A first step towards checking BGP routes in the dataplane

Thomas Wirtgen*
ICTEAM, UCLouvain
Louvain-la-Neuve, Belgium
thomas.wirtgen@uclouvain.be

Olivier Bonaventure
ICTEAM, UCLouvain
Louvain-la-Neuve, Belgium
olivier.bonaventure@uclouvain.be

ABSTRACT

BGP is a fragile routing protocol since it is based on an implicit system of trust between the Autonomous Systems (AS) participating in the exchange of routes on the Internet. Any router can announce the routes it wants without being the owner. Due to the lack of a validation system for the announcements made by BGP routers, a series of RFCs published after the release of BGP have partially solved this problem by introducing the Resource Public Key Infrastructure (RPKI).

In this paper, we aim to complement the security mechanisms of BGP by introducing a new active control system. We propose to validate BGP paths in the dataplane. We extend the BGP implementation of FRRouting (an open source Internet routing protocol suite) to demonstrate the feasibility of our approach. Finally, we discuss the potential of an active system in a routing protocol to both secure BGP announcements and improve the routing decision.

CCS CONCEPTS

• **Networks** → **Routing protocols**; **Network architectures**; Network management.

KEYWORDS

Routing Protocols, BGP, Dataplane Path Validation

ACM Reference Format:
Thomas Wirtgen and Olivier Bonaventure. 2022. A first step towards checking BGP routes in the dataplane. In *ACM SIGCOMM 2022 Workshop on Future of Internet Routing & Addressing (FIRA '22), August 22, 2022, Amsterdam, Netherlands.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3527974.3545723

1 INTRODUCTION

The Border Gateway Protocol (BGP) [41] is probably the most important routing protocol in today's Internet since it exchanges interdomain routes. As of this writing, more than 80,000 different Autonomous Systems (AS) use BGP to exchange routes [26]. For BGP, each AS is identified by a unique AS number. An AS announces each of its assigned IP prefixes by sending them inside BGP messages to its peers. There are roughly two types of ASes: stubs and transits [17]. Most ASes are stubs. Transit ASes provide transit services for their customers. For this, they re-announce the prefixes learned from their own peers. When announcing these prefixes, they add their own AS number inside the AS-Path, which is an important attribute of all BGP messages. BGP uses the AS-Path to detect routing loops, but also to select the best path to reach each prefix.

BGP was designed when the Internet was a research network [29]. The first BGP design did not include any security feature. The initial assumption was that network operators could be trusted. Over the years, this assumption appeared to be too optimistic. First, network operators, even trusted ones, sometimes make mistakes. This is known as the *fat finger* problem in the operator community. If a network operator makes a mistake when typing an IP prefix, it can advertise an IP prefix that belongs to another network. This and other types of network configuration errors occurred many times during the last decades [36]. Second, some network operators, either maliciously or through attacks, have advertised IP prefixes that belong to other ASes [46]. Researchers have identified various forms of such BGP hijacking [8].

To cope with these problems, and after many debates, the Internet Engineering Task Force (IETF) finally adopted the Resource Public Key Infrastructure (RPKI) model [4]. In a nutshell, the RPKI uses cryptography to create certificates that bind an IP prefix to an AS number. The Regional Internet Registries publish the RPKI certificates that indicate which AS can legitimately announce a given prefix. Thanks to these certificates, when a BGP router receives a new route, it can compare the origin of the route (i.e., the last AS in the AS-Path) with the information contained in the certificate. If they match, the router considers the route as valid and it can be used and reannounced by the router. Otherwise, the router may decide to discard the route. Researchers and network operators have studied the deployment and the usage of the RPKI during the last decade [10, 44, 49].

While the RPKI is slowly getting deployed, it only allows a BGP router to verify that a received route was announced by its legitimate origin AS. The RPKI does not verify the entire AS-Path, although there are proposals to perform such validation [2, 12]. In the longer term, BGPSec should improve [27] the security of interdomain routing, but its deployment has not yet started.

In this paper, we propose a new method to check the routes advertised by BGP. Instead of directly inserting the route into the routing table, we first check whether the route prefix is reachable in the dataplane. We ask the router to contact a given destination in the prefix sent in the BGP Update. If this destination is reachable, the route can be considered valid and thus can be added to the routing table. In Section 2, we describe a new architecture that can be used to check if paths are reachable. Next, to prove that it is possible to add the intelligence of the dataplane in the control plane, we

*Thomas Wirtgen is supported by a grant from F.R.S.-FNRS FRIA.

FIRA '22, August 22, 2022, Amsterdam, Netherlands

ACM ISBN 978-1-4503-9328-7/22/08…$15.00
https://doi.org/10.1145/3527974.3545723

implemented a new route validation method in FRRouting, an open-source implementation of BGP. Finally, we discuss the implications and new possibilities of using dataplane information to influence routing in Section 3.

2 VALIDATING BGP ROUTES IN THE DATAPLANE

During a maintenance routine in an AS, accidental configuration errors can disrupt connectivity. Configuration errors occur every day and can impact many prefixes [36]. All traffic passing through the misconfigured AS can cause losses on the affected path and thus can blackhole traffic. When BGP Updates are received from misconfigured routers, the route contained in the update is by default considered to be reachable in the control plane. Therefore, the route is a potential candidate for forwarding traffic, even if it is not reachable in the dataplane. This example raises an interesting question: **How can a router validate the reachabilty of a route received from a peer?** This is a complex question because an IP prefix may contain addresses for endusers, servers or even infrastructure such as routers. From the host viewpoint, a network prefix is reachable if the host can exchange packets with one IP address belonging to this prefix. It is important to note that simply receiving packets from sources belonging to a given IP prefix does not guarantee the reachability of this prefix since such packets could have been spoofed [34]. The host should receive a reply to the packets that it sends. Several protocols can be used to elicit such a reply from remote hosts: ICMP with `ping` or possibly `traceroute`, TCP or even application-level protocols such as DNS.

A first possibility to validate the reachability of a prefix is to send `ping` packets to one or more IP addresses belonging to this prefix. Researchers and network operators frequently use `ping` to verify that an IP address is reachable. However, `ping` has two important limitations. First, for security reasons, only a small fraction of the Internet hosts respond to `ping` packets. This means that finding reachable IP addresses inside a prefix can be difficult, in particular for IPv6, even if there are hitlists of *pingable* IPv4 [39] and IPv6 hosts [22]. Second, `ping` is not a secure protocol and spoofed replies are possible [1]. Still, `ping` has the advantage of being simple to use.

A better approach is to leverage secure protocols such as TLS [42]. If a client can establish a TLS session with one IP address belonging to the prefix of interest, then does this necessarily confirm that the prefix is reachable? TLS was designed to allow clients to authenticate connections with a server identified by a domain name. For this, TLS relies on X.509 certificates [14] that securely bind a domain name with the server's public key. In addition, the Subject Alternative Name (SAN) extension of X.509 [14][1] certificates also enables the generation of a TLS certificate to bind public keys to IP addresses. SAN allows trusting a specific host contained in a destination prefix. X.509 certificates can therefore be used to confirm that the prefix is reachable.

With TLS certificates, we have a way to authenticate IP addresses. Now we need to deploy a solution in the network to be able to use path validation messages. A first possibility would be to integrate a TLS server in the routers so that they can respond to TLS messages.

[1]See section 4.2.1.6 of RFC5280.

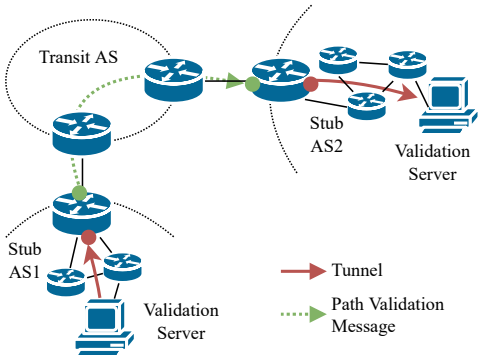

Figure 1: Architecture of the proposed solution.

Unfortunately, adding support for TLS certificates in routers would require modifications to router operating systems to include cryptographic mechanisms to validate TLS certificates. Some deployed router hardware is old and would not simply handle the introduction of TLS. For more recent network hardware, the router itself can run Virtual Machines (VM) to host small applications [16], which can be the ideal solution to include a TLS server in the router to validate paths. Instead, to support a broader range of network devices, we can opt for a decentralized solution as shown in Figure 1. A new type of service can be introduced to validate BGP routes, we call it the *Validation Server* (*VS*). Instead of contacting an existing service inside the AS such as HTTP or DNS servers, the *VS* can be added inside each AS so that routers can contact it to validate the prefix. Each time a router receive a BGP Update, it will contact the *VS* to check if the route is reachable in the dataplane. This *VS* brings several advantages. First, it reduces the load on existing service. For example, an operator would not want to have an additional load on its web server. Second, the service is dedicated only to validation and nothing more. Strong security policies can therefore be applied to this dedicated server. However, this would require additional coordination efforts between the network and the system team, as the routing devices must interact with the infrastructure. It requires more configuration to make the system work. Finally, the *VS* can be used as a cache that can speed up BGP convergence when a router performs a cold start or when it receives a large number of updates. If the route has already been validated by the *VS*, the AS routers can query it to retrieve the information, thus avoiding recontacting the target prefix with a new validation message.

The proposed architecture is flexible, thanks to the introduction of the *VS*, when the AS2 edge router, shown in Figure 1, sends a BGP Update with a prefix, AS1 has several possibilities to validate the path. Either the AS1 edge router asks its *VS* to perform the validation, or the router itself validates the path. On its side, AS2 can also choose which device will respond to AS1's validation request. It can choose to configure its router to respond to the request or to transfer the request to the *VS*. If the router or *VS* of AS1 receives the response from whatever device, the route is considered valid and therefore can be inserted into the routing table.

Using an external machine like the *VS* also brings disadvantages. When the BGP router receives a route to a prefix and asks to the *VS* to contact a destination to the target prefix, it has no routes to the

```
RoutePrefixValidation ::= SEQUENCE {
  pfxValidator IPAddress,
  ipAddrBlocks SEQUENCE (SIZE(1..MAX)) OF
                                    RPVAddressFamily }

RPVAddressFamily ::= SEQUENCE {
  addressFamily OCTET STRING (SIZE (2..3)),
  addresses SEQUENCE (SIZE (1..MAX)) OF RPVIPAddress }

RPVIPAddress ::= SEQUENCE {
  address IPAddress,
  maxLength INTEGER OPTIONAL }

IPAddress ::= BIT STRING
```

Listing 1: RPKI object related to the prefix validation.

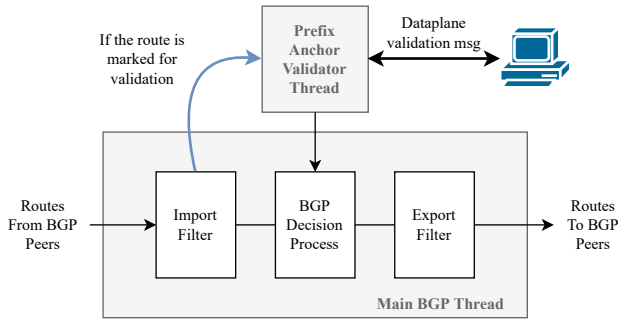

Figure 2: Modified Architecture of BGP to include the Prefix Anchor Validator.

corresponding destination. As the router waits for the validation, it does not install the route to its routing table. Hence, the *VS* cannot contact at all the destination. Instead, the *VS* can tunnel the request to the AS1's egress router as shown in Figure 1. The AS1 edge router is responsible for decapsulating the request and forwarding it to the target *VS* from the interface where it received the route.

Now that we have TLS to authenticate prefixes, we need to modify BGP to ask it to verify the received paths before integrating them into the routing table. To have a deployable solution, it is necessary to tell BGP the destination to be contacted by prefix in a secure way. Adding a new BGP community or a new BGP attribute is not secure because the messages can be intercepted. RPKI solves this problem. RPKI has been deployed to authenticate some Internet resources such as Route Origin Authorizations [31] (ROAs). As of this writing, ROAs are the only RPKI objects that are widely deployed. Other works propose adding new objects into RPKI to enhance routing security [3, 47, 48]. We can imagine using RPKI by proposing a new RPKI object, the Route Prefix Validation (*RPV*) as described in Listing 1. The syntax used in the definition of the *RPV* has been adapted from the definition of ROAs object [31]. In a nutshell, this extension of X.509 certificate contains a sequence of IP prefixes and an IP address that represents the *Validation Server* to contact in order to validate the IP prefixes. Just like ROAs, the *RPV* object will be pushed to the global infrastructure so that all RPKI validators can use them. Since the RPKI object is cryptographically signed, the IP address of the *VS* included in the certificate can be trusted. It is important to stress that the IP address contained in the certificate must be valid for the IP prefix. This means that the IP address must be included in the IP prefix. This prevents anyone from validating the path of other ASes.

When the BGP router receives a route, just like the ROAs, it can decide whether or not to contact the RPKI validator to obtain the IP address of the *VS* to contact to validate the BGP route. Depending on the configuration chosen by the network operator, the router or the *VS* will make the request to check if the IP prefix is accessible in the dataplane. With RPKI and the establishment of a secure communication with the *VS*, the infrastructure provides strong guarantees to find and contact the remote *VS*. It should be noted that the RPKI should not be used to prove the identity of the *VS* [5]. This concern is addressed by the TLS stack with X.509 certificates, as is the case for web server authentication. The *Validation Server* will contain another X.509 certificate that will only be provided for authentication when queried to validate the paths. The certificates of the *Validation Servers* will be signed by a certification authority which could be one of the five Regional Internet Registries (RIRs) and the IANA as the root certificate authority.

2.1 A First Prototype

We developed a first prototype in the BGP module of FRRouting v8.2.2 [21] with a *Prefix Anchor Validator* (*PAV*). Our modification requires ~1.1k lines of C code. The remaining of this section explains the architecture of the *Prefix Anchor Validator* we designed to validate the routes in the data plane.

Figure 2 shows the general architecture of our solution. The traditional BGP workflow remains unchanged as the routes received from BGP neighbors pass through the import filters first. If the import filters notice that a route needs to be validated, it will be passed to another thread that will perform the validation. As FRRouting does not parallelize the processing of BGP Updates, we could not simply perform the validation inside the BGP thread that processes a BGP Update. Doing so would have dramatically slowed down the processing of BGP routes since a route that needed to be validated would have blocked the subsequent BGP routes in the same session until the completion of the validation. This thread can be modified later to add the communication with an external *VS*. For simplicity, and for a first design, we deploy our solution on the router. Once the prefix is validated or not, the validator asks the main FRRouting thread to restart the BGP decision process in order to add or remove the route processed by the validator.

Our *Prefix Anchor Validator* implementation is flexible. The current implementation offers two types of path validation but others can easily be added in the future. The first uses a simple `ping` method. The router sends an ICMP packet and expects to receive the response within a given time period. If no response is received, the router repeats this process until the limit of retries is reached. Although using a `ping` to validate a route is not secure, we still decided to add this validation method to our *PAV* to simplify the evaluation of our proof of concept. The second type of validation is the use of a TLS server. To allow the router administrator to choose the validation method, we modified the FRRouting CLI to allow the setting of the entire *Prefix Anchor Validator*. Namely, the CLI allows the user to change the timeout before considering that no response has been received or the number of prefix validation attempts.

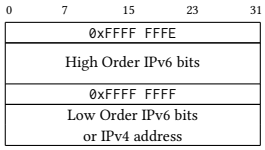

Figure 3: Structure of the Large-Community used for Path Validation.

```
route-map path_validation permit 10
 match path-validation notrequested
!
route-map path_validation permit 20
 match path-validation pending
 set community additive no-export
!
route-map path_validation permit 30
 match path-validation valid
 set community additive 65021:6
!
route-map path_validation deny 40
```

Listing 2: Example of using BGP Path Validation with the FRRouting CLI.

Our implementation keeps a route that has not yet been validated in the routing table with a `NO EXPORT` [7] community. This well-known community indicates that the router cannot propagate it to other neighboring ASes. When the validation process concludes, we either remove the route if it could not be validated (rejection) or remove the `NO EXPORT` community and trigger the BGP decision process (acceptation).

When a route is received, it must be ensured that the traffic can reach the target prefix. This is done by choosing an address that is within the prefix announced by the BGP route. In order to avoid choosing a random destination, our first prototype asks each AS to add a large-community [25] that contains the destination to contact in the prefix. The next prototype will be based on RPKI certificates. But for now, we enable this information by using large-communities as shown in Figure 3. Since a large-community value can only contain 12 bytes, two namespace identifiers had to be reserved to represent an IPv6 address. If path validation is enabled on the router, the network operator can choose to install an import or export filter to check whether a route should be verified with BGP path validation. If the filter matches the large-community, the router triggers a parallel thread that will validate the route without disrupting the initial BGP workflow.

We have modified the FRRouting CLI to integrate the path validation to work with route-maps [11]. Listing 2 shows an example of import filter using the route-maps. In this example, the router will accept routes that are path validated (`valid`), those which do not contain the large-community values (`notrequested`) and those that are in the process of validation (`pending`). If the route cannot be validated, it is withdrawn (`path_validation deny`). However, completely removing the route from the routing table is a difficult decision to take. In fact, it may happen that the route can be reached after a certain time. To avoid this, the operator can always import it into the routing table with the lowest possible local-pref to allow the router to use another valid path instead.

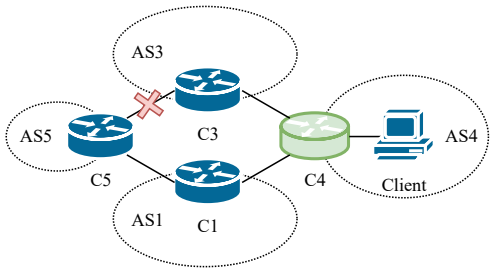

Figure 4: Simple network used for evaluations.

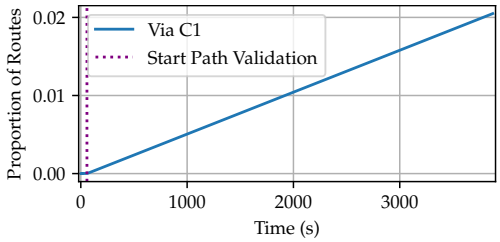

Figure 5: Proportion of the destinations passing via C1.

To show the feasibility of this approach, we use a simple network to validate BGP advertisements made by a BGP neighbor (C5) as shown in Figure 4. The C5 router sends a routing table of 873k routes from a RIPE RIS snapshot (June 3, 2021, at 4:15 PM). We modified this routing table to mark 2% of the routes (18k) with a large-community to instruct the C4 router to validate the routes only for those prefixes. All routers except C4 run BIRD v2.0.9 [15]. The C4 router is running our modified version of FRRouting v8.2.2 [21] to enable BGP to check the reachability of a route using the dataplane. The C4 router is running an Intel® Xeon® X3440 @2.53GHz with 16 GB of RAM, Linux kernel v5.15.29 and Debian 11.

We have configured C4 using local-pref so that all routes advertised by C3 are preferred to those advertised by C1.

To emulate failures in the dataplane when C4 validates routes, we decided to install IP filters for the 18k prefixes on the C3-C5 link. These failures are used to create invalid paths in the dataplane. This way, C4 will detect that the marked route passing through C3 should not be used. The C1-C5 link does not contain any filter. Thus, for all the routes that need to be validated, C4 will decide to go through C1 since the other route does not let path validation messages through. Figure 5 shows the proportion of routes that pass through C1. At the beginning, no route goes through C1 because C4 put a higher local-pref for routes advertised by C3. Then, as the validation is performed, the number of routes passing through C1 increases. Eventually, all the routes that have been marked for path validation pass through C1.

We observe that the routes are validated linearly. Since our prototype validates one path at a time and thus only `ping` one route as it goes, validation takes time. Nevertheless, it is possible to improve the speed of path validation by pinging faster. Tools such as ZMap [18] have shown that it is possible to `ping` the entire Internet very quickly.

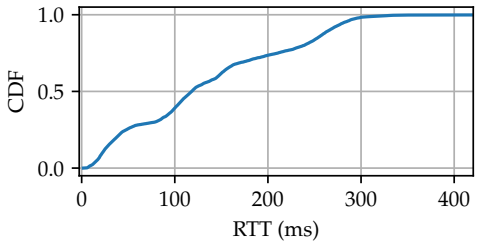

Figure 6: RTT of 7k IPv4 destinations evenly distributed over the Internet.

Even with a faster option to validate paths, the validator will still be limited by the time to contact a destination in the dataplane. Figure 6 shows the Round-Trip Time (RTT) of 7k destinations scattered across the Internet. The reachable destinations were retrieved from the CAIDA dataset [6]. This figure reflects the time BGP would take to validate a route with a validator over the Internet. If the destination to contact is close, the validation will take less time. On the contrary, if the destination is far from the validator, it will take more time to validate the prefix. However, it should be noted that more than 50% of destinations in the Internet have an RTT of more than 120 ms, which is high to validate a BGP route. From these numbers, it seems clear that the time to validate a BGP route is increasing. However, this is the price to pay to be sure that the prefix is reachable in the dataplane.

As paths are validated in the dataplane, routing traffic increases. Our early experiments show that establishing a full TLS connection to transmit a 56-byte payload takes about 3.2 KB of traffic. To make a comparison, we computed the average BGP Update length that carry one IPv4 prefix. The BGP Update message is generated from the same RIPE RIS snapshot used in this work. In average, each BGP Update takes 185 bytes on the wire. There is clearly a significant increase in traffic generated by the router interface performing the path validation. This will have an impact at cold boot when routers need to learn all routes from their neighbors. However, after the initial convergence, updates may occur, but at a slower rate. Recent reports [23] show that routers receives an average of 0.5 prefixes update per second. This additional traffic due to path validation is therefore negligible compared to the number of daily prefix updates.

3 DISCUSSION

Throughout this paper, we propose to augment the routing decision with information from the dataplane. We argue that the linking of both control plane and dataplane can lead to a more accurate and finer-grained routing. In this section, we first discuss the implication of using active probing to check the reachability of a route. Then, we discuss the potential use of the data collected from the dataplane to implement new routing decisions.

Beacons and prefix tests. Some prefixes on the Internet are advertised for analysis and testing purposes only [37]. Performing path validation on these routes is therefore meaningless since no services are expected to operate in these special prefixes. The path validation solution should therefore only be executed on paths for which it is explicitly asked to validate the path. That is, when sending a validation message makes sense. To implement this idea, an RPKI object that binds the prefix to a TLS server must be explicitly created. This object indicates that the validation must be performed.

Detecting zombie routes and censorship. A zombie route occurs when a route is still considered reachable by some routers but removed by others. This situation can occur due to a software bug where the BGP implementation fails to process a BGP Withdraw [20]. This situation leads to network degradation because part of the network will continue to use an invalid path, which can cause traffic blackholing. Our path validation method could periodically scan all destinations in the routing table to ensure that their routes are still reachable. If the validator detects that a route is no longer reachable, it will trigger an update by notifying its neighbors. Similarly, some ASes practice censorship on the Internet. A famous example occurred in 2008, when Pakistan Telecom announced the YouTube prefix throughout the world [24] to censor the platform in the country. Recently, other countries used BGP to censor part of the Internet [45]. Since this form of censorship is not detected in the control plane, adding a path validator in BGP allows routers to check if the prefix is reachable and thus prevents routers from propagating hijacked prefixes to the world.

Reconcile ASes that want to check their respective prefixes at the same time. Let's imagine two Stub networks, AS1 and AS2. These two ASes use path validation to ensure that there is a valid path between them. If both ASes receive each other's announcement at the same time, it will not be possible to perform path validation, since the routing table of neither ASes have an entry for the other AS. Then they have no way to return confirmation that the path is valid. When a peering link between a stub and its provider is formed, the provider will usually use an address belonging to the stub's prefix. Since the stub prefix is a subset of the provider's addressing space, the address chosen by the provider's router is reachable from both AS1 and AS2. Using the address of the provider's router, AS1 and AS2 can use the path validation again. Back to Figure 1, to perform path validation, AS1 and AS2 will need to use the addresses associated to the green interfaces of their edge routers.

Application in transit networks. Doing Path validation for transit network has to be done with great care. If a transit AS considers the prefix not reachable because of a TLS certificate missconfiguration, no traffic will transit through the AS for the prefix. This is an important decision for a large transit AS to make. Currently, with origin validation (e.g., with RPKI), ASes have the choice to decide whether or not to reject a prefix that is invalidated by the RPKI validator. Path validation can replicate the same behavior as implemented for route origin validation. If validation detects that a path to a prefix is invalid, it should not necessarily be rejected. The operator should have the option of inserting the path in the routing table in the hope that another valid path will replace it as soon as possible.

Augmenting the BGP Decision Process. The BGP decision process uses only the information in the control plane to choose the best route to a destination. However, relying solely on this information leads to a suboptimal decision about the path used. For example, taking into account dataplane metrics such as latency can

lead to a more optimal decision in terms of latency for 77% of the prefixes [38].

We could imagine integrating other metrics of the dataplane such as the bandwidth rate and thus form several types of targets to be reached. Depending on the metric we want to use, we could propose, thanks to BGP-EPE [19], a way to route traffic according to the desired metric.

Using dataplane data to influence the control plane. We saw previously how information from the dataplane can be used to select BGP routes. We could do the reverse, when an event in the dataplane occurs, the control plane is notified to solve the problem. For example, if we notice that the latency of the chosen route starts to exceed a certain threshold, we retrigger a new route selection by invoking the BGP decision process.

Using other secure transport layer. Throughout this paper, we propose to use TLS to validate BGP paths. To perform the validation, the router exchanges only a few packets to check if the *Validation Server* can respond to the query. However, TLS requires a reliable connection to be used. Maintaining a state to send only a few TLS data requires many resources that routers may not have. Instead, other secure protocols such as DTLS [43] or QUIC [28] can be used. These protocols are designed to secure data sent over an unreliable datagram connection and are therefore a good alternative to a full TLS stack for our use case. Since our prototype is quite flexible, changing the security layer is not a problem since we do not depend on any specific features of TLS. We use TLS only as a way to secure the data and authenticate the remote device.

Path Validation deployment. To deploy it, there are two steps to consider. The first is to support validation in BGP through the new RPKI object described in Section 2. The second is to set up the *VS* (or its equivalent on the router) that allows validation requests to be answered. These steps can be performed in any order and incrementally without disrupting the validation operation.

In addition, similar to the current ROA deployment, all ASes can choose to enable Path Validation and integrate with other participating ASes. The operator can decide whether or not to accept routes from ASes that have not yet deployed an RPKI object related to path validation.

4 RELATED WORK

`LIFEGUARD` [30] actively monitors the network with `traceroutes` to PlanetLab hosts [9] to catch failures. It is able to detect the exact location of the failure in the network. When noticed, `LIFEGUARD` sends a BGP advertisement with a modified AS-Path to invalidate the failed path and force all routers in the network to select another reachable path in the dataplane. When a failure is detected in AS1, BGP `LIFEGUARD` routers re-announce their prefixes by appending AS1 in their AS-Path. When AS1 receives the poisoned AS-Path, the BGP loop detection algorithm discards the path and thus withdraws the route, forcing all downstream routers to change their routes. With our BGP Path Validation, the routers reconfigure themselves without poisoning the AS-Path information. This allows to be compliant with recent RFC drafts that try to verify the AS adjacencies [48] or the complete AS-Path [32].

BGPsec [27], standardized by the IETF, verifies the complete AS-Path. Before announcing a BGP Update, all routers sign their messages. The AS-Path attribute is replaced by a new, more secure BGP attribute called "BGPsec Path". Upon receiving the announce, any BGP router is guaranteed that the dataplane will follow the control plane. While it verifies the integrity of the AS-Path, it cannot prevent a network failure inside an AS. Furthermore, deploying BGPsec is non-trivial and requires that all ASes participate in its deployment [35]. Other studies have shown that it is still possible to create forwarding loops or perform wormhole attacks with BGPsec enabled [33]. However, the BGP Path Validation can be used in addition to BGPsec to make sure that the dataplane will be able to forward packets on the path announced by BGP.

BGP Path-End Validation [13] is an alternative to BGPsec. It has shown that by only ensuring that the last AS hop is valid, a comparable level of security as BGPsec can be achieved even when BGP Path-End is not fully deployed. There is no need to replace routers to deploy BGP Path-End since the router does not need to compute cryptographic signatures in BGP messages. They only need a new RPKI object to work. Just like BGPsec, BGP Path-End validation only helps the control plane integrity but does not check the current state of the dataplane. These verification techniques do not guarantee that the path taken by the traffic will not result in losses or failures.

Instead of solving the security problems of BGP, `SCION` [50] proposes a new Internet architecture to overcome the current limitations of BGP. `SCION` uses a cryptographic authentication system to avoid configuration errors, route leaks or prefix hijacking, thus ensuring connectivity in the dataplane.

Other tools to scan the Internet have been developed to examine the general state of the network. Thunderping [40] or Trinocular [39] detect local and global problems respectively. By actively scanning the reachable destinations, these tools are able to accurately detect problems in the network. To improve the implementation of BGP path validation, the techniques developed in these tools could be used.

5 CONCLUSION

We introduced BGP Path Validation, a new validation system to check if a route is reachable in the dataplane. Before adding a route to the routing table, it will first contact a specialized service responsible for responding to the router's request. With this method, routing errors due to misconfiguration can be reduced because the router is now sure that the traffic can reach the destination AS and will not be discarded on the way. We have demonstrated the feasibility of this approach by proposing a working prototype implemented in the FRRouting suite. Finally, we discussed the possibility of bringing dataplane information into the routing world. This would provide a more accurate view of the network and therefore allow a more relevant and optimal path to be chosen. Similarly, the control plane can be notified by an event occurring in the dataplane to adapt the routing decision.

SOFTWARE ARTIFACTS

Our modified version of FRRouting, which includes our first BGP Path Validation prototype, is available in a GitHub repository at https://github.com/twirtgen/frr/tree/stable/8.2-dataplane.

ETHICAL CONSIDERATIONS

This work does not raise any ethical issues.

ACKNOWLEDGMENTS

This work has been supported by the French Community of Belgium through the funding of a FRIA (Fund for Research training in Industry and Agriculture) grant.

REFERENCES

[1] Lance Alt, Robert Beverly, and Alberto Dainotti. 2014. Uncovering network tarpits with degreaser. In *Proceedings of the 30th Annual Computer Security Applications Conference*. 156–165.

[2] Alexander Azimov, Eugene Bogomazov, Randy Bush, Keyur Patel, and Job Snijders. 2021. *Verification of AS_PATH Using the Resource Certificate Public Key Infrastructure and Autonomous System Provider Authorization.* Internet-Draft draft-ietf-sidrops-aspa-verification-08. Internet Engineering Task Force. https://datatracker.ietf.org/doc/html/draft-ietf-sidrops-aspa-verification-08 Work in Progress.

[3] Alexander Azimov, Eugene Uskov, Randy Bush, Keyur Patel, Job Snijders, and Russ Housley. 2022. *A Profile for Autonomous System Provider Authorization.* Internet-Draft draft-ietf-sidrops-aspa-profile-07. Internet Engineering Task Force. https://datatracker.ietf.org/doc/draft-ietf-sidrops-aspa-profile/07/ Work in Progress.

[4] R. Bush and R. Austein. 2013. The Resource Public Key Infrastructure (RPKI) to Router Protocol. RFC 6810 (Proposed Standard). https://doi.org/10.17487/RFC6810

[5] Randy Bush and Russ Housley. 2022. The 'I' in RPKI Does Not Stand for Identity. RFC 9255. https://doi.org/10.17487/RFC9255

[6] CAIDA. 2022. *The CAIDA Macroscopic Internet Topology Data Kit - 2021-03.* https://www.caida.org/catalog/datasets/internet-topology-data-kit

[7] R. Chandra, P. Traina, and T. Li. 1996. BGP Communities Attribute. RFC 1997 (Proposed Standard). https://doi.org/10.17487/RFC1997 Updated by RFCs 7606, 8642.

[8] Shinyoung Cho, Romain Fontugne, Kenjiro Cho, Alberto Dainotti, and Phillipa Gill. 2019. BGP hijacking classification. In *2019 Network Traffic Measurement and Analysis Conference (TMA)*. IEEE, 25–32.

[9] Brent Chun, David Culler, Timothy Roscoe, Andy Bavier, Larry Peterson, Mike Wawrzoniak, and Mic Bowman. 2003. PlanetLab: An Overlay Testbed for Broad-Coverage Services. *SIGCOMM Comput. Commun. Rev.* 33, 3 (jul 2003), 3–12. https://doi.org/10.1145/956993.956995

[10] Taejoong Chung, Emile Aben, Tim Bruijnzeels, Balakrishnan Chandrasekaran, David Choffnes, Dave Levin, Bruce M Maggs, Alan Mislove, Roland van Rijswijk-Deij, John Rula, et al. 2019. RPKI is coming of age: A longitudinal study of RPKI deployment and invalid route origins. In *Proceedings of the Internet Measurement Conference*. 406–419.

[11] Inc Cisco Systems. 2015. *Route-Maps for IP Routing Protocol Redistribution Configuration.* https://www.cisco.com/c/en/us/support/docs/ip/border-gateway-protocol-bgp/49111-route-map-bestp.html

[12] Avichai Cohen, Yossi Gilad, Amir Herzberg, and Michael Schapira. 2016. Jumpstarting BGP security with path-end validation. In *Proceedings of the 2016 ACM SIGCOMM Conference*. 342–355.

[13] Avichai Cohen, Yossi Gilad, Amir Herzberg, and Michael Schapira. 2016. Jumpstarting BGP Security with Path-End Validation. In *Proceedings of the 2016 ACM SIGCOMM Conference* (Florianopolis, Brazil) *(SIGCOMM '16)*. Association for Computing Machinery, New York, NY, USA, 342–355. https://doi.org/10.1145/2934872.2934883

[14] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk. 2008. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC 5280 (Proposed Standard). https://doi.org/10.17487/RFC5280 Updated by RFCs 6818, 8398, 8399.

[15] CZ.NIC Labs. 2020. *The BIRD Internet Routing Daemon.* https://bird.network.cz

[16] Cisco DevNet. 2020. *Application Hosting.* https://developer.cisco.com/docs/ios-xe/#!application-hosting-quick-start-guide/application-hosting-options

[17] Xenofontas Dimitropoulos, Dmitri Krioukov, George Riley, and KC Claffy. 2005. Classifying the types of autonomous systems in the internet. In *ACM SIGCOMM*. Cooperative Association for Internet Data Analysis (CAIDA).

[18] Zakir Durumeric, Eric Wustrow, and J. Alex Halderman. 2013. ZMap: Fast Internet-Wide Scanning and Its Security Applications. In *Proceedings of the 22nd USENIX Conference on Security* (Washington, D.C.) *(SEC'13)*. USENIX Association, USA, 605–620.

[19] Clarence Filsfils, Stefano Previdi, Gaurav Dawra, Ebben Aries, and Dmitry Afanasiev. 2021. Segment Routing Centralized BGP Egress Peer Engineering. RFC 9087. https://doi.org/10.17487/RFC9087

[20] Romain Fontugne, Esteban Bautista, Colin Petrie, Yutaro Nomura, Patrice Abry, Paulo Goncalves, Kensuke Fukuda, and Emile Aben. 2019. BGP Zombies: An Analysis of Beacons Stuck Routes. In *Passive and Active Measurement*, David Choffnes and Marinho Barcellos (Eds.). Springer International Publishing, Cham, 197–209.

[21] FRRouting. 2017. *FRRouting Project.* https://frrouting.org/

[22] Oliver Gasser, Quirin Scheitle, Pawel Foremski, Qasim Lone, Maciej Korczyński, Stephen D Strowes, Luuk Hendriks, and Georg Carle. 2018. Clusters in the expanse: Understanding and unbiasing IPv6 hitlists. In *Proceedings of the Internet Measurement Conference 2018*. 364–378.

[23] Geoff Huston. 2022. *The BGP Instability Report.* https://bgpupdates.potaroo.net/instability/bgpupd.html

[24] Sharon Goldberg. 2014. Why is It Taking so Long to Secure Internet Routing? *Commun. ACM* 57, 10 (sep 2014), 56–63. https://doi.org/10.1145/2659899

[25] J. Heitz (Ed.), J. Snijders (Ed.), K. Patel, I. Bagdonas, and N. Hilliard. 2017. BGP Large Communities Attribute. RFC 8092 (Proposed Standard). https://doi.org/10.17487/RFC8092

[26] Geoff Huston. 2013. *The 32-bit AS Number Report.* https://www.potaroo.net/tools/asn32/

[27] Geoff Huston and Randy Bush. 2011. Securing bgp with bgpsec. In *The Internet Protocol Forum*, Vol. 14.

[28] Jana Iyengar and Martin Thomson. 2021. QUIC: A UDP-Based Multiplexed and Secure Transport. RFC 9000. https://doi.org/10.17487/RFC9000

[29] Paula Jabloner. 2015. *The two-napkin protocol.*

[30] Ethan Katz-Bassett, Colin Scott, David R. Choffnes, Ítalo Cunha, Vytautas Valancius, Nick Feamster, Harsha V. Madhyastha, Thomas Anderson, and Arvind Krishnamurthy. 2012. LIFEGUARD: Practical Repair of Persistent Route Failures. *SIGCOMM Comput. Commun. Rev.* 42, 4 (aug 2012), 395–406. https://doi.org/10.1145/2377677.2377756

[31] M. Lepinski, S. Kent, and D. Kong. 2012. A Profile for Route Origin Authorizations (ROAs). RFC 6482 (Proposed Standard). https://doi.org/10.17487/RFC6482

[32] M. Lepinski (Ed.) and K. Sriram (Ed.). 2017. BGPsec Protocol Specification. RFC 8205 (Proposed Standard). https://doi.org/10.17487/RFC8205

[33] Qi Li, Yih-Chun Hu, and Xinwen Zhang. 2014. Even rockets cannot make pigs fly sustainably: Can BGP be secured with BGPsec. In *Workshop SENT'14, 23 February 2014, San Diego, USA, Copyright 2014 Internet Society: Proceedings*. Internet Society.

[34] Matthew Luckie, Robert Beverly, Ryan Koga, Ken Keys, Joshua A Kroll, and K Claffy. 2019. Network hygiene, incentives, and regulation: deployment of source address validation in the Internet. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. 465–480.

[35] Robert Lychev, Sharon Goldberg, and Michael Schapira. 2013. BGP Security in Partial Deployment: Is the Juice Worth the Squeeze? *SIGCOMM Comput. Commun. Rev.* 43, 4 (aug 2013), 171–182. https://doi.org/10.1145/2534169.2486010

[36] Ratul Mahajan, David Wetherall, and Tom Anderson. 2002. Understanding BGP Misconfiguration. *SIGCOMM Comput. Commun. Rev.* 32, 4 (aug 2002), 3–16. https://doi.org/10.1145/964725.633027

[37] Z. Morley Mao, Randy Bush, Timothy G. Griffin, and Matthew Roughan. 2003. BGP Beacons. In *Proceedings of the 3rd ACM SIGCOMM Conference on Internet Measurement* (Miami Beach, FL, USA) *(IMC '03)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/948205.948207

[38] Ryo Nakamura, Kazuki Shimizu, Teppei Kamata, and Cristel Pelsser. 2022. A First Measurement with BGP Egress Peer Engineering. In *Passive and Active Measurement*, Oliver Hohlfeld, Giovane Moura, and Cristel Pelsser (Eds.). Springer International Publishing, Cham, 199–215.

[39] Lin Quan, John Heidemann, and Yuri Pradkin. 2013. Trinocular: Understanding Internet Reliability through Adaptive Probing. *SIGCOMM Comput. Commun. Rev.* 43, 4 (aug 2013), 255–266. https://doi.org/10.1145/2534169.2486017

[40] Lin Quan, John Heidemann, and Yuri Pradkin. 2013. Trinocular: Understanding Internet Reliability through Adaptive Probing. In *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM* (Hong Kong, China) *(SIGCOMM '13)*. Association for Computing Machinery, New York, NY, USA, 255–266. https://doi.org/10.1145/2486001.2486017

[41] Y. Rekhter (Ed.), T. Li (Ed.), and S. Hares (Ed.). 2006. A Border Gateway Protocol 4 (BGP-4). RFC 4271 (Draft Standard). https://doi.org/10.17487/RFC4271

[42] E. Rescorla. 2018. The Transport Layer Security (TLS) Protocol Version 1.3. RFC 8446 (Proposed Standard). https://doi.org/10.17487/RFC8446

[43] E. Rescorla and N. Modadugu. 2012. Datagram Transport Layer Security Version 1.2. RFC 6347 (Proposed Standard). https://doi.org/10.17487/RFC6347 Updated by RFCs 7507, 7905.

[44] Andreas Reuter, Randy Bush, Italo Cunha, Ethan Katz-Bassett, Thomas C Schmidt, and Matthias Wählisch. 2018. Towards a rigorous methodology for measuring adoption of RPKI route validation and filtering. *ACM SIGCOMM Computer Communication Review* 48, 1 (2018), 19–27.

[45] Loqman Salamatian, Frédérick Douzet, Kavé Salamatian, and Kévin Limonier. 2021. The geopolitics behind the routes data travel: a case study of Iran. *Journal of Cybersecurity* 7, 1 (08 2021). https://doi.org/

10.1093/cybsec/tyab018 arXiv:https://academic.oup.com/cybersecurity/article-pdf/7/1/tyab018/39765655/tyab018.pdf tyab018.

[46] Pavlos Sermpezis, Vasileios Kotronis, Alberto Dainotti, and Xenofontas Dimitropoulos. 2018. A survey among network operators on BGP prefix hijacking. *ACM SIGCOMM Computer Communication Review* 48, 1 (2018), 64–69.

[47] Job Snijders, Mikael Abrahamsson, and Ben Maddison. 2022. *Resource Public Key Infrastructure (RPKI) object profile for Discard Origin Authorizations (DOA).* Internet-Draft draft-spaghetti-sidrops-rpki-doa-00. Internet Engineering Task Force. https://datatracker.ietf.org/doc/draft-spaghetti-sidrops-rpki-doa/00/ Work in Progress.

[48] Job Snijders, stucchi lists@glevia.com, and Melchior Aelmans. 2020. *RPKI Autonomous Systems Cones: A Profile To Define Sets of Autonomous Systems Numbers To Facilitate BGP Filtering.* Internet-Draft draft-ietf-grow-rpki-as-cones-02. Internet Engineering Task Force. https://datatracker.ietf.org/doc/html/draft-ietf-grow-rpki-as-cones-02 Work in Progress.

[49] Matthias Wählisch, Robert Schmidt, Thomas C Schmidt, Olaf Maennel, Steve Uhlig, and Gareth Tyson. 2015. RiPKI: The tragic story of RPKI deployment in the Web ecosystem. In *Proceedings of the 14th ACM Workshop on Hot Topics in Networks*. 1–7.

[50] Xin Zhang, Hsu-Chun Hsiao, Geoffrey Hasker, Haowen Chan, Adrian Perrig, and David G. Andersen. 2011. SCION: Scalability, Control, and Isolation on Next-Generation Networks. In *2011 IEEE Symposium on Security and Privacy*. 212–227. https://doi.org/10.1109/SP.2011.45