The ISP Column

A monthly column on things Internet

September 2019
Geoff Huston

Why is Securing BGP just so Damn Hard?

Stories of BGP routing mishaps span the entire thirty-year period that we've been using BGP to glue the Internet together. We've experienced all kinds of route leaks from a few routes to a few thousand or more. We've seen route hijacks that pass by essentially unnoticed, and we've seen others that get quoted for the ensuing decade or longer! There are ghost routes and gratuitous withdrawals. From time to time we see efforts to craft BGP packets of death and efforts to disrupt BGP sessions through the injection of spoofed TCP resets. After some 30 years of running BGP it would be good to believe that we've learned from this rich set of accumulated experience, and we now understand how to manage the operation of BGP to keep it secure, stable and accurate. But no. That's is not where we are today. Why is the task to secure this protocol just so hard?

Are we missing the silver bullet that would magically solve all these BGP issues? If we looked harder, if we spent more money on research and tried new approaches, then would we find the solution to our problems? I doubt it. It's often the case that those problems that remain unsolved for such a long time are unsolved because that are extremely hard problems and they may not even have a solution. I suspect securing BGP falls into this "extremely hard problem" category. Let's look at this in a bit more detail to explain why I'm so pessimistic about the prospects for securing BGP.

However, perhaps we might start with a more general question: Why are some Internet issues so challenging to solve, while others seem to be effortless and appear to solve themselves? For example, why was the IPv4 Internet an unintended runaway success in the 90's, yet IPv6 has been a protracted exercise in industry-wide indecision?

Success and Failure Factors in the Internet

Some technologies have enjoyed success from the outset in the Internet. IPv4, of course, would be clearly placed in ther runaway success category, but perversely enough IPv6 would not. NATs have been outstanding successful, and the TCP transport protocol is still with us and it still drives the Internet. The DNS is still largely unchanged after some 30 years. More recently, content distribution systems and streaming protocols have been extremely successful, and most of today's public Internet service could be characterized as a gigantic video content streaming network.

Why did these technologies succeed? Every case is different, of course, but there are some common success factors in all these technologies.

Piecemeal deployment

> One important factor in many aspects of the Internet is the ability to support piecemeal deployment. Indeed this *loosely coupled* nature of many aspects of the Internet is now so pervasive that central orchestration of many deployed technologies in the Internet is now practically impossible. The Internet is just too big, too diverse, and too loosely coupled to expect that flags days will work. Any activity that requires some general level of coordination of actions across a diversity of networks and operational environments is a forbidding prospect. Instead, we need to be able to deploy these technologies in a piecemeal basis where one network's decision to adopt

a technology does not force others to do the same, and one network's decision not to adopt a technology does not block others from adoption.

Relative Advantage to Adopters

The Internet is not a command economy and, generally, technologies are not adopted by fiat or regulatory impost. Market economies still operate in the Internet, and adoption is often fuelled by the perception of relative market advantage to early adopters. Technologies that reduce the service cost in some manner, or improve the service offering, or preferably both at the same time, tend to support an early adopter market advantage, and in so doing the technology enjoys rapid market uptake.

Economies of Scale

Technologies where more is cheaper also tend to be adopted. As the number of adopters increase the unit price of the technology and its use should go down, not up. This implies greater market incentives to adopt as adoption increases, creating a positive feedback loop between adoption and service operation costs.

Alignment of Common and Individual benefit

A common question in the Internet context is: What if everyone did it? If the technology generates benefits only when it is used by a few entities and is less efficient when used by everyone is less likely to succeed. For example, an aggressive TCP flow management protocol may generate benefits when only one or two users use it, but when everyone uses it, the protocol may be poor at generating a stable equilibrium across all users.

These success factors relate to success in a diverse, widely distributed and loosely coupled environment.

But the Internet has left a trail of failures every bit as voluminous, if not more so, than its history of successes. For example, spam in the email space is a massive failure for the Internet, as is our vulnerability to many forms of DDOS attacks. In a similar vein, after more than 20 years of exhortations to network operators, I think we can call spoofed source address filtering (or BCP 38) a failure. It's very sensible advice and every network operator should do it. But they don't. Which makes it a failure.

Secure end systems and secure networks are both failures, and the Internet of Trash looks like amplifying these systemic failures by many orders of magnitude by introduc. The broader topic of securing our transactions across the Internet also has its elements of failure, particularly in the failure of the public key certification framework to achieve comprehensive robustness. IPv6 adoption is not exactly a runaway success so far. The prospects of the Internet of Things amplifying our common vulnerability to poorly crafted, poorly secured and un-maintained endpoints should create a chilling prospect of truly massive cascading failure.

Again, there appear to be common factors for failure which are the opposite of the previous attributes. These include technologies where there is dependence on orchestration across the entire Internet, and technologies that require universal or near universal adoption. The case where there are common benefits but not necessarily individual benefits, and where there is no clear early adopter advantage lies behind the issues relating to the protracted transition to an IPv6-only Internet.

What makes a technical problem *hard* in this context?

It might be **technically challenging**: While we understand what we might want that does not mean we know how to construct a solution with available technologies.

It might be **economically perverse**: The costs of a solution are not directly borne by the potential beneficiaries of deploying the solution.

It might be motivated by **risk mitigation**: We are notorious for undervaluing future risk!

So now let's look at BGP routing security in this light. After 30 years why are we still talking about securing BGP?

Why is Securing BGP So Hard?

Here is my top ten of the reasons why securing BGP represents such a challenging problem for us.

1. **Noone is in Charge**
   There is no single 'authority model' for the Internet's routing environment. We have various bodies that oversee the Internet's domain name space and IP address space, but the role of a routing authority is still a vacant space. The inter-domain routing space is a decentralised distributed environment of peers. The implication of this characterisation of the routing space is that there is no objective reference source for what is *right* in routing, and equally no clear way of objectively understanding what is *wrong*. When 2 networks set up an eBGP session neither party is necessarily forced to accept the routes advertised by the other. If one party is paying the other party then there may be a clearer motivation to accept their advertised routes, but the listener is under no specific obligation to accept and use advertised routes. Noone is in charge and there is no authority that can be invoked to direct anyone to do any particular action in routing. To be glib about this, there is no such thing as the routing police.

2. **Routing is by Rumour**
   We use a self-learning routing protocol that discovers the network's current inter-AS topology (or part of that topology to be more accurate). The basic algorithm is very simple, in that we tell our immediate eBGP neighbours what we know, and we learn from our immediate BGP neighbours what they know. The assumption in this form of information propagation is that everyone is honest, and everyone is correct in their operation of BGP. But essentially this is a hop-by-hop propagation, and the reachability information is not flooded across the network in the form of an original route reachability advertisement. Instead, each BGP speaker ingests neighbour information, applies local policy constraints, generates a set of advertisements that includes locally applied information and, subject to outbound policy constraints, advertises that information to its neighbours. This is in many ways indistinguishable from any other form of rumour propagation, and as there is no original information that is necessarily preserved in this protocol it is very challenging to determine if a rumour (or routing update) is correct or not, and impossible to determine which BGP speak was the true origin of the rumour.

3. **Routing is Relative not Absolute**
   Distance Vector protocols (such as BGP) work by passing their view of the best path to each destination to their immediate neighbours. They do not pass all their available paths, just the best path. This is a distinct point of difference to the operation of Shortest Path First (SPF) algorithms, which flood link level reachability information across the entire network, so that each SPF speaker assembles a (hopefully) identical view of the complete topology of the network and each SPF speaker assembles a set of nest hop decisions that (hopefully) is consistent with all the other local decisions by each other SPF speaker. What this means is that not only does each BGP speaker only have a partial view of the true topology of the network, it is also the case that each BGP speaker assembles a view that is relative to their own location in the network.

   Each eBGP speaker will assemble a different routing table, and that mean that there is no single 'reference' routing view could be used to compare with these dynamically assembled local views. In BGP there is no absolute truth about the topology of the network, as there is only a set of relative views that is assembled by each eBGP speaker.

4. **Routing is Backwards**

Routing works in reverse. When a network advertises reachability information relating to an IP address prefix to a neighbour, the result is that the neighbour may use this link to send traffic to this network. Similarly, if a BGP speaker accepts an inbound routing advertisement from a neighbour it may use this to send outbound traffic to that neighbour. The flow of routing information is the opposite to the consequent flow of traffic in the network.

5. **Routing is a Negotiation**
Routing has two roles to play. The first is the discovery and maintenance of a usable view of the topology of the network, relative to the local BGP speaker as we've already noted. The second is that of routing policy negotiation. When two networks peer using BGP (here I'm using the term *peer* in the strict protocol sense, in that the two networks are adjacent neighbours rather than describing any business relationship between the two networks) there is a policy negotiation that takes place. Each network has local traffic export preferences and will selectively filter incoming route advertisements to that the preferred outbound routing paths that are selected maximises the local traffic export policy preferences of the network. Similarly, each network has local traffic import preferences, and will attempt to advertise route advertisements that maximise conformance to its preferred traffic import preferences.

   Such policies are often entirely logical when viewed as business relationships. Customer routes are preferred to transit and peer routes (*peer* in a business sense). Customer networks should not re-advertise provider or peer routes to other providers or peers. When given a choice, networks would prefer to use provider paths that present the lowest cost and highest performance, while at the same time would prefer to use customer routes that represent the highest revenue potential. BGP is the protocol that attempts to discover a usable state within this set of route import and export constraints.

6. **Routing is non-Deterministic**
This may sound odd, given that there is an underlying inter-AS topology and a part of BGP's task is to discover this topology. This part of BGP's operation is deterministic, in that a stable BGP state represents a subset of this overall topology. BGP (or at least untampered BGP) cannot create fictitious inter-AS links. However the policy constraints introduce a level of non-determinism, See BGP Wedgies (http://www.potaroo.net/ispcol/2004-09/2004-09-isp.htm, RFC4264) for a description of one such case of non-determinism.

   BGP is able to generate outcomes that can be described as "unintended non-determinism" that can result from unexpected policy interactions. These outcomes do not represent misconfiguration in the standard sense, since all policies may look completely rational locally, but their interaction across multiple routing entities can cause unintended outcomes, and BGP may reach a state that includes such unintended outcomes in a non-deterministic manner.

   Unintended non-determinism in BGP would not be so bad if all stable routing states were guaranteed to be consistent with the policy writer's intent. However, this is not always the case. The operation of BGP allows multiple stable states to exist from a single configuration state, where some of these states are not consistent with the policy writer's intent. These particular examples can be described as a form of *route pinning*, where the route is pinned to a non-preferred path.

7. **There is no Evil Bit**
For many years April 1 saw the publication ofg an April Fool's RFC. In 2003 RFC3514 described the *evil bit*: "If the bit is set to 1, the packet has evil intent. Secure systems SHOULD try to defend themselves against such packets. Insecure systems MAY chose to crash, be penetrated, etc."

   In a security framework *bad* data does not identify itself as being *bad*. Instead, we use digital signatures and other forms of credential management to allow others to correctly identift *good* or *genuine* data. The assumption here is that if all of the *good* data carries credentials that can be verified, then all that's left is *bad* or, at best, *untrustworthy*. However, there is a major assumption in this assertion, name.ly one of universal adoption. If we know that only some data has credentials, then the absence of such credentials does not help us in identifying what is trustworthy data.

In some environments, such as TLS, we are not interested in everyone, just the credentials of the remote party we are trying to connect to. In this case partial deployment can be mitigated to some extent by labelling those destinations where TLS validation is required. However, BGP is the entirety of the routing system. A BGP speaker amasses a complete view of reachability of all prefixes. In a scenario of partial deployment, where some routes have associated credentials, and some do not then the task of determining which routes to use becomes a significant challenge.

8. **Risk is Hard**
   Taking measures to mitigate risk is a bit like buying a ticket in a reverse lottery. In a normal lottery everyone spends money to buy a ticket, and there is only one winner. All the ticket buyers can see that there is a winner, and in some manner this justifies their purchase of a ticket. But in a reverse lottery the winner is rewarded by not being a victim of some malicious attack. Because the attack has been deflected the winner is completely unaware that they are a winner and no one can see the value in buying a ticket in the first place. In such systems of common risk mitigation, where everyone pays, but there are no clear winners, the system is difficult to sustain.

9. **Because Business**
   In the internet each component network is motivated by conventional business economics, attempting to balance factors of risk and opportunity in their enterprise. Spending resources on security must be seen to either reduce business risk or increase an enterprise's competitive advantage.

   But it's all too often the case that network enterprises under-appreciate risk. Such investments in risk mitigation do not necessarily translate into a visible differentiator in the market, and in a competitive environment the result is a higher cost of service without some associated service differentiation. Where the risk mitigation results in a common outcome there is little to be had in the way of a competitive advantage.

10. **We actually don't know what we want!**
    It is extremely challenging to identify a 'correct' routing system, and it is far easier to understand when and where an anomaly arises and react accordingly. This situation could be characterized as: we know what we don't want when we see it, but that does not mean that we can recognize what we actually want even when we may be seeing it! This is partially due to the observation that the absence of a recognizable 'bad' does not mean that all is 'good'!

The task of trying to build a secure BGP system is a bit like trying to stop houses from burning. We could try to enforce behaviours of both the building industry, of our furniture and fittings and of our own behaviours that make it impossible for a house to catch fire. Or we could have a fire brigade to put out the fire as quickly as possible. For many years, we've opted for the latter option as being an acceptable compromise between cost and safety.

There are parallel here with BGP security. It would be an ideal situation where it would be impossible to lie in BGP. Where any attempt to synthesis BGP information could be readily identified and discarded as being bogus. But this is a very high bar to meet, and some thirty years of effort are showing just how hard this task really is.

It's hard because no one is in charge. It's hard because we can't audit BGP, as we have no standard reference data set to compare it with. It's hard because we can't arbitrate between conflicting BGP information, because there is no standard reference point. It's hard because there are no credentials that allow a BGP update to be compared against the original route injection, because BGP is a hop- by-hop protocol. And it's hard because BGP is the aggregate outcome of a multiplicity of opaque local decisions.

There is also the problem that it is just too easy to be bad in BGP. Accidental misconfiguration in BGP appears to be a consistent problem, and it's impossible to determine the difference between a mishap and a deliberate attempt to inject false information into the routing system.

We've become accustomed to ignoring an inter-domain routing system that can be easily compromised, as acknowledging the issue and attempting to fix it is just too hard. But maybe this passive acquiescence to BGP abuse is in fact a poor response in a broader context. If the only response that we can muster is hoping that individually our routes will not be hijacked, then we are obviously failing here.

Consequences

What are the consequences of routing mishaps and malfeasance? If this is an ever-present threat, then how have we coped with it in today's Internet?

There are three major risk factors in route hijacks: disruption, inspection and interception.

Disruption involves injecting a false route that makes the intended destination unreachable or injecting a withdrawal that also generates a similar outcome. It could be that the radius of disruption is highly localised, or it could be intended to be Internet-wide. In either case the result is that communications are disrupted, and the service is rendered unavailable.

Inspection involves an exercise of redirecting the traffic flow to a destination to pass through a network that performs traffic inspection in some manner. Depending on the form of transport level encryption that is being performed such forms of traffic inspection can be of limited value, but even the knowledge of communicating pairs as endpoints can in and of itself be a valuable source of information to the eavesdropper. Such inspection is not necessarily detectable by the endpoints, given that the packets are not altered in any manner, such their route through the network.

Interception is perhaps the more insidious threat. The threat involves the same technique of redirection of a traffic flow to a point where the traffic is intercepted and altered. Prior to the widespread use of end-to-end transport security, it could be argued that this was a thoroughly pernicious form of attack, where user credentials could be stolen, and the integrity of network transactions could be compromised. It has been argued that the widespread use of TLS negates much of this threat from interception. An interceptor would need to have knowledge of the private key of the site being attacked in order to break into a TLS handshake and inject themselves into the session in a seamless manner. But perhaps this is too glib a dismissal of this threat. Firstly, as has been seen in a number of recent attacks, many users are too quick to dismiss a certificate warning and persist when the wiser course of action would be to refrain from proceeding with the connection. Secondly, as also has been seen numerous times, not all trusted CAs are worthy of the implicit trust we all place in them. If a trusted CA can be coerced into issuing a false certificate where the private key is known to the interceptor, then the interception attack is effective even where the session is supposedly 'protected' by TLS.

Let's put this together in a hypothetical attack scenario. Let's say you find an online trusted CA that uses a DNS query as a proof-of-possession of a DNS name. This is then the criteria used by the CA to issue a domain name certificate. Let's find a target domain name that is not DNSSEC-signed. This is of course a not uncommon criteria given the relative paucity of DNSSEC-signing in today's DNS. A fake certificate can be generated by using a routing interception attack on the name servers of the target domain name and providing a crated response for the CA's DNS challenge. The attacker now has a fake Certificate for the target name. Now the CA will enter this fake certificate into the certificate transparency logs, but the attacker still has enough time to launch the second part of the attack, which is an interception attack using this fake, but still trusted, certificate to intercept TLS sessions directed to the target name server.

Conclusions

BGP security is a very tough problem. The combination of the loosely coupled decentralized nature of the Internet and a hop-by-hop routing protocol that has limited hooks on which to hang credentials relating to the veracity of the routing information being circulated unite to form a space that resists most conventional forms of security.

It's a problem that has its consequences, in that all forms of Internet services can be disrupted, and users and their applications can be deceived in various ways where they are totally oblivious of the deception.

It would be tempting to throw up our hands and observe that as we've been unable to come up with an effective response in thirty years we perhaps should just give up with the effort and concede that we just have to continue to live with a vulnerable and abused routing system.

But I'm unwilling to make that concession. Yes, this is a hard and longstanding problem, but it's a very important problem. We will probably spend far more time and effort in trying to prop up the applications and services environment when the underlying routing infrastructure is assumed to always be unreliable and prone to various forms of abuse.

I'll look at what we have done so far in this space and try and place current efforts into this broader context in a followup article.

Disclaimer

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

Author

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

www.potaroo.net