

Securing Interdomain Routing with Reachability Validation Graphs

Bora Akyol and Andrew Wright
Critical Infrastructure Assurance Group, Cisco Systems
bora@cisco.com, andwri@cis.com

Abstract—This paper presents a new approach to validating origin and path information carried in the BGP-4 UPDATE messages that govern interdomain routing in the Internet. The goal is to provide effective defenses against both invalid origin and incorrect path injection attacks, while remaining scalable, incrementally deployable, and operationally simple. The approach centers around constructing a Reachability Validation Graph (RVG), against which origin and path information in route advertisements can be checked for validity. The paper proposes three alternative mechanisms for obtaining authenticated information with which to build the RVG.

I. INTRODUCTION

Interdomain routing involves the exchange of origin and path information in route advertisements between the *autonomous systems* (ASes) that form the Internet. Border Gateway Protocol (BGP) version 4 is the primary routing protocol that is used to perform this task [1], [2]. However, there are currently no Internet standards, either part of BGP or otherwise, for authenticating this reachability information.

We present a new approach to validating the origin and path information carried in BGP UPDATE messages. Our goal is to provide effective defenses against both invalid origin and incorrect path injection attacks, while remaining scalable, incrementally deployable, and operationally simple. Our approach centers around constructing a *Reachability Validation Graph* (RVG), against which origin and path information can be checked for validity. An RVG need not be constructed on every BGP-speaking border router. Rather, an autonomous system may choose to compute RVGs on one or several *Reachability Validation Servers* (RVS) and periodically download the information to its border routers. We propose three alternative mechanisms for obtaining authenticated information with which to build RVGs.

In the next section, we discuss the problem of authenticating origin and path information in BGP, outline requirements for an acceptable solution, and review previous work. Section III describes Reachability Validation Graphs and how they are used to authenticate origin and path information. Sections IV, V, and VI describe our three approaches to computing RVGs, Section VII discusses the merits of the three, and Section VIII discusses future work.

II. INTERDOMAIN ROUTING SECURITY

The purpose of interdomain routing protocols in the Internet is to distribute reachability information between autonomous systems in a scalable and reliable manner. BGP (version 4) is widely used as the routing protocol to accomplish this task. This section reviews BGP route advertisements, describes some attacks against BGP, outlines the defenses against these attacks, and reviews existing work on securing interdomain routing.

A. BGP Route Advertisements

BGP route advertisements are carried in BGP UPDATE messages and express reachability information that originates from a particular autonomous system. For our purposes, the primary pieces of information present in an UPDATE message are:

- **Address Range:** A range of IP addresses that are advertised by the autonomous system. In BGP terminology the address range is commonly referred to as the *prefix*. For example, 10.16/16 is a prefix representing the addresses from 10.16.0.0 through 10.16.255.255.
- **Autonomous System Path List:** The AS path list includes the list of autonomous systems by which the address range is reachable. The AS path list is used primarily to prevent routing loops.

As an example, a typical route advertisement might announce that address range 10.16/16 can be reached via the AS path list (AS6, AS5, AS2, AS1). The AS path list is ordered with the originating autonomous system number on the right. As route advertisements are passed from one AS to another, AS numbers are prepended to the path list[1]. Thus AS1 originated the advertisement, and it traveled to the router via AS2, AS5, and finally AS6.

B. Attacks Against Interdomain Routing

In this paper, we focus on attacks against interdomain routing whereby the attacker seeks to disrupt routing for one or more address ranges by injecting incorrect information. The potential goals of the attacker are numerous, including snooping the traffic going to that address range, black holing the traffic, acting as a man-in-the-middle modifying the traffic, etc. [3], [4]. The basic attacks on route advertisements that we are concerned with are:

- 1) Unauthorized Prefix Injection: the attacker advertises one or more prefixes that do not legitimately belong to the attacker. The most likely effect of this attack is to completely disrupt traffic going to that address range.
- 2) AS Path List Forgery: the attacker pretends to be directly connected to an AS that he is not a peer of. A likely effect of this attack is to route traffic through the attacker, facilitating eavesdropping.

An attacker can inject incorrect information into the interdomain routing system in multiple ways. An attacker might be a legitimate autonomous system or a large corporation seeking to spy on its enemies or competitors, or to offer such services through an underground organization. Corrupt employees (insiders) of an organization running an autonomous system might collaborate with an attacker. Bugs in router firmware can allow an attacker direct access to compromised routers.

C. Interdomain Routing Peering

In this section we present a brief review of peering relationships in the Internet. The nature of current peering relationships must be taken into account in any solution to securing interdomain routing.

In the first decade of the Internet, autonomous systems peered with each other in public network access points. A public network access point brought together many autonomous systems in the same facility, where the autonomous systems peered with each other typically using a shared media link like Ethernet. As the Internet grew, network service providers took over the responsibility of

providing Internet connectivity from government-funded organizations such as NSF. The largest of these network service providers that provided backbone service for the Internet chose to peer with each other privately; that is, they would peer with each other mostly over point-to-point high speed links. The differentiating feature of private peering versus peering at a public network access point is that the transit of traffic between the peering autonomous systems is *highly controlled and enforced* according to financial contracts. The network service providers (try to) maintain strict control over the address ranges and paths that they advertise to their peers and routinely apply routing policy including summarizing, AS path list padding, or prefix filtering. While the application of routing policy results in modification and loss of some aspects of reachability information received by each AS, the autonomous systems enjoy control of traffic passing through their network without intervention from a central authority. For further reading on this topic we refer the reader to [5] and [6].

D. Solution Requirements for Securing Interdomain Routing

Solutions for securing interdomain routing must satisfy the following requirements:

- 1) Defense against injection of incorrect originating AS information: Since we expect the deployment of secure interdomain routing to be incremental, prefixes that are injected into interdomain routing by autonomous systems that participate in the security solution must be protected even when a large portion of the Internet does not participate in the secure interdomain routing.
- 2) Defense against incorrect AS path list injection: For incremental deployments, even when only a few autonomous systems participate in the secure interdomain routing, the routing information exchanged by these networks must be protected.
- 3) Incremental Deployability and Protection: The Internet is a very large network. Any solution that requires a *flag day* for switching over all border routers in the Internet to a brand new version of BGP is not deployable. A solution that is incrementally deployable but that does not provide a significant benefit until most of the autonomous systems deploy it is also far from being useful.
- 4) Operational Simplicity: The security solution for interdomain routing must be deployable without major changes to the router operation or configuration of the routers. Moreover, it must not require

- significant hardware upgrades on existing routers.
- 5) Reflect existing business relationships: To gain the acceptance needed for broad deployment, trust must be based on existing financial and business arrangements. In particular, institution of a centralized authority where one does not currently exist is unlikely to be acceptable. If a centralized authority is used, all data in the centralized authority must be *originated and controlled* by the individual autonomous systems.
 - 6) Dynamic: The solution for securing interdomain routing must allow dynamic changes to the ownership of prefixes or peering topology within a time span of five to ten minutes.¹
 - 7) Safely Bootable: The solution must not rely on interdomain routing to secure interdomain routing unless it allows a method to bootstrap the network while security information is being retrieved. Injection of false security information must be minimized during the bootstrapping period.

E. Previous work on securing Interdomain routing

The security issues with interdomain routing have been discussed in multiple texts. In [7], the authors introduce an extension to BGP, referred to as S-BGP. To summarize, S-BGP aims to secure interdomain routing by cryptographically signing BGP Update messages as they travel between autonomous systems, and using IPSEC to secure the integrity of the communication between the border routers. S-BGP also introduces a hierarchical PKI structure for verifying: the ownership of prefixes, and also the identities of routers. Since each BGP Update message is signed, S-BGP requires significant cryptographic capability in router hardware as well as storage for certificates and revocation lists. Additionally, S-BGP does not allow BGP UPDATE packing, a common and critical optimization impacting BGP convergence times. S-BGP has not been deployed in any commercially viable network since the protocol does not meet the requirements of network service providers for deployability [8] and has negative business and performance ramifications.

A proposal known as soBGP [9] seeks to verify the origin and path information in UPDATE messages using cryptographic signatures. It differs from S-BGP in that it requires fewer cryptographic operations, need not rely on a hierarchical PKI, does not sign individual UPDATE

messages, and appears to provide most but not all of the protection that S-BGP provides. Our third approach discussed in Section VI to authenticating the origin and path information used to build RVGs builds on the certificate structure of soBGP.

Goodell et al. [10] propose using Interdomain Route Validation (IRV) servers to exchange and validate both static routing policy such as authorized prefixes and dynamic information such as current advertisements. In our opinion, any system that attempts to make current route advertisements available through a parallel channel to BGP is likely to run into both inconsistency problems and performance problems, as this system must maintain essentially the same information that BGP does. Restricted to static or mostly static information, however, IRV servers have similarities to our proposals. Our work delves more deeply into the mechanics of authenticating origin and path information and of checking advertisements for validity.

An alternative to cryptographic security in interdomain routing is the judicious use of filtering of the prefixes being advertised by routing peers [11]. Filters can either be manually generated or can be downloaded via Internet web sites run by the network service provider community [12]. These filters incorporate information from Internet address registries as well as adding addresses that should never appear in advertisements such as private address spaces and multicast. While the filter-based schemes for protection of BGP information are powerful, they rely on complex filter configuration and are only as reliable as the data that is being used. It is also difficult to check for AS path forgery using filters.

III. SECURING INTERDOMAIN ROUTING WITH REACHABILITY VALIDATION GRAPHS

In this paper, we present a new approach to validating the origin and path information carried in an UPDATE message. We first define a Reachability Validation Graph (RVG), which expresses information about origin and path information that has somehow been authenticated. We then describe how an RVG is used in a router to validate information in an UPDATE message. Finally, we discuss three approaches to obtaining the authenticated origin and path information necessary to build RVGs.

A. Reachability Validation Graphs

An RVG is a directed graph in which nodes are autonomous systems and directed edges represent peering relationships. Nodes that are participating in the

¹Five to ten minutes is approximately the time it takes for a route advertisement to travel through the Internet today (2004).

protocol by supplying trustworthy authenticated information about the prefixes they originate are marked as trusted and labeled with those prefixes. An edge from AS_1 to AS_2 indicates that AS_1 is participating in the protocol, is trusted, and may supply authenticated route advertisements to AS_2 . Untrusted nodes must not have any prefix labels nor outgoing edges. Nodes that supply authentication information that cannot be validated are treated the same as if they provided no authentication information at all, and thus cannot be trusted. We say an RVG is well-formed if all prefix labels are unique; that is, two ASes do not advertise the same prefix.

B. Validating Route Information

Given a route advertisement and a well-formed RVG, we use the RVG to determine whether the prefix in the advertisement is valid and whether the path is plausible.

First, however, a simple sanity check should be performed on the advertisement. If the last (leftmost) entry in the AS Path, which was just added by the transmitting BGP speaker, is not the same as the configured AS of the transmitting router, the advertisement should be rejected.

Now, for a route advertisement announcing path $AS_n \cdots AS_1$ to prefix p :

- (i) if AS_1 exists in the RVG and is labeled with p , the prefix in the advertisement is valid;
- (ii) if AS_1 exists in the RVG and is not labeled with p and AS_1 is trusted, the prefix in the advertisement is invalid;
- (iii) if some other $AS' \neq AS_1$ exists in the RVG and is labeled with p , the prefix in the advertisement is invalid;
- (iv) otherwise the validity of the prefix is undetermined.

For a path $AS_n \cdots AS_1$, the path is implausible if there is any AS_i in the path such that AS_i exists in the RVG, AS_i is trusted, and there is no edge from AS_i to AS_{i+1} . Otherwise the path is plausible. If the prefix in the advertisement is invalid or the path is implausible, the advertisement is not authentic. We leave the decision to discard or flag such an advertisement up to the operator of the router.

Figure 1 illustrates a simple RVG. In this example, AS_1 , AS_3 , and AS_4 are trusted, while AS_2 is not. AS_1 and AS_3 could be ISPs, while AS_4 could be a multihomed customer. To indicate that RVGs can be augmented to carry additional information useful to path validation, we have additionally annotated the link between AS_1 and AS_4 as nontransit, meaning that route advertisements received by AS_1 from AS_4 should not

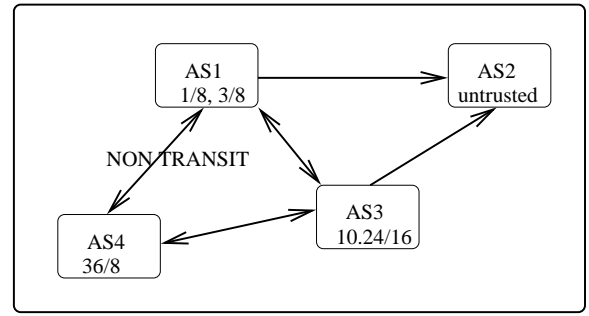


Fig. 1. A Simple internet

be forwarded to other ASes, nor should advertisements received by AS_4 from AS_1 . If AS_3 receives a route advertisement containing a path (AS_2, AS_1, AS_4) , it should use this nontransit information to decide that this path is implausible.

The information expressed in this RVG can be represented in two data structures suitable for answering prefix validity and path plausibility. A *Prefix Ownership Map* answers prefix validity by mapping prefixes to AS numbers:

$$\left\{ \begin{array}{l} 1/8 \mapsto AS_1, \\ 3/8 \mapsto AS_1, \\ 10.24/16 \mapsto AS_3, \\ 36/8 \mapsto AS_4 \end{array} \right\}$$

An *AS Connectivity Map* answers path plausibility by mapping AS numbers to their neighbours:

$$\left\{ \begin{array}{l} AS_1 \mapsto \{AS_2^{nt}, AS_3, AS_4\}, \\ AS_3 \mapsto \{AS_1, AS_2, AS_4\}, \\ AS_4 \mapsto \{AS_1^{nt}, AS_3\} \end{array} \right\}$$

where the superscript ^{nt} indicates nontransit. While the worst case size of the AS Connectivity Map is N^2 , in practice the AS connectivity is sparse even at the core of the network when compared the total number of autonomous systems.

When not all autonomous systems are participating in the protocol, the RVG may not be a connected graph. The path from the AS checking an advertisement back to the AS that originated it may pass through several islands of ASes participating in the protocol, separated by spans of ASes not participating. In this case, provided the AS originating the advertisement is participating, it is still possible to check the validity of the prefix, and thereby ensure that the AS advertising it is authorized to do so. However, plausibility of the path can be only partially verified. Since the path passes through ASes that are not participating in the protocol, there is no guarantee that

an adversary will not be able to redirect packets as they pass through those ASes.

C. Computing Reachability Validation Graphs

While the routers of an AS could directly compute RVGs, a more efficient and manageable approach is to compute RVGs on a (redundant) *reachability validation server* (RVS) managed by the AS. An RVS can be either a router or a network management computer that has access to the information required to calculate the RVG.

The RVG or an update to the RVG is downloaded from the RVS to all border routers in the autonomous system after every rebuild of the RVG. The RVG should be rebuilt periodically as the RVS receives new information. The communication from the local registry to the border routers should be secured.

D. Validating UPDATE Messages

After receiving the RVG from the RVS, the router examines its route information base (RIB) and purges or flags prefixes that do not match the plausibility criteria according to the algorithm given in section II. The decision to purge or flag is left to the operator of the router. Once the pre-existing RIB is checked, then there are two alternatives for validating UPDATE messages that are received from the router's peers: Real-time or Off-line validation. In real-time validation, the router checks the RVG for every prefix that is received via a reachability information update. In off-line validation, the router periodically runs a process to check the RIB against the reachability validation database and purge/flag incorrect prefixes.

We prefer the off-line validation approach due to two reasons: convergence time of routing and bootstrapping the network. The bootstrapping issues are discussed in the next section. We expect attacks against interdomain routing to be rare but well-planned. Due to this expectation, running real-time validation will put a high cost on securing interdomain routing for a rare event. Periodically (for example every 3 minutes) scanning the RIB and purging implausible information from the RIB provides nearly equivalent operational security while dramatically improving the cost of providing the security.

E. Bootstrapping

When a network is coming up, there will be a window in which the RVS and the border routers in the AS may not have access to a RVG. In this case, the default behavior should be to accept all prefixes in the RIB and wait until the RVG is downloaded to the routers. If the

RVS has a cached copy of the reachability validation database, it may choose to supply the cached version to its border routers until a new RVG is computed.

IV. RVGS FROM CENTRALIZED ROUTING REGISTRIES

In this section, we discuss how the routing registries that are in use today [13] can be used to construct RVGs.

A. Objects in the Centralized Registry

The Internet Engineering Task Force (IETF) has defined the routing policy specification language (RPSL) in [14]. We would like to focus on the two objects in RPSL: *Aut-Num* and *Route*. These objects are used to indicate the AS peering policy including the other autonomous systems that this AS peers with and the routes that this AS originates. For example, using the information in these objects we can learn that AS1 peers with AS2 and AS3, and originates route 128.9.0.0/16. If we also have the related information for AS2 and AS3, we can expand our database which can then be used to validate reachability advertisements. We also note that summarization is handled by constructing the route object to contain the summary prefixes.

B. Information Flow in and out of the Centralized Registry

The value of any database is measured by the accuracy and the breadth of information stored in it. In order to be able to improve the security of interdomain routing, the routing registries that are used must meet the following requirements:

- accurate and timely information;
- widespread participation;
- periodic purging of stale information;
- able to handle registry queries rapidly for a large number of clients;
- able to receive updates from a large number of clients on a real-time basis.

A registry system that meets these requirements can be constructed with minor modifications to RPSL. First, a time-to-live (TTL) value is added to each object in the routing registry. This prevents stale or incorrect information from forever corrupting the registry. Second, each participating autonomous system uploads their information periodically (for example, every 12 hours) to the centralized registry. Each AS has its own local route registry server which is a replica of the centralized registry. The local registry performs the role of reachability validation server. The local registry may be

updated in two ways: periodically, the local registry pulls information from the centralized registry or in critical cases, the registry can push information to the known local registries. The local registry is then responsible for building an RVG and pushing this to the routers in the AS. The communication between the local registries and the centralized registry should be secured using TLS [15], [16].

One of the concerns about an RPSL-based centralized registry is the authenticity of the data in the registry. The strongest form of authentication in RPSL is the use of a private key to sign the data being submitted to the database. However, this method is not widely used. Other methods available in RPSL, including checking the originator electronic mail address, are considered weak from a security perspective. Hence, we require signature-based authentication to be used for the data in the centralized registries.

C. Optimizations

The following are optimizations to the architecture presented above:

- 1) Do not upload the entire routing policy to the centralized registry unless there is a change. If there is no change, send a refresh message to reset the TTL for the objects.
- 2) Divide the centralized database into segments. Then download only the updated segments to the local registries.

V. RVGS FROM SECURE WEB PAGES

This section describes an approach to authenticating the reachability information needed to build RVGs by leveraging the existing mechanisms of TLS [15].

Each AS participating in the protocol runs a secure web server through which it publishes over HTTPS (ie. HTTP using TLS) a *routing page* describing:

- (i) its AS number;
- (ii) a list of every prefix the AS owns;
- (iii) a list of all of its peers, as links to their routing pages.

If an AS regularly summarizes routes from certain peers, it should publish summarized information in its routing page. The network operators at peer ASes exchange routing page URLs with each other, manually, once, when they first establish a peering relationship. They add these URLs to the peering sections of their routing pages. An RVS crawls this web of routing pages, starting from its own AS's page, to build an RVG whose structure matches that of the web of routing pages.

This approach to collecting information to build RVGs does not rely on any central server or central authority. Rather, it uses a web of trust, where ASes trust reachability information received from their peers on the basis of their peering contracts. When conflicts arise, such as two ASes in the web of routing pages advertising the same prefix, the AS building the RVG can apply local policy to decide how to handle the conflict. Local policy might place more trust in direct peers, more trust in specific backbone ASes, more trust in ASes belonging to a specific domain, etc.

The information in a routing page is authenticated by the TLS channel, which requires the server to present a certificate identifying the server. Expiration or revocation of this certificate should be checked by a mechanism such as OCSP [16]. The identity of the server is ultimately attested to by the root certificate that is used to verify the server certificate. A recent version (1.7) of the Mozilla web browser includes root certificates issued by 19 different certificate authorities. Whether such a broad basis of trust is wise is debatable, but this mechanism is currently in use to validate the identity of thousands of e-commerce web sites. The crawler used to build RVGs could be configured with a more selective set of root certificates.

VI. RVGS FROM A DISTRIBUTED REGISTRY

In this section we present a third method of constructing RVGs that uses a PGP-like certificate-based web of trust to build a distributed registry.

A. Distributed Registry Architecture

In the distributed registry architecture that we propose, there is *no* centralized authority. The information that is required to build the RVG is shared between participating autonomous systems. The shared information includes prefixes advertised by the autonomous systems as well as their peering connectivity. The prefixes advertised by an AS may include their non-BGP speaking customers as well as prefixes owned by the AS itself.

The autonomous systems that participate in the distributed registry transmit their connectivity information and the prefixes that they advertise as the originating AS to their peer autonomous systems. Before they transmit this information, they sign *the information* using one of their private keys. As an example, if there are ten autonomous systems that participate in the registry, then each AS sends the same information to nine peers. Since the information is signed, its authenticity can be verified even when an intermediary AS is used to transmit the

information to its final destination. The ability to use intermediary systems to transfer information securely allows the distributed registry architecture to scale well with large number of participating autonomous systems.

Each RVS, after receiving the connectivity and prefix origination information, builds their version of the RVG and downloads it to the border routers of the AS. Note that the distribution of the prefix and connectivity information may be performed by either a dedicated server, or a router. Similarly, the information can be distributed within BGP (as defined in [9]) or can be distributed out of band using a suitable communication method².

B. Web of Trust between Peers: The Certificates

The distributed registry system is designed to mimic the nature of peering agreements today. We expect the autonomous systems that have peering agreements to exchange and sign each other's public keys. These *signed* public keys are then used to sign routing and connectivity data as well as extend the trust borne by the contractual agreement to other autonomous systems. We expect this trust to be transitive similar to how PGP keys are signed and distributed today [17]. While security purists may argue that this form of security is ultimately not as strong as a fully implemented hierarchical PKI, we believe that a security architecture based on existing operational practices has a much better chance of being deployed than one that is not. Ultimately, the trust criteria is left to the decision of the network operators; however, we recommend to have three classes of trust:

- 1) Trusted Keys (Class A): These are keys that we have seen and signed and stored in our key ring. Class A keys are able to extend the web of trust by signing other keys.
- 2) Semi-trusted Keys (Class B): These are keys that are signed by any class A key. The data signed by these keys are accepted as trustworthy, but these keys are not allowed to extend the web of trust.
- 3) Non-trusted keys (Class C): These keys are not trusted since they are not signed by a Class A key.

The distributed registry servers will trust certificates that are either signed by class A keys that are in their key ring or class B keys that are signed by the class A keys.

The certificates that we propose to use for the distributed registry are the certificates proposed in

²As an extreme example, in networks where the dynamics are very slow and change happens rarely, the information required to construct the RVG can be sent via overnight courier on a CD-ROM

soBGP [18]. We briefly summarize the certificates and their contents below:

- Entity Certificate (Entitycert): An Entitycert establishes a mapping between an AS number and a public key. The Entitycerts correspond to the public keys that we described above.
- Authorization Certificate (Authcert): An Authcert is used to define the prefix(es) that an AS is authorized to advertise.
- AS Policy Certificate (ASPolicyCert): An ASPolicycert defines the routing policy for an AS including the autonomous systems that the AS peers with. The ASPolicycert also is used to revoke old certificates.
- Prefix Policy Certificate (Prefixpolicycert): A Prefixpolicycert is used to communicate per-prefix authentication policy to participating peers.

The Authcerts, ASPolicyCerts and Prefixpolicycerts, when combined, describe the network peering topology as well as the ownership of address ranges for each AS.

The web of trust is achieved by configuring local policy in the distributed registry servers such that the servers are populated by the self-signed class A entity certificates as we discussed. These class A entities extend a web of trust to the class B entities by signing (and hence validating) their entity certificates. Class C certificates when presented to the distributed registry servers are either ignored or used to raise an alarm condition based on configured policy. We also note here that the Authcerts are signed by the class A entities as opposed to a root server entity such as ICANN, RIPE, or ARIN. Therefore, the distributed registry has no dependence on a central authority.

As described in [9], the certificates that we noted above give us both the prefix ownership and peering topology. Given this information, the rest of the steps for securing interdomain routing are identical to what is described in section III. If BGP protocol itself is being used to distribute the security certificates between the RVSSs, then the distributed registry solution does not require a bootstrapping period. Extensions to BGP for distributing security certificates are discussed in [19].

C. Optimizations

Similar to the centralized registry, the effects of changes to the peering information or to the prefixes owned by an AS, can be minimized by propagating incremental changes to the RVG.

VII. COMPARING THE THREE APPROACHES

Our three approaches to constructing RVGs—using routing registries, using secure web pages, and using a distributed registry—present different tradeoffs between security and practicality.

Relative to the other two approaches, the approach using centralized routing registries has the following advantages:

- + easiest deployment;
- + leverages existing infrastructure with minimal modifications.

However, this approach suffers the following drawbacks:

- authentication of the data is hard to achieve based on the existing RPSL specification;
- requires availability of the centralized servers;
- may place significant load on the servers;
- relies on a central authority;
- does not quite reflect existing peering relationships.

The advantages of our second approach based on secure web pages are:

- + easy deployment;
- + no central server;
- + no *single* central authority (but multiple root CAs);
- + leverages existing infrastructure with minimal modifications.

This approach suffers the following drawbacks:

- places trust in the root certificate authorities, where it doesn't really belong.

The advantages of our distributed registry approach are:

- + no central server;
- + no central authority;
- + robust authentication;
- + no unsafe bootstrapping period if certificates are distributed using BGP.

The drawbacks to this approach are:

- requires development of infrastructure/protocols to exchange certificates between service providers.

Presently, we are evaluating all three proposed approaches for suitability to operational deployment.

VIII. CONCLUSION AND FUTURE WORK

Reachability Validation Graphs and Reachability Validation Servers are a new approach to securing the reachability information in BGP UPDATE messages. We have described three means of authenticating the information used to build RVGs. These three approaches

provide different tradeoffs, but all have the following benefits over previous approaches:

- + authentication of prefix and path information without signing BGP Update messages;
- + low overhead on routers;
- + minimal changes needed to existing routers (firmware updates);
- + no modifications to BGP;
- + negligible impact on route convergence times;
- + no change to existing business practices
- + incrementally deployable, and security benefit ramps up rapidly as more autonomous systems participate;
- + a practical, deployable, and manageable architecture.

Most importantly, these approaches fit the operational requirements for network service providers.

In the near future, we will be investigating the suitability of these approaches to operational deployment, exploring incremental updates to the registries, and investigating approaches that allow real-time prefix validation while resulting in negligible convergence time impact and allowing the network to bootstrap.

ACKNOWLEDGMENT

The authors would like to thank Russ White for reviewing this manuscript as well as help guide our research. We gratefully acknowledge the help given to us by Alvaro Retana and Brian Weis. We also want to thank the members of NIAC Internet Hardening Working Group for being early reviewers of this work.

REFERENCES

- [1] Y. Rekhter, T. Li, *A Border Gateway Protocol 4 (BGP-4)*, RFC1771, March 1995.
- [2] John W. Stewart, *BGP4 Interdomain Routing in the Internet*, 1st edition, Addison-Wesley, 1998.
- [3] S. Convery, D. Cook, M. Franz, *An Attack Tree for the Border Gateway Protocol*, Work in Progress, draft-ietf-rpsec-bgpattack-00.txt.
- [4] Sandra Murphy (editor), *BGP Security Vulnerabilities Analysis*, Work in Progress, draft-ietf-idr-bgp-vuln-00.txt.
- [5] Bassam Halabi, *Internet Routing Architectures*, Cisco Press, January 15, 1997.
- [6] Russ White, Bora Akyol, Nick Feamster, *Considerations in Validating the Path in Routing Protocols Draft*, Work in Progress, draft-white-pathconsiderations-02.txt.
- [7] S. Kent, C. Lynn, J. Mikkelsen, and K. Seo, *Secure Border Gateway Protocol (S-BGP)*, Proceedings of ISOC Network and Distributed Systems Security Symposium, Internet Society, Reston, VA, February 2000.
- [8] Ran Atkins, Sally Floyd (editors), *IAB Concerns and Recommendations Regarding Internet Research and Evolution*, Work in Progress, draft-iab-research-funding-03.txt.

- [9] Russ White (editor), *Architecture and Deployment Considerations for Secure Origin BGP (soBGP)*, Work in Progress, draft-white-sobgparchitecture-00.txt
- [10] Geoffrey Goodell, William Aiello et al., *Working Around BGP: An Incremental Approach to Improving Security and Accuracy of Interdomain Routing*, Symposium on Network and Distributed Systems Security, February 2003.
- [11] Barry R. Greene, Philip Smith, *Cisco ISP Essentials*, Cisco Press, April 16 2002 (first edition).
- [12] *Team Cymru BGP Data Page*, <http://www.cymru.com/BGP/>.
- [13] Merit Internet Routing Registry Services, <http://www.irr.net/>.
- [14] C. Alaettinoglu et al., *Routing Policy Specification Language*, RFC2622.
- [15] T. Dierks and C. Allen, *The TLS Protocol, Version 1.0*, RFC2246, January 1999.
- [16] M. Myers et al., *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP*, RFC2560, June 1999.
- [17] Philip R. Zimmermann, *PGP: Source Code and Internals*, MIT Press, June 9, 1995.
- [18] Brian Weis (editor), *Secure Origin BGP (soBGP) Certificates*, Work in Progress, draft-weis-sobgp-certificates-01.txt.
- [19] James Ng (editor), *Extensions to BGP to Support Secure Origin BGP (soBGP)*, Work in Progress, draft-ng-sobgp-bgp-extensions-01.txt.