The ISP Column
A column on various things Internet

July 2021
Geoff Huston

A Survey on Securing Inter-Domain Routing

Part 1 – BGP: Design, Threats and Security Requirements

The Border Gateway Protocol (BGP) is the Internet's inter-domain routing protocol, and after some thirty years of operation BGP is now one of the more venerable of the Internet's core" protocols. One of the major ongoing concerns related to BGP is its lack of effective security measures, and as a result the routing infrastructure of the Internet continues to be vulnerable to various forms of attack.

In Part 1 of this study, we will look at the design of BGP, the threat model and the requirements from a security framework for BGP. In Part 2 we will look at the various proposals to add security to the routing environment and also evaluate the current state of the effort in the Internet Engineering Task Force (IETF) to provide a standard specification of the elements of a secure BGP framework.

1. Introduction

The Internet is a decentralised collection of interconnected component networks (autonomous systems). These networks are composed of end hosts (who originate and/or receive IP packets, and are identified by IP addresses) and active forwarding elements (routers) whose role is to direct IP packets as they pass through the network. The routing system is responsible for propagating the relative location of IP addresses to each routing element, so that routers can make consistent and optimal routing decisions in order to pass a packet from its source to its destination. Routing protocols are used to perform this information propagation.

The Internet's routing system is divided into a two-level hierarchy. One level is *intra-domain* routing, used by the set of autonomous routing systems operating within each component network. The other level is a single *inter-domain* routing system that maintains the inter-network connectivity information that straddles these component networks. A single inter-domain routing protocol, the Border Gateway Protocol (BGP) [1], has provided interdomain routing services for the Internet's disparate component networks since the late 1980's [2]. Given the central role of routing in the operation of the Internet, BGP is one of the critical protocols that provide essential coherence to the Internet.

BGP's underlying distributed distance vector computations rely heavily on informal trust models associated with information propagation to produce reliable and correct results. It can be likened to a hearsay network — information is flooded across a network as a series of point-to-point exchanges, with the information being incrementally modified each time it is exchanged between BGP speakers. The design of BGP was undertaken in the relatively homogeneous and mutually trusting environment of the early Internet. Consequently, its approach to information exchange was not primarily designed for robustness in the face of various forms of negotiated trust or overt hostility on the part of some routing actors.

Hostile actors are a fact of life in today's Internet. It's quite reasonable to characterise today's Internet environment as one where trust must be explicitly negotiated rather than assumed by default. This environment is no longer consistent with the inter-domain trust framework originally assumed by BGP. BGP's mutual trust model involves no explicit presentation of credentials, no propagation of instruments of authority, nor any reliable means of verifying the authenticity of the information being propagated through the routing system. Hostile actors can attack the network by exploiting this trust model in inter-domain routing to their own ends.

An attacker can easily transform routing information in ways that are extremely difficult for any third party to detect. For example, false routing information may be injected, valid routing information removed, or information altered to cause traffic redirection [3] [4] [5]. This approach can be used to prevent the correct operation of applications, to conduct fraudulent activities, to disrupt the operation of part (or even all) of the network in various ways. The consequences range from relatively inconsequential (minor degradation of application performance due to sub-optimal forwarding paths) through to catastrophic (major disruption to connectivity and comprehensive loss of any form of cohesive Internet.

Resisting this subversion of integrity of routing information requires that each BGP speaker has:

- Sufficient information at hand to verify the authenticity and completeness of the information being provided to it via the inter-domain routing system, and
- The ability to generate authoritative information such that other BGP speakers may verify the authenticity of information that this speaker is passing into the inter-domain routing system.

A key question is whether further information can be added into the inter-domain routing environment such that attempts to pervert, remove, or withhold routing information may be readily and reliably detected. Any proposed scheme must also be evaluated for their impact on the scaling properties of BGP.

To ground any such evaluation of BGP, it's useful to briefly review the design of the BGP protocol.

2. The Design of BGP

BGP has undergone a number of refinements over its early operational life. BGP was originally described in RFC1105 in June 1989 [6], allowing the Internet's inter-domain architecture to move on from a constrained architecture of a "core" and attached "stub" domains into a framework of peer routing domains without any central "core". A refinement to this protocol, BGP-2, was described in RFC1163 in June 1990 [7], and a further refinement, BGP-3, was described in RFC1267 in October 1991 [8]. The current version, BGP-4, was first deployed within the Internet in 1993. The RFC describing this protocol, RFC1771 [9], was published in March 1995, and was subsequently refined with the publication of RFC4271 in January 2006 [1]. The core protocol has been stable for some years now, although further refinement has been undertaken through the use of negotiated capabilities undertaken at BGP session startup.

BGP is an instance of what we commonly refer to today as a Bellman-Ford distance vector routing algorithm [10], [11]. This algorithm allows a collection of connected devices (BGP speakers) to each learn the relative topology of the connecting network. The basic approach of this algorithm is very simple: each BGP speaker tells all its other neighbours about what it has learned if the new learned information alters the local view of the network. This is a lot like a social rumour network, where every individual who hears a new rumour immediately informs all their friends. BGP works in a very similar fashion: each time a neighbour informs a BGP speaker about reachability to an IP address prefix, the BGP speaker compares this new reachability information against its stored knowledge that was gained from previous announcements from other neighbours. If this new information provides a "better" path to the prefix, then the local speaker moves this prefix and associated next hop forwarding decision to the local forwarding table and informs all its immediate neighbours of a new path to a prefix, implicitly citing itself as the next hop. BGP keeps a track of the propagation of route advertisements across the inter-domain space by recording the sequence of networks (Autonomous Systems, or AS's) that propagate the route in a route attribute called the "AS Path". A "better" route is one with a shorter AS path and a loop is detected when a BGP speaker sees its own AS in the received AS Path (Figure 1).

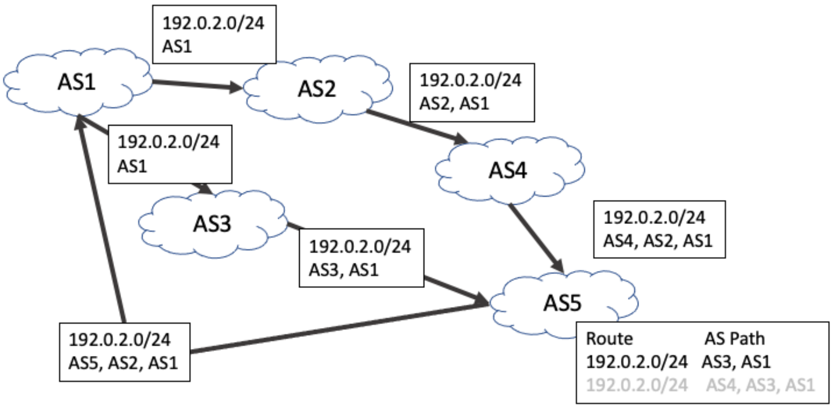

Figure 1 – The Propagation of a route in BGP

In addition, there is a withdrawal mechanism, where a BGP speaker determines that it no longer has a viable path to a given prefix, in which case it announces a "withdrawal" to all its neighbours. When a BGP speaker receives a withdrawal, it stores the withdrawal against this neighbour. If the withdrawn neighbour happened to be the currently preferred next hop for this prefix, then the BGP speaker will examine its per-neighbour data sets to determine which stored announcement represents the best path from those that are still extant. If it can find such an alternative path, it will copy this into its local forwarding table and announce this new preferred path to all its BGP neighbours. If there is no such alternative path, it will announce a withdrawal to its neighbours, indicating that it no longer can reach this prefix.

Across the deployment lifetime of BGP-4 the IPv4 Internet has grown from an average of 20000 distinct routing entries in 1993 to almost 1 million routing entries in 2021 [12]. The growth of the size of the Internet's IPv4 routing table over time is shown in Figure 2.

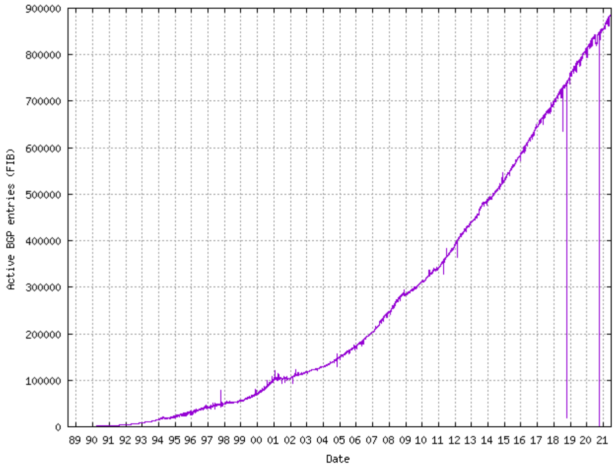

Figure 2 – Internet IPv4 Routing Table Size, from [12]

2.1 BGP and TCP

BGP is not a link-level topology maintenance protocol. BGP assumes the existence of a relatively robust IP forwarding environment at the link level between BGP peers. This has allowed BGP to use the IP transport protocol TCP as a reliable transport protocol to support the protocol's transactions across a BGP peer session.

TCP manages reliable message delivery and flow control between the BGP peers and allows BGP to operate across end-to-end connections whether they reside on the same subnet, or across the Internet. There is no requirement for BGP speakers to be connected on a common media connection, and the choice of TCP allows this flexibility of connectivity by requiring only that a BGP peering session is supported by an IP network.

The TCP stream is divided into messages using BGP-defined markers, where each message is between 19 and 4096 octets in length, extensible to 65,535 octets [11]. The use of a reliable transport service implies that BGP itself need not explicitly confirm receipt of protocol messages. This removes much of the protocol overhead seen in other routing protocols that sit directly on top of a media level connection. There are no message identifiers, no message number initiation protocol, no explicit acknowledgement of messages nor any provision to manage lost, re-ordered or duplicated messages. All this is handled by TCP. The use of a reliable transport protocol also obviates the need for BGP to periodically refresh the routing state by automatically re-flooding the entire routing information set between BGP speakers. After the initial exchange of routing information, a pair of BGP routers exchange only incremental changes to routing information.

2.2 BGP Messages

As TCP is a stream protocol rather than a record-oriented protocol, BGP uses record marking within the TCP stream to delineate logical protocol units, or messages with a 16- byte marker as the BGP message delimiter. The marker is followed by a 2-byte length and a 1-byte type field, making the minimum BGP message size 19 bytes. The repertoire of defined messages are:

- an OPEN message to start a BGP session,
- an UPDATE message to exchange reachability information,
- a NOTIFICATION message, which is used to convey a reason code prior to termination of the BGP session,
- a KEEPALIVE message, used to confirm the continued availability of the BGP peer, and
- a ROUTE-REFRESH request message to request a resend of the routing information.

The common format of BGP messages is shown in Figure 3.

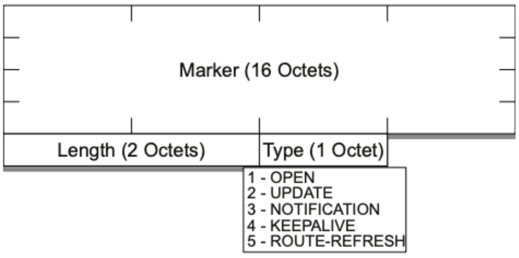

Figure 3 – BGP Common Header Message Format

BGP uses an explicit OPEN message to commence a BGP peering session. This message exchange confirms the identity of the BGP speakers and includes the option for a capability negotiation to understand what optional or extended capabilities are supported by each BGP speaker. A session is active only when both BGP speakers have sent their OPEN messages and neither has rejected the others offered capabilities through a NOTIFICATION response.

Once the session is active, BGP operates via the exchange of UPDATE messages. Each UPDATE message contains a set of address prefixes that are unreachable (withdrawals), followed by a set of common route object attributes, and a set of address prefixes that share this set of attributes (announcements). The withdrawn prefixes are those prefixes where the local BGP speaker sees no reachability, and now wants to withdraw a previous advertisement of reachability. No routing attributes are associated with these withdrawn prefixes. The announced prefixes are those prefixes where the local BGP instance has an updated view of the reachability of a prefix that was previously withdrawn or unannounced or has an updated view of the routing attributes of the locally selected "best" route for a prefix. BGP may group multiple prefixes together in a single UPDATE message but can only do so if all the updated prefixes share a common set of attributes. Within an UPDATE message the withdrawn prefix set, or the announced prefix set, may be empty, but not both. The layout of the BGP UPDATE message is shown in Figure 4.

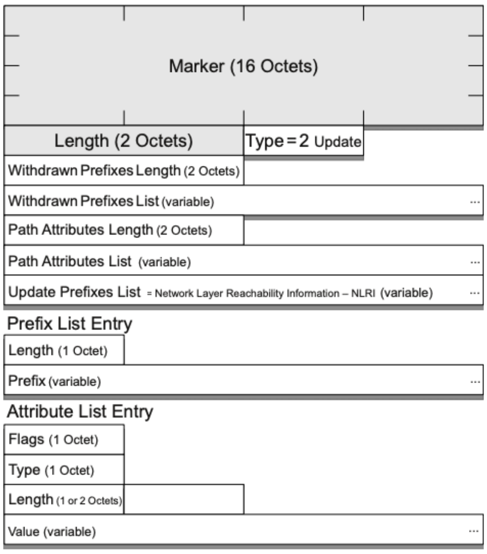
Marker (16 Octets)
Length (2 Octets)
Type = 2 Update
Withdrawn Prefixes Length (2 Octets)
Withdrawn Prefixes List (variable)
...
Path Attributes Length (2 Octets)
Path Attributes List (variable)
...
Update Prefixes List = Network Layer Reachability Information – NLRI (variable)
...
Prefix List Entry
Length (1 Octet)
Prefix (variable)
...
Attribute List Entry
Flags (1 Octet)
Type (1 Octet)
Length (1 or 2 Octets)
Value (variable)
...

Figure 4 – BGP UPDATE Message Format

2.3 AS Path Attribute

BGP binds together the concept of network address blocks and autonomous systems into a path vector-based routing technology. Every route object represented within a BGP-4 route database contains an address prefix and an associated path vector of AS values. BGP does not indicate the precise path a packet should following within an AS, nor does it maintain a complete map of the topology of the Internet at a link-by-link level. BGP uses a level of abstraction which views the Internet as a set of per-AS routing domains, and the role of BGP is to maintain a routing map of the network at this AS level, associating every reachable address prefix with an AS transit path from the current location to the address prefix's originating AS.

One of the most important route object attributes in BGP is the AS Path attribute of UPDATE messages that contain announced routes. As address prefix reachability information traverses the Internet in the form of individual route objects in BGP, this BGP routing information is augmented by the list of ASes that have processed this route information thus far, forming the AS Path attribute of a route object. Each BGP speaker adds its own AS value to the route object's AS Path attribute when passing the route object through an eBGP session.

This AS Path attribute allows straightforward suppression of the looping of routing information, using the simple algorithm that a local AS will reject any forwarded route object that already contains its own AS in the AS Path attribute. Also, the length of the AS Path vector forms the BGP route metric. A local BGP system, when attempting to select one from a number of potential route objects that refer to the same address prefix, will, in the absence of any local policy directive, prefer the route object with the shortest AS Path length. In addition to undertaking the role of path metric and loop detector, the AS Path attribute serves as a versatile mechanism for policy-based routing, where a local AS can alter the default preferences for route selection based on local policy settings coupled with pattern matching rules to be performed on the AS Path.

Withdrawals have no associated AS Path.

2.4 BGP Route Selection Process and Routing Policies

A BGP speaker may receive two or more announcements for the same address prefix from different peers. The "best" announcement is selected as the locally used announcement, and this announcement is the one that is announced to its BGP peers. BGP defines an ordered sequence of comparisons to determine which route object is selected by the local BGP speaker as the preferred route to use

- Prefer the route object with the highest value for LOCALPREF attribute value
- Prefer the route object shortest AS PATH attribute length
- Prefer the lowest origin value

- Prefer the lowest MULTI EXIT DISCRIMINATOR attribute value
- Prefer the minimum IGP cost to the NEXT HOP address given in the route object
- Prefer eBGP over iBGP-learned routes
- If using iBGP, Prefer the lowest BGP Identifier value.

Although a network administrator's usually employs routing policies depending on her needs [14], [15], within the generic BGP route selection process the highest priority selection rule is that a route for a more specific address prefix is to be preferred over that of a covering prefix

3. The BGP Threat Model

One approach to providing a taxonomy for threats in routing in general, and BGP in particular, is to view a BGP peer session as a conversation between two BGP speakers and pose a number of questions relating to this conversation. These questions are:

- How do we talk?
  The manner in which the BGP session between the BGP speakers is secured such that the conversation is not altered, disrupted or hijacked and is protected from unauthorised eavesdropping.

- Whom am I talking to?
  Verifying the identity of the other party and verifying that they are authorised to speak for the routing entity that they purport to represent.

- What are you saying?
  Verifying the authenticity and completeness of the routing information being passed in the BGP session.

- Why should I believe you?
  Verifying that the routing information represents the current state of the forwarding system.

- How recent is your information and is it still valid?
  Verifying for how long routing information is valid and whether the information is still current.

Each of these security questions can be further deconstructed to a set of specific objectives, as well as recognising a set of specific threats.

3.1 Securing a BGP Session

A BGP session between two BGP speakers is assumed to have some level of integrity at the session transport level.

BGP assumes that the messages sent by one party are precisely the same messages as received by the other party, and assumes that the messages have not been altered, reordered, have spurious messaged added into the stream or have messages removed from the conversation stream in any way, and given that BGP uses a TCP transport session some of these assumptions are reasonable, but others less so.

As with any long-held TCP session, a BGP peer session is vulnerable to eavesdropping, spurious session reset, session capture, message alternation and denial of service attacks, all via what we might think of as conventional TCP attack vectors.

The threat at the BGP level is that a third party may attempt to break into the TCP session as an interception attack in the middle, and thereby alter the BGP message flow between the two end points. One form of threat is by injection, where the attacker injects spurious messages into the BGP session. Direct on-the-wire interception allows the attacker to have knowledge of the TCP sequence numbers, thereby making injection a trivial task. Even if the attacker is not able to intercept or eavesdrop the BGP session, it is still possible to attempt to guess the current sequence number.

While this is often impractical in the case of injecting data into the session, if all that is to be injected is a TCP Reset, then the sequence number guess only has to sit within the current TCP window in order to be recognised as a valid reset TCP message [16]. Another form of threat is by active intermediation where the attacker sits on the connection between the two BGP speakers and intercepts all traffic in both directions. In this case the attacker has complete control of the BGP message stream and can perform any form of message alteration. A variation of this form of threat is by session hijacking, where the third party intrudes upon an active BGP session and injects its own traffic into the message stream that allows the third party to take over the session and masquerade as one of the parties to the BGP session. As timing is important in the overall performance of BGP another form of attack at the session level is to delay messages. While the content of the messages are unaltered, the implicit timing signals within the message stream are altered by this form of intervention, potentially causing the local BGP speaker to behave differently and fall out of sync with its routing peers.

Another form of attack is a replay attack, where older BGP messages are replayed into a hijacked TCP session. One form of this replay attack could be to replay a pair of messages that withdraw and then announce the same address prefix. Route Flap Damping (RFD) [17] [18] is a widespread defensive BGP configuration that monitors the frequency of BGP updates for a given prefix from each peer, and if the update rate exceeds a locally set threshold the peer's advertisement of this prefix will be locally suppressed for a damping interval. The replay of updates could be used to trigger an RFD response in the remote BGP speaker [19]. If a route is fully dampened through RFD, updates for this prefix will not be advertised by the BGP speaker for a damping interval (commonly 60 minutes), possibly causing a route to be disrupted within that time frame. Another form of replay attack is to replay a route advertisement for a previously withdrawn prefix, possibly in conjunction with some form of prefix hijack attack.

Another form of threat is by withholding traffic. BGP uses keepalive timers to determine remote end "liveness". By intercepting and withholding all messages for the hold down timer interval, a third party can force the BGP session to be terminated and reset. This causes the entire route set to be re-advertised upon session resumption so that repeated attacks of this form can be an effective form of denial of service for BGP.

It is also possible to undertake a saturation attack on a BGP speaker by sending it a rapid stream of invalid TCP packets. In this case the processing capability of the BGP speaker is put under pressure, and the objective of the attack is to overwhelm the BGP speaker and cause the BGP session to fail and be reset. This is particularly problematic if the BGP session uses MD5 or IPSEC as session protection protocols, as the cryptographic function overhead also applies to the injected packets, increasing the processing overhead on these spurious injected packets.

The underlying aspect of the BGP protocol is that BGP itself has no enforced minimum level of message protection. BGP messages are, by default, placed into the TCP stream without encryption or additional message wrapping of message sequencing. Any threat that is applicable to long held TCP sessions applies to this default mode of BGP operation.

3.2 Verifying BGP Identity

BGP sessions commence by passing the local AS to the remote end of the session in the BGP OPEN message and receiving the remote end's AS in the received OPEN message. BGP itself does not verify these asserted AS identities, and it is theoretically possible for a remote party to masquerade itself as another AS and assert an identity in BGP that cannot be directly verified by the other party, or by any third party that subsequently receives this routing information. Most BGP implementations provide a level of protection against this threat by applying a constraint that the local BGP speaker will only initiate a peer session with a configured remote IP address, and reject all other TCP connection attempts, and furthermore will not complete the BGP OPEN message exchange if the AS in the OPEN message does not match the AS number associated with the remote end IP address in the configuration.

This approach places a heavy reliance on the out-of-band process of BGP configuration, and if an attacker can compromise or take control over BGP equipment connected to the Internet or use social engineering to convince a network administrator to configure incorrect information into the BGP configuration then it is possible to masquerade as a different party in BGP and potentially inject incorrect information into the routing system.

The real question here is: “Are you really who you claim to be?” Here is it necessary for the BGP speaker to be able to confirm the validity of the peer’s claim to be speaking for an AS.

3.3 Verifying BGP Information

The objective here is that of verifying the authenticity and completeness of the routing information being passed in the BGP session. The intention of BGP is that a local BGP speaker provides to all its BGP peers a complete feed of its locally selected route objects.

Once a session is opened with a remote BGP speaker the local BGP instance believes everything it is told without further qualification. The threat is that a BGP peer can deliberately feed false information to the local BGP instance, which BGP itself will be unable to detect as false. This could be in the form of suppression of routing information, or in the form of alteration of the route object that is being passed, or the invention of spurious route objects. The BGP speaker could be asserting that an AS Path is genuine when it reflects an artificial path, or that it has the authority to originate an advertisement for a prefix when, in fact, no such authority exists.

A BGP speaker may preserve all the attributes of a route object, but alter the prefix set to be the equivalent collection of more specific prefixes. The deliberate alteration of routing information can cause the local BGP instance to make an incorrect choice of a local best path and also cause the local BGP instance to propagate this incorrect information to its neighbours.

Not only could the BGP speaker be passing incorrect attributes for an address prefix in order to bias the local route selection process, it could also be providing incorrect information regarding the prefix itself. The prefix that is the subject of the route object could be a prefix that has never been allocated and should not be legitimately routed, or the prefix is an aggregate address prefix that spans both allocated and unallocated address space.

Prefix hijacking is a major threat to the integrity of the BGP routing. The fundamental weakness here is that BGP provides no explicit means of verifying the authenticity of the address prefixes that are listed in a BGP UPDATE message, nor in the authenticity of the attributes of the prefix, including the origination information and the AS Path vector. The threat here is that by deliberately altering this information the local BGP speaker can be induced to make incorrect route selection decisions and thereby make incorrect forwarding decisions for IP traffic.

A known common problem illustrative of exploiting this vulnerability is operational misconfiguration [20], which could result in propagating more specific routes, and other forms of route leakage or withholding that may impact on the routing decisions made by other BGP speakers. This form of verification of intentionality by a remote BGP speaker is far more challenging — while these forms of security mechanisms are intended to verify that the received information matches the original information that was passed into the routing system, they are incapable of verifying that such information is consistent with the true intent of the originator of the information.

3.4 Verifying Forwarding Paths

The overall intention of the BGP protocol is to distribute the current binding of address to location such that individual routers can make accurate judgements about how to populate their local forwarding tables and hence make optimal local decisions for each packet that passes along the shortest path to its ultimate destination.

BGP does not provide any ability for a local BGP speaker to validate that the route advertisements it receives from a BGP peer accurately represents the current state of the network’s forwarding system. The threat model here is that a bad actor in the routing system may make a different forwarding decision to that being advertised in the routing system.

This can represent a subversion of local policies, theft of carriage capacity, deliberate denial of service, the potential to eavesdrop on a conversation or to support the interception and alteration of application-level transactions. Even a completely secured control plane does not avert such vulnerabilities [21].

3.5 The Consequences of Attacks on the Routing System

The ability to alter the routing system provides a broad array of potential consequences [3]. The consequences fall into a number of broad categories, which are briefly described here.

1) The ability to eavesdrop. The forwarding system can be altered so as to pass all traffic to a class of destination addresses via a certain path. This allows the attacker to attempt to pass all such traffic through an eavesdropping location prior to conventional delivery. In such a case the parties may not be aware that an eavesdropping attack is taking place.

2) Denial of service. The simplest form of a denial of service is where traffic to an address prefix is passed to a point where it is then discarded. Routing loops also are a form of denial of service, where not only will the traffic to a destination address prefix never reach its intended destination, but the traffic will be held in the loop for the life of the packet TTL field. For sufficiently short loops the potential exists for the loop to act as a link load amplifier, where the traffic on the loop is several times the traffic load being addressed to the affected destination address prefix.

3) The potential to masquerade. Subversion of routing allows sites to masquerade as other sites; The routing system will misdirect the traffic to the masquerading site. The consequences of such an attack can vary from the specific, where a particular site is targeted, to the more generic where authoritative DNS servers are the subject of the masquerading attack, and the DNS responses are believed as being authentic. In this case if the masquerading occurs at the level of the root of the DNS hierarchy incorrect information can be provided to any query, allowing for the attack to then be extended to any site.

4) The ability to steal addresses and obscure identity. Routing an unallocated address is subtly different to routing an already allocated address. Here the consequence is not displacement of traffic forwarding to incorrect locations in the network, but the assertion of the existence of addresses and forwarding paths to those addresses that should not exist in the network in the first place. The consequence is the ability to use addresses on the network that have no allocation registration information associated with them, allowing the originator of the routing attack some degree of ease to mount an anonymous attack at the application level. Such forms of attack have been observed to be associated with SPAM and botnet controllers where anonymity of the attack coordinator is desired.

4. Security Requirements

The primary requirements for securing BGP are securing the transmission of the data payload of the BGP protocol and securing the semantics of that payload.

The security requirements for transmission are such that the data received by a BGP speaker can be cryptographically verified to have been sent by the BGP peer, that the data is not a replay of previously transmitted data, and that no data has been removed from the transmission [22].

There is no strict requirement for encryption of the BGP payload, as the routing information being exchanged is not intrinsically confidential to the two parties involved. The security requirements for the semantics of the payload concern specifically some selected fields (transitive attributes) of the BGP UPDATE message. The BGP speaker must be able to verify that the advertised prefix is valid, and that the originating AS has been duly authorised by the legitimate right-of-use holder for that prefix. The BGP speaker should also be able to validate that the AS Path in the UPDATE represents a valid inter-AS transit path through the network in terms of inter-AS topology and AS transit policies, and that the prefix reachability information has been propagated along the reverse inter-AS Path [22].

It is noted that route withdrawals and non-transitive announcement attributes are local in nature, and thus do not need to be transitively protected in a similar fashion to route origination and the AS Path attribute of announcements. Withdrawals and local attributes can be adequately protected by BGP peer session protection.

The associated requirements for a secure inter-domain routing system include that the additional use of security credentials and verification of routing information should not alter the temporal properties of the BGP protocol, and that authentication of the security credentials should occur in the same time frame as the BGP message processing operation. It is also a requirement that piecemeal incremental deployment should be feasible [23], [24], [25]. A secure operational mode should be a capability negotiation with each BGP peer, with the ability to support backward compatibility with those BGP peers who do not recognise such a capability. It seems to be a good idea to start deployment of BGP security on the most connected nodes and incrementally deploy it towards least connected nodes.

Additionally, it puts the question on how a party which uses security credentials deals with information arriving from a peer which does not use any security credentials. Having no security credentials does not necessary mean that the information is wrong. Importantly, in these piecemeal deployment scenarios there should be some incremental benefit of piecemeal deployment to those actors who choose to supply such security credentials and to those who chose to validate routing information using these credentials.

A routing system, secure or otherwise, should never make route selections that include routing loops. It is preferred that in a fully secured environment a secure routing system would be able to converge on best paths that are either identical to or no worse than an unsecured BGP speaker would select, assuming that such paths can be validated in a secure environment. In an environment of partial adoption of secure routing systems, it is recognised that a BGP speaker may use local preference settings that prefer sub-optimal paths that have preferred security credentials over unsecured paths.

The trust model of routing appears to involve two forms of trust. The first is a trust environment related to the public network and the legitimacy of use of a public address and the legitimacy of use of a public AS number. It is necessary to be able to verify that a particular party has the right to use these number resources in a public context. The closest fit in the form of a trust model for verification of this assertion of right of use is a public authority that can provide authoritative information on the distribution of these numbers. This approach leads to a rooted hierarchy model of trust, where the trust anchor is this public authority.

The second form is a trust environment in private contexts, where the use of an address or AS number is bounded by a specific context of use, and the trust in an assertion of a right of use is one made in the context of this bounded environment. In this environment there is no clear ability to use public authorities as a trust anchor and other means of trust that may involve reputation or web of trust concepts may be appropriate.

A general security approach to BGP should be able to encompass that diversity of deployment environments and the corresponding diversity of authority models.

5. Tools for Securing BGP

The vulnerabilities of BGP arise from four fundamental weaknesses in the BGP and the inter-domain routing environment. These are:

- No mechanism to protect the integrity, currency and source authenticity of BGP messages.
- No mechanism to verify the authenticity of an address prefix and an AS origination of this prefix in the routing system.
- No mechanism to verify the authenticity of the attributes of a BGP UPDATE message.
- No mechanism to verify that the local cache RIB information is consistent to the current state of the forwarding table.

The other observation about BGP security is that it appears that by far the most straightforward form of attack is to obtain control and configuration access to a deployed router and use this compromised platform as the base for launching attacks on the routing system. In the face of such an encompassing attack on the control instruments of the routing system, BGP session-level security needs to be placed in some perspective. It is not possible to prevent routers from attempting to generate false information as long as routers themselves are in a position to be compromised.

The consequent vulnerability on the routing system, as distinct from a narrower view of BGP, is that there is no mechanism that limits the extent to which a misbehaving routing element can make inaccurate claims about reachability in the routing system.

5.1 The Security Toolset for BGP Session Protection

The available tools for securing BGP start at the level of the BGP TCP session and encompass the tools that are used to protect TCP and the two ends of the TCP session.

The TCP protection mechanisms include the generalized TTL security mechanism [26], [27], which is intended to limit the effective radius of potential attack on the session to hosts that lie on or within the worst-case hop count radius between the two BGP speakers, and host-level defences against TCP SYN attacks [28]. In many ways this is an effective form of defence when using multi-hop BGP sessions in that the attacker cannot subvert the defence, but it still leaves the session vulnerable to any attacker that lies within the TTL radius.

Greater levels of session protection can be obtained by using cryptographic protection. Over time the IETF has worked on three approaches to protect the BGP TCP through cryptographic protection. These are:

- the use of IPSEC [29]. IPSEC has not been widely used for BGP sessions, and the reasons behind this relate to the complications for rekeying IKE/IPSEC sessions and the potential DDOS vector [30].

- the TCP MD5 signature option [31]. While the MD5 signature option has some potential weaknesses when compared with IPSEC [29], MD5 is considered preferable to no form of TCP protection at all, particularly with respect to the TCP Reset injection attack. However, there are issues with re-keying a long-held session, and the BGP speakers probably need to use graceful restart mechanisms in conjunction with MD5 to perform a re-key of the session.

- The TCP Authentication Option [32], which the IETF has marked as a replacement to the earlier MD5 approach. The TCP Authentication Option supports stronger crypto algorithms compared to MD5. It uses a two-fold security approach that reduces the critical reliance on a user configured key. This approach also allows the configuration of up to 64 keys for a session and provides a simple key coordination mechanism by giving the ability to change keys (move from one key to another) within the same connection without causing any TCP connection closure. By comparison, changing TCP MD5 keys during an established connection might cause a flap or restart in the connection, which in the content of BGP may have operational implications.

From time to time the topic of BGP over TLS [33] is raised and its possible that sooner or later we might hear of BGP over QUIC [34]. The salient question is one of balancing the additional burden of adding more transport choices to BGP implementations to the likely benefits that these additional choices may provide. As we've seen in the IPv6 transition and more recently in the increasing diversity of choices for encrypting DNS transactions, adding more options can offer just add confusion and impede adoption instead of accelerating it.

However, the most important guideline in securing BGP sessions is to use multi-hop BGP and multi-access LAN sessions sparingly and prefer to use a direct 1:1 channel connection when such a choice is available.

5.2 The Security Toolset for BGP Message Protection – The Number Resource Public Key Infrastructure (RPKI)

In addition to message integrity protection provided by transparent session level protection mechanisms, the tools to provide protection of the integrity of BGP messages relate to the use of digital signatures to provide a set of credentials that allow relying parties to verify the correctness of the information carried as the message payload in BGP.

The reason for the use of digital signatures as opposed to an integrity check using some form of shared secret is due to the observation that the number and identities of all eventual recipients of the information is not known in advance, and non-repudiation is desirable [3]. Verification of the contents of a message is not only a test of whether the message has been altered in any way during its transit between BGP speakers, but a test of whether the message represents correct origination information and correct operation of the processing of the message during the process of message propagation (*authenticity*).

This requirement implies a need to establish a means of verification of information where the author of any security credentials relating to origination and propagation are not necessarily known to the relying party that is attempting to validate the information. This typically invokes a form of validation that relies upon *third party transitive trust*, where the relying party is attempting to build a testable chain of trust between its trust anchor and the party or action that is the subject of the verification operation. Conventionally, this requirement implies the use of some form of Public Key Infrastructure (PKI). In this case we are not looking to use such a PKI to validate claims of identity, authority to perform a particular function, or some form of verifiable attribution. We need some form of mechanism to associate a public key with an IP address prefix or an AS number in a sense of *functional control*, where the certification authorities in this PKI are attesting that the certified subject has functional control of a collection of IP number resources (AS numbers and IP address prefixes). The associated certificate issuance practices are intended to support transitive trust in such attestations of association.

We have adopted a structure using X.509 public key certificates and a certificate extension that uses a canonical list of IP address resources and AS numbers [35] as the foundation for this Number Resource PKI (RPKI) [36]. Verification of a digital signature entails a test of the authenticity and current validity of the associated certificate that describes the public key of the address or AS number holder in the context of a structured set of signed relationships between certificate issuers and subjects. To put it another way, the holder of the matching private key is the current functional controller of those IP addresses and AS number and can digitally sign authorities and attestations about such number resources on the basis of that functional control.

Given that the discourse of BGP messages is about address prefixes and AS numbers, the RPKI provides a solid foundation for digital signatures to be associated with various routing actions that are described in BGP messages [37]. It does not attest in any way to the identity of these number resource holders.

Anchoring the model of authority and trust in the RPKI certificate structure has resulted in a framework where the issued certificates are aligned with the IP address and AS number allocation and assignment framework. If an Internet Registry has issued a set of IP addresses and AS numbers to an entity, then this registry would be able to publish a public key certificate that associated a private key provided by the entity with the IP resource set. Further allocations from a registry to a registry address holder would result in re-issuance of the certificate for the address holder with a larger resource set, while reduction in this set would result both re-issuance and revocation of the previous certificate. This certificate framework would allow auditing of the certificate state by inspecting the registry contents of the Internet Registries, as the intention of this PKI is to mirror the overall state of the number registries with the set of issued certificates.

The RPKI is different from many other PKIs as the requirements related to adding digital signatures to the routing domain is different from many other PKI deployment environments. The common question that PKI's attempt to answer is: "Is this data authentic?" The data is signed with a digital signature, and the key used to generate that signature is described in a certificate. The validity of that certificate can be ascertained through the use of a collection of certificates and certificate revocation lists (CRLs), such that a relying party may validate the data by using their local trust anchor(s) and constructing a *validation path* of issuer-subject chained certificates from a trust point to the digital signature. If this collection of certificates is bundled with the digital signature and the data itself, then the only data item that needs to be distributed outside the data flow are the PKI trust anchors.

5.3 Distributing the RPKI Data Collection

When the RPKI is combined with a use case for the routing domain we are looking at a design space which is somewhat atypical in the PKI world. For example, in the WebPKI the certificates that are passed between server and client in the initial exchange of a Transport Level Security (TLS) session are certificates that related to the particular domain name used in the TLS session being established [33]. The critical distinction here between the secure client/server transactions using the WebPKI and the promulgation of routing information in the routing system using RPKI is that the routing system presents the *entire* routing domain to each relying party on a continuous basis. Each relying party needs to have continual access to the entire RPKI certificate and CRL collection, rather than the TLS practice of processing individual signatures and certificates on an as-needed basis.

This requirement for all participating entities to have access to all the RPKI data at all times poses a design challenge in how to manage this RPKI and use it in a routing protocol such as BGP.

A basic approach here is for each Internet Registry to publish their certificate products in their own publication point. This is analogous to the pre-CDN model of web content publication, where each element is independently published. Of course, in this case while publication is easy, while the onus is shifted to the relying party client, or BGP validator, who has to assemble a local cache of all RPKI signed data. It becomes the task of clients of the RPKI to maintain local cache of the entire RPKI by continuously sweeping across these publication points looking for, and retrieving, changes, and validating all such signed objects as they are received.

At this point BGP updates could be passed this local RPKI engine and the data in the update can be compared against the validated information contained in the local RPKI cache. If RPKI validation was performed at the point of acceptance into the local cache (i.e. discarding all RPKI products that cannot be validated within the framework of the RPKI validation procedures), then the route information could be verified against the assembled (and validated) crypto data without a high on-demand crypto processing overhead. An alternative approach is the express the validation outcomes from the local RPKI cache as a filter list. If this were maintained on a router then the overheads in passing route objects through such a filter would be little different to the many other routing policy maps used in operational configurations.

The drawback in this distributed approach is the need for these clients to constantly sweep all the RPKI publication points to ensure that their local cache is up to date. What is "up to date" is relative here, but it is worth remembering is that the average time to propagate a BGP update across the global Internet depends on the average AS path length (around 4 to 5 AS's on average at present) and the interaction with BGP's MRAI timers. While the worst case would 300 seconds (assuming that the full MRAI delay would be applied on each eBGP session), the fastest case is well under a second. So how quickly should the local cache be populated to keep up with the propagation of routing information in BGP? Before leaping to a target time its also worth remembering the scaling question. With around 100,000 distinct ASes in the Internet's routing system, today's worst-case scenario is some 100,000 RPKI clients performing a sweep across 100,000 distinct RPKI publication points every few seconds (or even more frequently if the RPKI system is intended to be highly responsive).

In some ways this is putting the load on the wrong side of the information distribution process. By making the relatively infrequent publication process one that involves a local action without any associated notification of a change then the burden is shifted to the client set, who have to poll every publication point continuously just to ascertain if anything has changed. To put it as plainly as possible, this particular information distribution design is completely broken! If the client set is known in advance (such as is the case in the DNS in synchronising the information across primary and secondary authoritative services) one could use notification mechanisms. But in the case of the RPKI system the publishers of authoritative information have no information as to who are the clients who need to be notified of a change in a CA's part of the RPKI data collection. Hence, notification is not a viable option in this framework.

These relatively formidable scaling issues can be mitigated by changing the publication behaviour, in a manner analogous to the way in which CDNs have improved web performance by shifting content publication models to various permutations of anycast-related models of content replication. In the context of the RPKI this could entail the use of a smaller set of RPKI publication points that are shared by many RPKI certificate issuers, or the reduction in the number of independent Certification Authorities who each publish their own products through the extensive use of Registration Agents (RAs). The information being published is signed there is no particular benefit to retrieve the data from any particular publication point. As long as the data can be validated by the client, then the client can be assured that the data that they have retrieved is most likely to be genuine, irrespective of the location used for the retrieval. It is possible to use third party aggregators in such a role, who would take on the task of continuous monitoring of all RPKI CA publication points and publish an aggregated data set of all current RPKI data. This could be taken further into a *push* model by having clients register their interest in updates from the intermediary and allow the intermediary to send them information updates as they are received from the primary CA publication point sources. Again, it has to be noted that the information is signed, so the potential of the intermediary to alter the RPKI information is limited. The design gap in such mediated distribution approaches is to provide a mechanism for clients of these aggregated intermediaries to be assured that the collection being provided by the intermediary is the entire collection of RPKI data, and any credible intermediary approach would need to explicitly address this issue of *information completeness*.

However, while these approaches reduce the load imposed on clients of the RPKI by increasing the load on information publication, such aggregated publication models also create critical points of concentration of routing data, and a sustained denial of service attack again such aggregate publication points could have a major impact on the routing system as the local RPKI caches lose currency and coherency.

These approaches have their own strengths and risks. Highly distributed publication models impose undue costs on clients as the clients need to maintain an aggregate and current data collection in their local cache. Aggregate data publishing models relieve load from clients but have some unresolved issues in terms of assured completeness of the aggregated data collection and also runs the risk of creating new points of vulnerability in terms of the consequence of denial-of-service attacks launched against these aggregated publication points.

The current RPKI operational framework that is used in the Route Origination Validation (ROV) tool [38] uses this approach of an *out of band* RPKI *pull* system together with some use of aggregated RPKI publication points. An RPKI client's local cache currency performance level is phrased in units of minutes rather than seconds, and the overall system operates at a level of coherency that is at a time scale of hours rather than minutes. The initial design of this RPKI distribution system is for each client to operate autonomously and maintans a local cache to keep synchronised with the current state of all the RPKI publication points using the *rsync* protocol [39] together with the concept of a *manifest* [40] that allows a client to ensure that they have retrieved the entirety of the data available at each RPKI publication point. The *rsync* protocol was subsequently found to be a poor choice for this role [41], and these days the RPKI Repository Delta Protocol (RRDP) is the preferred RPKI repository synchronisation tool [42].

In terms of the application of the RPKI to the BGP environment there is the obvious question to be asked here: If the intent of the flooding system is to provide a reliable and efficient way to flood current information to all clients, then why not just use BGP itself? BGP is an internet-wide information flooding protocol using a *push* based approach that is intended to ensure that all BGP speakers have a consistent and current collection of reachable route objects. If the set of clients who want to maintain an up-to-date synchronised local cache is isomorphic to the set of BGP speakers, then adding a BGP message payload type in the same manner that AFI/SAFI indicators are already used in Multi-Protocol BGP today seems only logical.

Part of the reason why the RPKI has had to re-invent this particular wheel of reliable flooding lies in the strictures imposed on the standardisation effort in the IETF, where the SIDR Working Group was constrained from proposing changes to the BGP protocol itself. In retrospect, this apperars to have been a rather suboptimal and, in hindsight, extremely poor piece of guidance from the IESG at the time.

If one could contemplate changes to BGP, then one approach to the RPKI distribution tasks is to maintain the association of the validation material with the data, and in the context of the routing environment this would staple a collection of certificates (and CRLs) to each route object. In a sense this would attempt to reproduce the TLS model in BGP, where each prefix being updated would have a subset of the RPKI certificates stapled to the update that will permit an associated signed attestation to be validated within the framework of the RPKI.

This is not without additional impositions and costs that would be imposed on the operation of the BGP protocol and upon BGP speakers. Stapling crypto credentials to BGP updates would bloat both the volume of stapled data (through the use of long validation chained paths and long-term certificate issuance policies which, in turn, create extended CRL lists) and the amount of crypto processing of these stapled digital credentials. There would be a significant level of the retransmission of certificates on a pair-wise basis in such a system if the protocol was to bundle the entire RPKI validation chain data with every routing protocol update. The validation processing load would also be likely to be beyond the processing capabilities of most routers, and there are considerations of the maximum message size in the BGP protocol itself (which, until RFC8654 [13], published in October 2019, was 4,096 octets) which limited the amount of attached data that can be placed into BGP.

None of these issues are intractable in nature, and there have been a number of proposals that attempt to optimise such additional loads and processing demands. We will look at some of these proposals in Part 2 of this survey.

Coming in Part 2

In Part 2 we will take these various requirements and tools and look at the various proposals that have been published for securing BGP. We will also evaluate the current state of the effort in the IETF to standardise a secure BGP Framework.

References

[1] Rekhter, Y., Li, T., and Hares, S. *A Border Gateway Protocol 4 (BGP4)*, RFC 4271, DOI 10.17487/RFC4271, January 2006. https://www.rfc-editor.org/rfc/rfc4271.txt

[2] Rekhter, Y., *Experience with the BGP Protocol*, RFC 1266, DOI 10.17487/RFC1266, October 1991. https://www.rfc-editor.org/rfc/rfc1266.txt

[3] Murphy, S., *BGP Security Vulnerabilities Analysis*, RFC 4272, DOI 10.17487/RFC4272, January 2006. https://www.rfc-editor.org/rfc/rfc4272.txt

[4] Barbir, A., Murphy, S., and Yang, Y., *Generic Threats to Routing Protocols*, RFC 4593, DOI 10.17487/RFC4272, October 2006. https://www.rfc-editor.org/rfc/rfc4272.txt

[5] Ballani, H., Francis, P., and Zhang, X., *A study of prefix hijacking and interception in the Internet*, SIGCOMM Computer Communications Review, vol. 37, no. 4, pp. 265–276, 2007.

[6] Lougheed, K. and Rekhter, Y., *Border Gateway Protocol (BGP)*, RFC 1105, DOI 10.17487/RFC1105, June 1989. https://www.rfc-editor.org/info/rfc1105

[7] Lougheed, K. and Rekhter, Y., *Border Gateway Protocol (BGP)*, RFC 1163, DOI 10.17487/RFC1163, June 1990. https://www.rfc-editor.org/info/rfc1163

[8] Lougheed, K. and Rekhter, Y., *Border Gateway Protocol 3 (BGP-3)*, RFC 1267, DOI 10.17487/RFC1267, October 1991. https://www.rfc-editor.org/info/rfc1267

[9] Rekhter, Y. and Li, T., *A Border Gateway Protocol 4 (BGP-4)*, RFC 1771, DOI 10.17487/RFC1771, March 1995. https://www.rfc-editor.org/info/rfc1771

[10] Bellman, R., “On a routing problem”, *Quarterly of Applied Mathematics*. **16**: 87–90, 1958. doi:10.1090/qam/102435

[11] Ford, L., *Network Flow Theory*, RAND Corporation, Paper P-923, 1956. https://www.rand.org/pubs/papers/P923.html

[12] G. Huston, *The BGP Report*, https://bgp.potaroo.net

[13] Bush, R., Patel, K., and Ward, D., *Extended Message Support for BGP*, RFC 8654, DOI 10.17487/RFC8654, October 2019, https://www.rfc-editor.org/info/rfc8654

[14] Griffin, T. and Huston, G.,, *BGP Wedgies*, RFC 4264, DOI 10.17487/RFC4264, November 2005, https://www.rfc-editor.org/info/rfc4264

[15] F. Wang and Gao, L., *On inferring and characterizing internet routing policies*, in IMC '03: Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement. New York, NY, USA: ACM, 2003, pp. 15–26.

[16] Ramaiah, A., Stewart, R., and Dalal, M., *Improving TCP's Robustness to Blind In-Window Attacks*, RFC 5961, DOI 10.17487/RFC5961, August 2010, https://www.rfc-editor.org/info/rfc5961

[17] Villamizar, C., Chandra, R., and R. Govindan, *BGP Route Flap Damping*, RFC 2439, DOI 10.17487/RFC2439, November 1998, https://www.rfc-editor.org/info/rfc2439

[18] *RIPE Routing Working Group Recommendations on Route Flap Damping*. January 2013. https://www.ripe.net/publications/docs/ripe-580

[19] Sriram, K., Montgomery, D., Borchert, O., Kim, O., and D. Kuhn, *Study of BGP peering session attacks and their impacts on routing performance*, Selected Areas in Communications, IEEE Journal on, vol. 24, no. 10, pp. 1901–1915, Oct. 2006.

[20] Mahajan, R., Wetherall, D., and Anderson, T., *Understanding BGP misconfiguration*, in SIGCOMM '02: Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications. New York, 2002, pp. 3–16.

[21] Goldberg, S., Halevi, S., Jaggard, A., Ramachandran, V., and Wright, R., *Rationality and traffic attraction: incentives for honest path announcements in BGP*, SIGCOMM Computer Communications Review, vol. 38, no. 4, pp. 267–278, 2008.

[22] Christian, B., and Tauber, T., Eds, "BGP Security Requirements", November 2008, Internet Draft. https://datatracker.ietf.org/doc/html/draft-ietf-rpsec-bgpsecrec-10

[23] He, X., Papadopoulos, C., and P. Radoslavov, *A framework for incremental deployment strategies for router-assisted services*, in INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE, vol. 2, March-3 April 2003, pp. 1488–1498 vol.2.

[24] Suchara, M., Avramopoulos, I., and Rexford, J., *Securing BGP incrementally*, in CoNEXT '07: Proceedings of the 2007 ACM CoNEXT conference. New York, NY, USA: ACM, 2007, pp. 1–2.

[25] Rexford, J., and Feigenbaum, J., *Incrementally-deployable security for interdomain routing*, Conference For Homeland Security, 2009. CATCH '09. Cybersecurity Applications & Technology, March 2009, pp. 130–134.

[26] Gill, V., Heasley, J., and D. Meyer, *The Generalized TTL Security Mechanism (GTSM)*, RFC 3682, DOI 10.17487/RFC3682, February 2004. https://www.rfc-editor.org/info/rfc3682

[27] Gill, V., Heasley, J., Meyer, D., Savola, P., Ed., and C. Pignataro, *The Generalized TTL Security Mechanism (GTSM)*, RFC 5082, DOI 10.17487/RFC5082, October 2007. https://www.rfc-editor.org/info/rfc5082

[28] Eddy, W., *TCP SYN Flooding Attacks and Common Mitigations*, RFC 4987, DOI 10.17487/RFC4987, August 2007, https://www.rfc-editor.org/info/rfc4987

[29] Kent, S. and K. Seo, *Security Architecture for the Internet Protocol*, RFC 4301, DOI 10.17487/RFC4301, December 2005. https://www.rfc-editor.org/info/rfc4301

[30] Weis, B., *Why IPsec and BGP don't play well together in real networks*, Security Area Working Group presentations, IETF 66, July 2006. https://www.ietf.org/proceedings/66/slides/saag-2.pdf

[31] Heffernan, A., *Protection of BGP Sessions via the TCP MD5 Signature Option*, RFC 2385, DOI 10.17487/RFC2385, August 1998. https://www.rfc-editor.org/info/rfc2385

[32] Touch, J., Mankin, A., and R. Bonica, *The TCP Authentication Option*, RFC 5925, DOI 10.17487/RFC5925, June 2010. https://www.rfc-editor.org/info/rfc5925

[33] Rescorla, E., *The Transport Layer Security (TLS) Protocol Version 1.3*, RFC 8446, DOI 10.17487/RFC8446, August 2018. https://www.rfc-editor.org/info/rfc8446

[34] Iyengar, J., Ed., and M. Thomson, Ed., *QUIC: A UDP-Based Multiplexed and Secure Transport*, RFC 9000, DOI 10.17487/RFC9000, May 2021. https://www.rfc-editor.org/info/rfc9000

[35] Lynn, C., Kent, S., and K. Seo, *X.509 Extensions for IP Addresses and AS Identifiers*, RFC 3779, DOI 10.17487/RFC3779, June 2004. https://www.rfc-editor.org/info/rfc3779

[36] Huston, G., Michaelson, G., and R. Loomans, *A Profile for X.509 PKIX Resource Certificates*, RFC 6487, DOI 10.17487/RFC6487, February 2012. https://www.rfc-editor.org/info/rfc6487

[37] Lepinski, M., Chi, A., and S. Kent, *Signed Object Template for the Resource Public Key Infrastructure (RPKI)*, RFC 6488, DOI 10.17487/RFC6488, February 2012 https://www.rfc-editor.org/info/rfc6488

[38] Huston, G. and Michaelson, G., *Validation of Route Origination Using the Resource Certificate Public Key Infrastructure (PKI) and Route Origin Authorizations (ROAs)*, RFC 6483, DOI 10.17487/RFC6483, February 2012. https://www.rfc-editor.org/info/rfc6483

[39] Tridgell, A., Mackerras, P., *rsync*, 1996. https://rsync.samba.org/

[40] Austein, R., Huston, G., Kent, S., and M. Lepinski, *Manifests for the Resource Public Key Infrastructure (RPKI)*, RFC 6486, DOI 10.17487/RFC6486, February 2012. https://www.rfc-editor.org/info/rfc6486.

[41] Michaelson, G., and Ellacott, B., *rsync considered Inefficient and Harmful*, IETF 89, March 2014. https://www.ietf.org/proceedings/89/slides/slides-89-sidr-6.pdf

[42] Bruijnzeels, T., Muravskiy, O., Weber, B., and R. Austein, *The RPKI Repository Delta Protocol (RRDP)*, RFC 8182, DOI 10.17487/RFC8182, July 2017. https://www.rfc-editor.org/info/rfc8182

Disclaimer

The above views do not necessarily represent the views or positions of the Asia Pacific Network Information Centre.

Author

Geoff Huston AM, B.Sc., M.Sc., is the Chief Scientist at APNIC, the Regional Internet Registry serving the Asia Pacific region.

www.potaroo.net