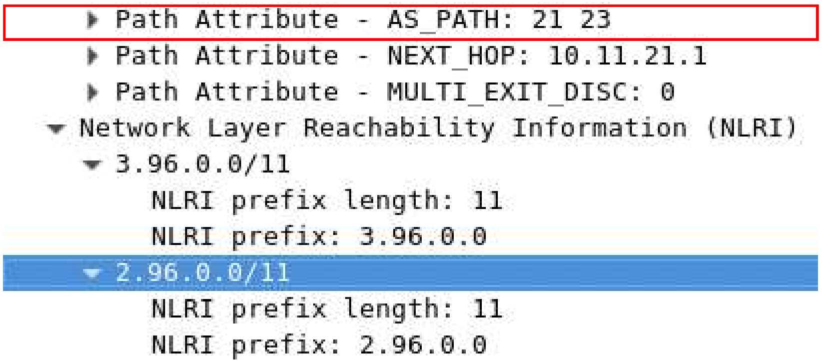
```
down_4:1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 2.96.0.2  netmask 255.224.0.0  broadcast 2.127.255.255
```

To finish, it just declares the bogus route announcements, since these are not valid in the AS graph.

For this case trying to perform NextHop hijack for the whole prefixes advertised by AS23:

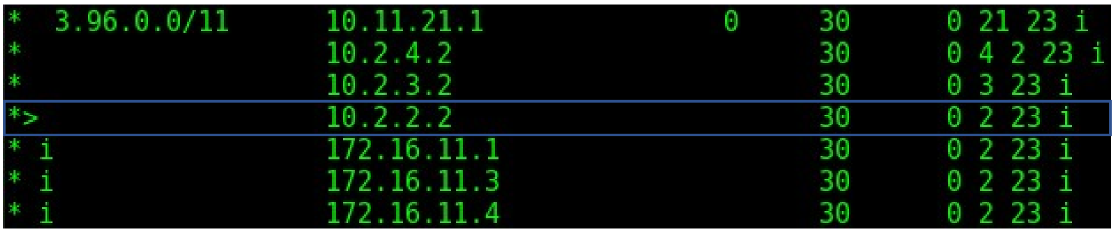
```
router bgp 21
        network 3.96.0.0/11 route-map NextHopHijack
        network 2.96.0.0/11 route-map NextHopHijack
```

That produces route advertisements as follows:



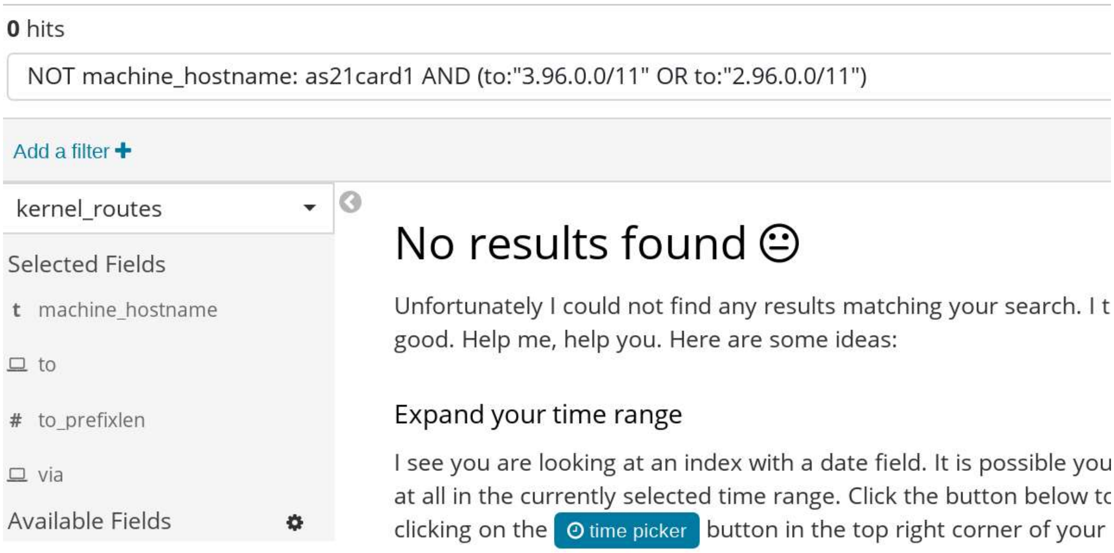
```
    ▸ Path Attribute - AS_PATH: 21 23
    ▸ Path Attribute - NEXT_HOP: 10.11.21.1
    ▸ Path Attribute - MULTI_EXIT_DISC: 0
  ▾ Network Layer Reachability Information (NLRI)
    ▾ 3.96.0.0/11
        NLRI prefix length: 11
        NLRI prefix: 3.96.0.0
    ▾ 2.96.0.0/11
        NLRI prefix length: 11
        NLRI prefix: 2.96.0.0
```

The impact, the routing table of As1Card2 has a new entry:



```
*  3.96.0.0/11      10.11.21.1          0    30       0 21 23 i
*                   10.2.4.2                 30       0 4 2 23 i
*                   10.2.3.2                 30       0 3 23 i
*>                  10.2.2.2                 30       0 2 23 i
* i                 172.16.11.1              30       0 2 23 i
* i                 172.16.11.3              30       0 2 23 i
* i                 172.16.11.4              30       0 2 23 i
```

The BGP route selection algorithm is not selecting these routes, since it has already another way to reach the given prefixes. The paths are of the same length in terms of AS-Path. Therefore, it keeps the route through 10.2.2.2 through AS path 2 23. Considering that, there is no impact on the effective forwarding table (the kernel routing table) on any router on the network:



0 hits

NOT machine_hostname: as21card1 AND (to:"3.96.0.0/11" OR to:"2.96.0.0/11")

No results found ☹

Unfortunately I could not find any results matching your search. I t good. Help me, help you. Here are some ideas:

Expand your time range

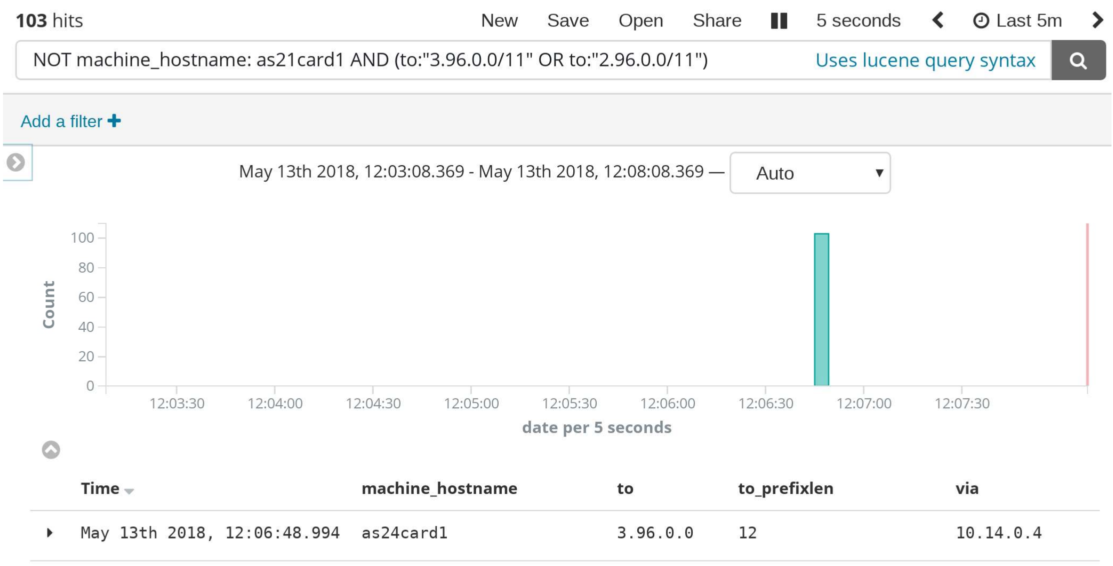
I see you are looking at an index with a date field. It is possible you at all in the currently selected time range. Click the button below tc clicking on the ⏲ time picker button in the top right corner of your

It's important to regard that the routes are accepted since they are valid in RPKI terms, but they are not used since others of the same length are currently in use. The attacker may try to reset BGPv4 sessions of this router with others until his route is selected. Therefore, that demonstrates that every protection means.

On this context, the traffic volume measurement is not applicable, since there are no changes on effective routing tables.

With that in mind, lets trick it by both the BGP route selection algorithm and Longest Prefix Matching, by advertising the prefixes 3.96.0.0/12 and 2.96.0.0/12:

Once done it produces 103 kernel routing table updates on the whole network.



NOT machine_hostname: as21card1 AND (to:"3.96.0.0/11" OR to:"2.96.0.0/11")   Uses lucene query syntax

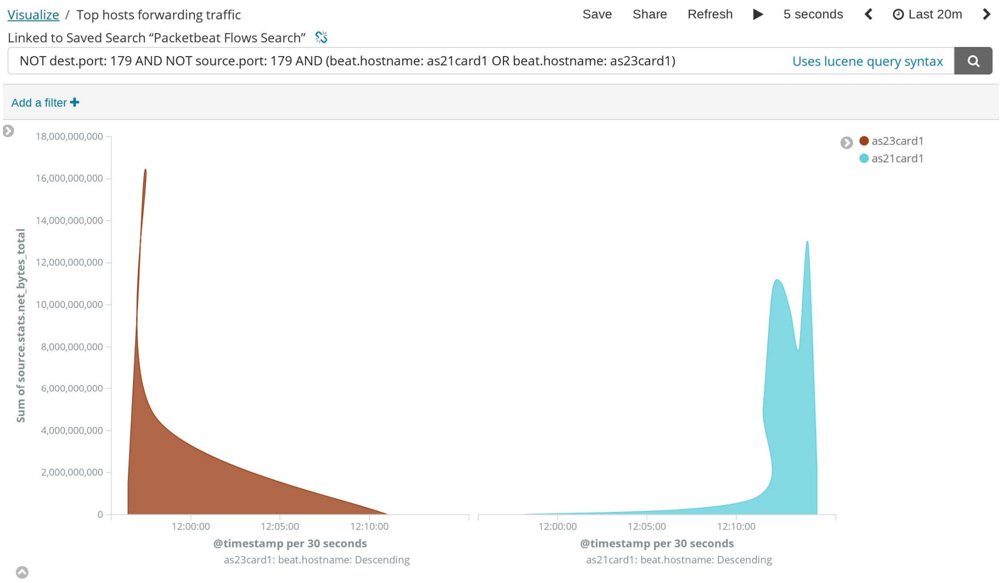
May 13th 2018, 12:03:08.369 - May 13th 2018, 12:08:08.369 —

Time	machine_hostname	to	to_prefixlen	via
May 13th 2018, 12:06:48.994	as24card1	3.96.0.0	12	10.14.0.4

These advertisements are:

- Accepted because of the value (32) of the maxLength attribute of VRPs in the cache server. This authorizes advertisements of prefixes from length 11 to 32 from AS23 for the prefix 3.96.0.0/11 and 2.96.0.0/11.
- Once accepted, the advertised routes are introduced in the kernel routing tables and then, the Longest Prefix Matching makes the rest.

The histogram below shows that the attacker has successfully hijacked AS23's traffic:



Linked to Saved Search "Packetbeat Flows Search"

NOT dest.port: 179 AND NOT source.port: 179 AND (beat.hostname: as21card1 OR beat.hostname: as23card1)   Uses lucene query syntax

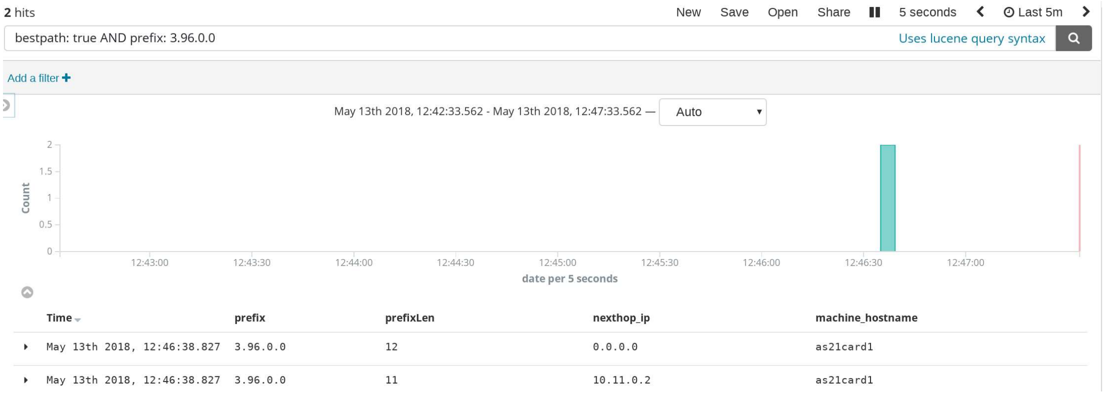
3.2. Short Max Prefix Length VRPs

Let’s roll back the configuration to define valid ROAs for prefixes with exact lengths.

The VRPs are now declared as:

```
roas" : [ {
…
      "asn" : "AS23",
      "prefix" : "3.96.0.0/11",
      "maxLength" : 11,
      "links" : {}
    },
…
```

This way, the sub-prefixes advertisements, 3.96.0.0/12 and 2.96.0.0/12 are not valid and they are discarded by all routers in the network:



2 hits

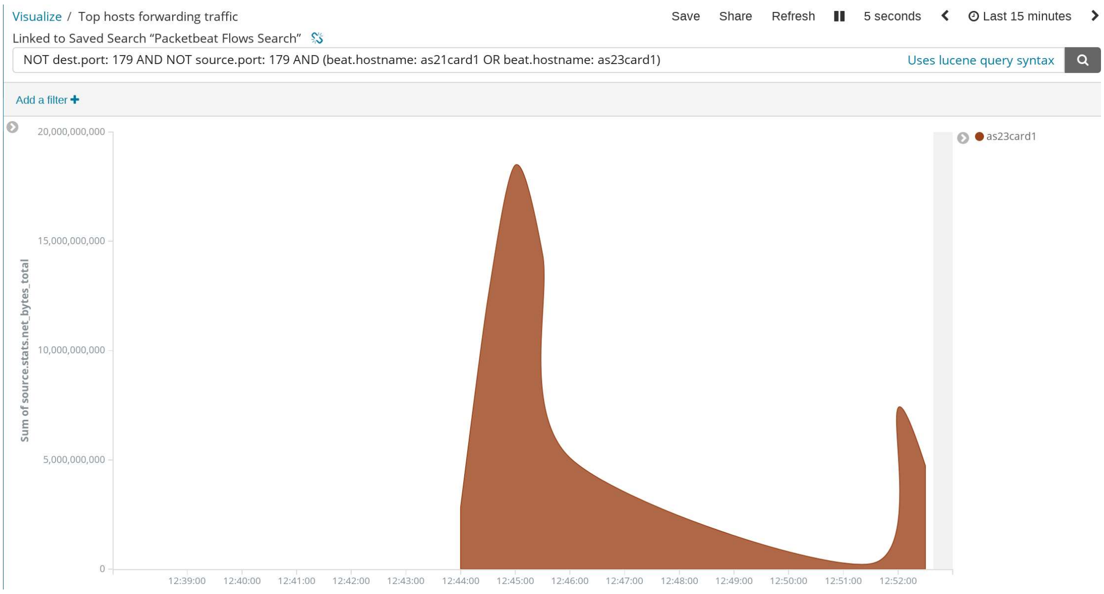
bestpath: true AND prefix: 3.96.0.0

May 13th 2018, 12:42:33.562 - May 13th 2018, 12:47:33.562 —

Time	prefix	prefixLen	nexthop_ip	machine_hostname
May 13th 2018, 12:46:38.827	3.96.0.0	12	0.0.0.0	as21card1
May 13th 2018, 12:46:38.827	3.96.0.0	11	10.11.0.2	as21card1

Note that the only one who has changes on the routing table is As21Card1, the attacker.

The attack is not successful when the ROAs are correctly defined:



Visualize / Top hosts forwarding traffic

Linked to Saved Search “Packetbeat Flows Search”

NOT dest.port: 179 AND NOT source.port: 179 AND (beat.hostname: as21card1 OR beat.hostname: as23card1)

On this point the attacker can only attempt to perform a NextHop hijack against valid prefixes on the ROAs (VRPs) otherwise routes will be dropped.

The attack will be successful when the BGP route selection algorithm selects the bogus routes. This will happen only when the attacker announces the shortest path the victim router knows to the legitimate AS.

4. Bibliography

"BGP Routing Table Analysis Reports." 2018. <http://bgp.potaroo.net/>.

Foces Vivancos, José María. "Rehtse." 2016. <https://github.com/JmFoces/Rehtse>.

—. "Securing The BGPv4: PEC2 Security Analyses - Part 1." 2018.

—. "Securing The BGPv4: Working Environment." 2018.

"IANA - IPv4 Address Space." 1998. <https://www.iana.org/assignments/ipv4-address-space/ipv4-address-space.xhtml>.

"IRRToolSet." 2002. <https://github.com/irrtoolset/irrtoolset>.

"JohnTheRipper." 1996. <https://github.com/magnumripper/JohnTheRipper>.

"Linux Kernel Documentation." 1991. <https://www.kernel.org/doc/Documentation/>.

Mutually Agreed Norms for Routing Security. 2014. <https://www.manrs.org>.

NIST. "BGP Secure Routing Extensions." (n.d.). <https://www.nist.gov/services-resources/software/bgp-secure-routing-extension-bgp-srx-prototype>.

"RFC1105 - A Border Gateway Protocol (BGP)." 1989.

"RFC1163 - A Border Gateway Protocol (BGP)." 1990.

"RFC1267 - A Border Gateway Protocol 3 (BGP-3)." 1991.

"RFC1337 - TIME-WAIT Assassination Hazards in TCP." 1992. <https://www.ietf.org/rfc/rfc1337.txt>.

"RFC1654 - A Border Gateway Protocol 4 (BGP-4)." 1994.

"RFC1771 - A Border Gateway Protocol 4 (BGP-4)." 1995.

"RFC2385 - Protection of BGP Sessions via the TCP MD5 Signature Option." 1998.

"RFC2827 - Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing." 2000.

"RFC3013 - Recommended ISP Security." 2000.

"RFC3704 - Ingress Filtering for Multihomed Networks." 2004. <https://tools.ietf.org/html/rfc3704>.

"RFC4012 - Routing Policy Specification Language next generation (RPSLng)." 2005.

"RFC4271 - A Border Gateway Protocol 4 (BGP-4)." 2006.

"RFC4272 - BGP Vulnerability Analysis." 2006.

"RFC5082 - The Generalized TTL Security Mechanism (GTSM)." 2007.

"RFC5925 - The TCP Authentication Option." 2010.

"RFC5961 - Improving TCP's Robustness to Blind In-Window Attacks." 2010. <https://tools.ietf.org/html/rfc5961>.

"RFC6192 - Plane, Protecting the Router Control." 2011. <https://tools.ietf.org/html/rfc6192>.

"RFC6480 - An Infrastructure to Support Secure Internet Routing." 2012.

"RFC6483 - Validation of Route Origination Using the Resource Certificate Public Key Infrastructure (PKI) and Route Origin Authorizations (ROAs)." 2012.

"RFC6487 - A Profile for X.509 PKIX Resource Certificates." n.d. <https://tools.ietf.org/html/rfc6487>.

"RFC6488 - Signed Object Template for the Resource Public Key Infrastructure (RPKI)." 2012.

"RFC6493 - The Resource Public Key Infrastructure (RPKI) Ghostbusters Record." 2012. <https://tools.ietf.org/html/rfc6493>.

"RFC6810 - The Resource Public Key Infrastructure (RPKI) to Router Protocol." 2013.

"RFC6811 - BGP Prefix Origin Validation." 2013.

"RFC6841 - A Profile for Resource Certificate Repository Structure." 2012. <https://tools.ietf.org/html/rfc6481>.

"RFC6842 - A Profile for Route Origin Authorizations (ROAs)." n.d. <https://tools.ietf.org/html/rfc6482>.

"RFC7196 - Making Route Flap Damping Usable." 2014. <https://tools.ietf.org/html/rfc7196>.

"RFC7454 - BGP Operations and Security." 2015.

"RFC7715 - Origin Validation Operation Based on the Resource Public Key Infrastructure (RPKI)." 2016.

"RFC791 - INTERNET PROTOCOL." 1981. <https://tools.ietf.org/html/rfc791>.

"RFC7947 - Internet Exchange BGP Route Server." 2016.

"RFC8205 - BGPsec Protocol Specification." 2017.

Robert Lychev, Michael Schaipira & Sharong Goldberg. "Rethinking Security for Internet Routing." 2016.

"TCPMD5 Signature - Socket Programing Examples on Linux." 2015. <https://criticalindirection.com/2015/05/12/tcp_md5sig>.

"Ubuntu - Kernel Security Settings." 2006. <https://wiki.ubuntu.com/ImprovedNetworking/KernelSecuritySettings>.