December 2013
Geoff Huston

MITM and Routing Security

An internet draft was added to the drafts repository in late November this year that contained some interesting observations about efforts to secure the Internet's inter-domain routing system.

> "This document describes a very simple attack vector that illustrates how RPKI-enabled BGPSEC machinery as currently defined can be easily circumvented in order to launch a Man In The Middle (MITM) attack via BGP."
> http://tools.ietf.org/html/draft-ietf-grow-simple-leak-attack-bgpsec-no-help-03

This is apparently not just a theoretical exercise. Through extensive monitoring of trace route data, the folk at Renesys believe that they have isolated a number of instances where traffic has been deliberately redirected using this form of MITM routing attack.

> "For years, we've observed that there was potential for someone to weaponize the classic Pakistan-and-Youtube style route hijack. Why settle for simple denial of service, when you can instead steal a victim's traffic, take a few milliseconds to inspect or modify it, and then pass it along to the intended recipient?
>
> This year, that potential has become reality. We have actually observed live Man-In-the-Middle (MITM) hijacks on more than 60 days so far this year. About 1,500 individual IP blocks have been hijacked, in events lasting from minutes to days, by attackers working from various countries."
> http://www.renesys.com/2013/11/mitm-internet-hijacking/

If the motivation behind the effort behind securing BGP was to allow any BGP speaker to distinguish between routing updates that contained "genuine" routing information and routing updates that contained contrived or false information, then these two reports point out that we've fallen short of that target. What's gone wrong? Why are certain forms of routing MITM attacks all but undetectable for the RPKI-enabled BGPSEC framework?

Perhaps its useful to go back to the basics of the BGP protocol first, and use this to understand the strengths and limitations of the BGPSEC framework, and then see how these simple route leaks evade detection.

BGP and BGPSEC

BGP is a distance vector routing protocol, where each active routing element selects its best local forwarding decision for each known destination, based on selecting the minimal path cost, and then advertises this set of known destinations and the associated path cost to all its routing neighbours. Distance vector algorithms are conceptually similar to a sequential computation, where each router's local decision process depends on the outcomes of a similar process being perform by neighbouring routers. BGP is an instance of an explicit path distance vector routing protocol. Unlike link state routing algorithms no single routing entity has a complete "map" of the network topology. The total sum of knowledge each routing entity can assemble is that which is provided by its immediately adjacent neighbours, and all it can provide back to these neighbours is a list of all known destinations its its preferred next hop to reach that destination.

If one is looking to secure the information being passed in BGP, then the issue here is that the mode of operation of the protocol is essentially hop-by-hop. Very little information is actually passed from the original point of advertisement of the route. For example, withdrawals are local transactions, and are a

signal that a BGP speaker's immediate neighbour cannot reach a certain network. It does not necessarily imply that the network has been shut down and withdrawn at the source of the network, but can also be the result of a local connectivity change that affects reachability. For this reason withdraws cannot be secured with origin-based credentials attached to the withdrawal. The result, however, is that an aberrant BGP speaker can withhold the propagation of withdrawals from its BGP neighbours, or generate spurious withdrawals in the form of a MITM "attack".

What BGPSEC can secure in the context of propagation of routing information is essentially limited to two fields in the update message: the prefix being announced, and the AS Path attribute of the announcement. The way in which this is undertaken is through a "forward signing" context. The prefix holder generates a digital signature that covers the prefix and the AS numbers that have been authorised to originate an announcement for this prefix. AS the route announcement is propagated across the inter-AS domain each AS generates a digital signature that covers its own AS and the AS to whom the announcement is being sent. When a BGP speaker receives a fully signed announcement it can assess for itself whether the prefix is a genuine prefix, whether the prefix has been authorised to be advertised as reachable, and whether the AS Path in the announcement matches the sequence of AS's in the route propagation path from the original advertisement to the BGP speaker.

What this security mechanism is intended to achieve is to limit the manner in which an attacker can manipulate routing information. An attacker cannot simply announce more specific prefixes and attempt to launch a traffic redirection attack, as, presumably, the attacker has not been duly authorised by the prefix holder to originate such a route announcement. An attacker cannot take an existing announcement and synthesise a re-advertisement that has a shorter AS Path in an attempt to redirect traffic. In theory, all a potential attacker can do is take an advertisement that it has received, add its own AS number to the AS Path and propagate this advertisement to its BGP neighbours, which is precisely the same set of actions that are permitted for all other BGP speakers.

MTIM Attacks

If BGPSEC limits the set of actions permitted to a potential attacker to precisely the same set of actions that are available to any other BGPSEC-enabled BGP speaker, then how can this attack take place?

The elements of the answer lie in the business arrangements that occur between networks, which are expressed in BGP in the form of routing policies.

If one network contracts another to be its "transit provider" then it is expected to announce its routes to its transit provider, and the transit provider is expected to announce all the routes it is aware of to its customer (commonly this takes the form of a single announcement of a default route, but there are circumstances where the explicit enumeration of routes is necessary).

What if a network is a customer of two or more transit providers? Supposedly, much the same thing happens - the customer network announcements its routes to each of the transit network providers, and learns routes from each of its transits.

The MITM routing attack is a very simple perversion of this latter case: it selects routes announced by one transit provide and propagates them to the other transit provider. The additional AS hop via the attacker's AS would extend the AS Path length and thereby reduce the relative preference for this readvertisement, were it not for a widely used routing policy setting that prefers routes announced by customers over routes announced by peers or transit upstreams. Even though the customer-announced re-advertisment may have a longer AS Path the local routing policies may still prefer this route simply because it is announced by a customer.

Critically, there is no protocol violation in such a form of re-advertisement, and even if all the parties involved used BGPSEC, it will not flag this re-advertisement as an invalid route announcement.

This is illustrated in Figure 1, where the conventional traffic route to reach destinations in 192.0.2.0/24 from Upstream B is a path via Upstream A. If the MITM network also advertises this prefix to upstream B, then it is possible that Upstream B will prefer this path, due to a local preference of preferring routes announced by customers over routes announced via peers. If this is the case then the MITM network can now inspect all traffic passing between Upstream B (and its other customers) and the target network.

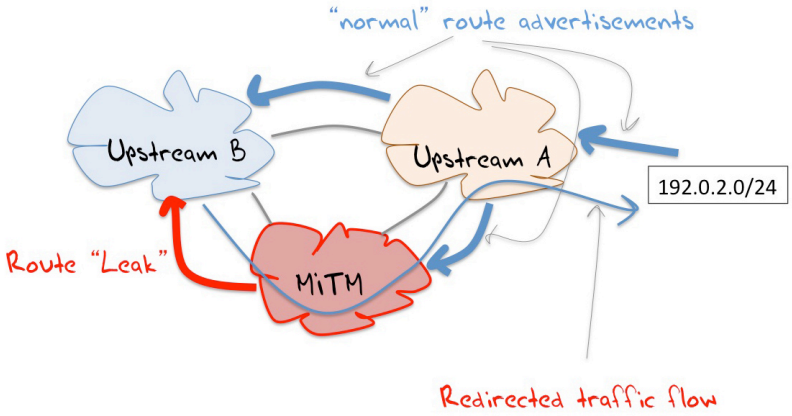

Figure 1 – MITM Route Redirection

Attack Mitigation

There have been various proposals intended to mitigate this form of attack.

Upstream Tokens?

One approach is intended to exploit the "no valley" constraint on routing paths. This constraints starts with a classification of all inter-AS BGP peering sessions into one of Customer-to-Transit ("upstream"), Transit-to-Customer ("downstream") or Peer. Networks typically wish to avoid being placed into a position of acting as an unfunded transit network, and this occurs when prefixes learned from one upstream provider are re-advertised to another upstream.

This can be transformed into a protocol mechanism. The originating AS can to sign a "upstream attribute" (or "token", as shown in Figure 2) in the BGP announcements before passing it to its upstream service providers. Each AS that receives an announcement with this signed attribute signs across the token with its own AS key before passing it to its upstreams, and so on. If the announcement is passed to a peer AS, or to a customer AS, then the attribute, if present, is stripped from the announcement. An upstream should not accept an announcement from its customers unless the announcement has the "upstream attribute" intact.

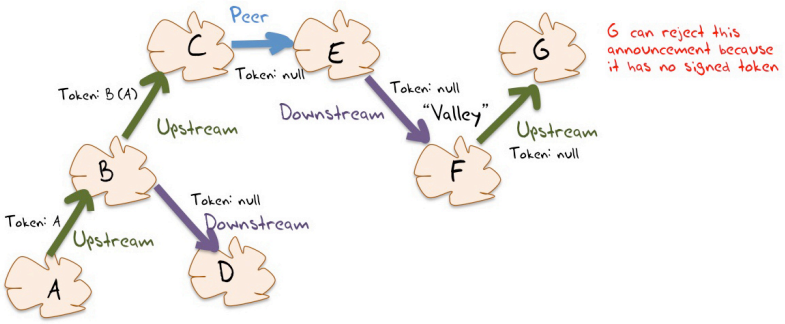

Figure 2 – Upstream Tokens

The weakness in this approach is that it assumes that this classification of inter-AS relationships into customer/provider or peer is comprehensive and uniform, whereas the evidence gathered from observing BGP points to some degree of variation. Protocol mechanisms that codify this constrained set of relationships into BGP has the risk of forcing otherwise valid BGP announcement paths to be considered as invalid.

But maybe in leaping into protocol mechanisms and cryptography we are overlooking the blindingly obvious in this form of approach, and perhaps we should consider whether we already have the tools to effectively address this type of risk. The conventional approach to preventing such route leaks is to maintain route filters.

Route Filters?

A network operator can insist that all customers and all peers enumerate specifically the list of prefixes that they intend to announce. The network operator can use these lists to maintain filter lists on the edge routers to the network's customers and peers. When a route is received, the route can be passed through the filter list, and is only accepted once it passes through the filter. (Figure 3)

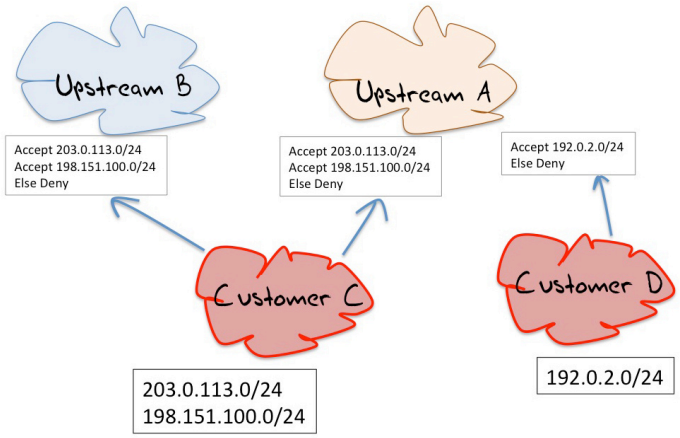

Figure 3 – Route Filters

For customers and peers that present with a small number of prefixes this can be maintained relatively easily, but its an approach that does not have good scaling properties. Its one thing to run filter lists for a handful of customers, each with a handful of routes, but when the numbers start to head into the hundreds, then its a case of using automated tools. And when the numbers rise again into the thousands then the efforts of maintaining large filters, even with various operational support tools starts to get quite challenging.

Maintaining route filters for the larger providers pose operational challenges in terms of escalating administrative overhead, cost of maintenance, and accuracy and timeliness of the entries that are in the filters. For a customer while there may be a strong motivation to add new entries to the filter on a timely basis, there is actually no motivation to remove the out-of-date entries, and the consequent filter bloat becomes a real challenge to manage.

It seems that when the collection of routes gets sufficiently large, or when then the level of administrative updates in terms of adds, removals and amendments gets too large, then providers often choose to take each other "on trust" and drop the use of administratively maintained routing filters.

At this point filtering based on AS path rather than by prefix starts to look tempting. It is possible to augment, or even replace, the filter lists of prefixes with filter lists of AS Paths. In this case if the other party attempts to re-advertise learned routes in an unanticipated manner, then the AS Path of these routes would trigger the filter action.

Unfortunately this approach not as good as it sounds. The issue with AS Path filters is that they assume a universal environment of well-intentioned actors. If the route "leak" also involves AS Path manipulation, then the AS Path filter approach is of little use. It is also possible to have unintentional route leaks that involve AS Path manipulation. Some route leaks have involved mapping externally-learned eBGP routes into the interior routing domain and then mapping all such interior routes routes back into eBGP and passing them out to the peer as if they were originated directly in this network.

Another form of route leak involves leaking out a bevy of more specific internal routes to externally connected networks. In this latter case there is no direct subversion of third party routes, but if the internal route set encompassed a million or more routes, the leak of such a large volume of routes into the inter-domain routing space would likely trigger a number of limits and result in BGP session teardowns and consequent third party connectivity damage. In both of these cases the advertised leak looks like the local AS is the originator, and an AS Path filter would not be effective in managing the leak.

Is filtering the only approach? This thought then leads to some further questions: Can we do a better job without necessarily involving manually-maintained filters? If this is all about the consistency between route advertisements and routing policy, then are the rout registries of use in this context?

Route Registries?

The use of Internet Routing Registries and the associated Routing Policy Specification Language (RPSL) (RFC 2622, RFC40122) is an alternative approach to the manual management of route filters. RPSL is a relatively rich language and, as the name says, it allows a user to describe a network's import and export policies in terms of relationship with adjacent AS's and its transit (re-advertisement) policies.

If this is used in the context of a routing registry it allows a network operator to enumerate the prefixes originated by the local AS and the transit policies that are associated with these routes. It also allows the network operator to describe its re-advertisement policies by specifying its AS neighbours and the routing policies applied to routes learned from adjacent ASes.

If every AS maintained an accurate, up-to-date and complete set of prefix and route policy entries in an Internet Routing Registry, then it appears that it would be theoretically possible for an AS to generate a prefix and AS path filter set for all of its network adjacencies through a computation across the registry's contents. Indeed there are tools that attempt to do precisely that for the existing route registries.

Why aren't we all doing precisely this? Why aren't we using these route registry tools as part of our standard operating practice?

The story about the use of route registries is a very mixed one.

They have been around for almost twenty years now in one form or another, and some regions of the world have been very diligent in compelling every network operator in their region to maintain accurate information in their local routing registry. But in other cases the route registry story is not so encouraging.

RPSL is a complex language and it can be challenging to accurately describe the intricacy of some routing policies in RPSL. Its often the case that the registry is populated with "just in case" entries, as well as historic entries, so sorting out what is current routing intention from other extraneous data in the registry is extremely difficult, and to do so with an automated registry scanning tool has proved not to be possible so far. Its also the case that network operators often use a level of granularity of each eBGP session between adjacent ASes, while RPSL uses a coarser level of granularity of individual ASes. It is therefore more challenging to describe the individual routing policies that apply to each BGP session between the same two ASes, and there is also the question as to whether network operators

would be comfortable in publishing such a detailed level of information about their network's routing policies.

The route registries we use today have various models of authenticity and integrity. It's possible in many cases for a registry user to enter routing information for third party prefixes without the authority of the actual prefix holder. Sorting out what is recognisable as authoritative information from what is not authoritative is not helped by a registry data model that typically includes no validation or authority information. There are also many route registries, and its often the case that they contain conflicting information. Which registry should be "preferred" if one wanted to resolve these contradictions in information? Why?

> A NANOG presentation from October 2008 is still once of the better summaries of the problems we face with route registries. I do not believe that much has change in the five years since this presentation was given.
> (http://www.nanog.org/meetings/nanog44/presentations/Tuesday/RAS_irrdata_N44.pdf)

This would be challenging enough, but the problem is further compounded by the observation that in many areas of the Internet operators have eschewed the route registry approach and rely on their own customised tools. So not only is the quality of the information in route registries variable, the coverage of the information in route registries is also variable.

Where to from here?

This MITM attack form exposes a broader issue here about the difference between routing intent and routing protocol operational correctness. A protocol correctness tool, such as secure BGP, is able to tell you that the routing information has been faithfully propagated across the network via the operation of the routing protocol, but such a tool cannot tell you whether the routes that are being propagated were intentionally distributed or not.

But devising solutions based on routing intentions, along the lines of the route registry has its own failings, as we've examined.

Some longstanding problems are longstanding because we have not quite managed to apply the appropriate analytical approach to the problem. in other words, for some problems, there is a solution out there, but it involves some searching!

We could try, yet again, to coerce the industry to diligently use route registries for all external routing, but what would be different from this call to use route registries from all the other calls in the past? And if its no different, then why would such a call enjoy any greater levels of take up than has happened in the past?

Maybe we could use digital signatures and the Resource PKI (RPKI) to combine information authenticity with the route registries. However the issue we may want to consider in this case is would this only make an already complex and difficult system yet more complex and even harder to use?

Maybe we could go back to using prefix route filters, and attempt automation of the filter function through using the RPKI to validate signed filter requests. But this approach relies on universal adoption, or at least very widespread adoption of prefix filters in order to be effect. And this falls into the category of "good housekeeping" tasks that we all talk about, and many of us actually do, but many others do not. Like BCP 38 it seems that it never quite gets adopted in enough places to be effective.

Maybe this particular problem of support a secure routing environment is a different kind of longstanding problem. Because some problems are longstanding problems simply because they are just exceptionally hard problems!

What we would like is some form of automated mechanism that would allow a BGP speaker to detect the difference in routing updates between what is intended and what is not. and so far what we have is a set of "part" tools that can perform this type of discrimination between good and bad part of the time for part of the possible set of updates for part of the set of routed networks. However, we seem to find it exceptionally hard to figure out what form of approach would offer us some form of completeness.

This makes me wonder if there are alternate perspectives on the space we are working in. For example, would we think about this problem differently if we were to think about routing not as a topology and reachability tool, but an distributed algorithm to solve a set of simultaneous equations. The equations here are expressions of routing policies, and the aim of the algorithm is to converge on solutions that solve individual equations as well as converging on a network-wide solution of maximal connectivity and minimal cost. Would such a different perspective provide a different insight as to the way in which routing policies and routing protocols interact? And could such a perspective provide some leads as to how we could not only secure the routing system against deliberate abuse and malfeasance but also secure it against inadvertent misadventure in the form of route leaks?

Disclaimer

About the Author

Geoff Huston B.Sc., M.Sc., is the Chief Scientist at APNIC, the Regional Internet Registry serving the Asia Pacific region. He has been closely involved with the development of the Internet for many years, particularly within Australia, where he was responsible for the initial build of the Internet within the Australian academic and research sector. He is author of a number of Internet-related books, and was a member of the Internet Architecture Board from 1999 until 2005, and served on the Board of Trustees of the Internet Society from 1992 until 2001.

www.potaroo.net