October 2023
Geoff Huston

Notes from NANOG 89: BGP Error Handling

The original specification of the BGP routing protocol, RFC 1105, from 1989, has the following directive: "NOTIFICATION messages are sent when an error condition is detected. The BGP connection is closed shortly after sending the notification message." Ahh, you might think, that might be a potential problem, but the directive persisted for many years through successive generations of the BGP protocol specification. RFC 4271, from January 2006, still contains the same text. If you tried to pass a route object that contained an error into the routing system, such as a malformed attribute, then the first BGP speaker that processed the update would shut down the BGP session. However, not every route attribute is processed by every BGP speaker. There are attributes that are classified as *transitive opaque attributes*, where a BGP speaker will pass on an attribute to adjacent BGP speakers even if it is not configured to recognise the attribute itself.

What happens if just one implementation of BGP "recognises" a particular transitive attribute and all other implementations do not? (This is not unusual in BGP, by the way. During the transition from two-byte to four-byte AS numbers BGP used a combination of translation and tunnelling to pass the four-byte AS Path across a sequence of two-byte BGP speakers, and the tunnelling component was implemented through transitive opaque attributes.) The result is that only this subset of BGP routers who recognise the attribute will look "inside" this attribute. Now, what if this attribute is internally malformed? Well, this would mean that those BGP speakers who recognise this attribute will drop the sessions which passed the erroneous attribute, even if the sender had no idea what it was passing onward.

Now let's take this same scenario and apply it to the Internet, and let's suppose that the set of BGP speakers that will process this attribute by terminating the session are deployed at the edges of the Internet, rather than in the transit core. In the worst case this malformed update will now cause all these BGP speakers to disconnect themselves from their transit paths to the rest of the network, as this was the path used to pass the malformed attribute to the BGP speaker in the first place (Figure 1).

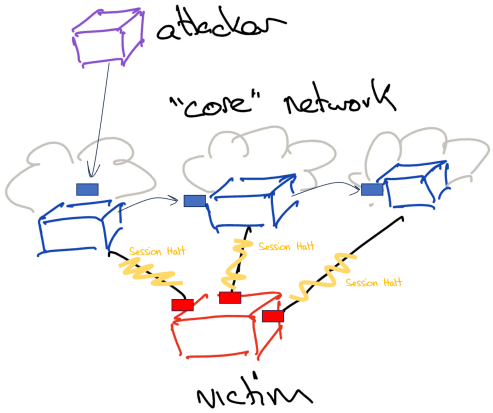

Figure 1 – Attack Scenario exploiting BGP Error Handling

If the session is restored, then the route object with the malformed update will be passed across once more, and session will be bought down again. This cycle will continue until the route object is removed from the route set.

It was not until RFC 7606, from August 2015, that a revision to BGP error handling behaviour was published. As this document explains: "The goal for revising the error handling for UPDATE messages is to minimize the impact on routing by a malformed UPDATE message while maintaining protocol correctness to the extent possible. This can be achieved largely by maintaining the established session and keeping the valid routes exchanged but removing the routes carried in the malformed UPDATE message from the routing system." RFC 7606 indicates that a BGP engine, when encountering an attribute error, may simply ignore the errored attribute and process the rest of the update. This is the case when the malformed attribute is recognised by the local BGP engine to the extent that it can determine that the attribute itself has no effect on route selection and route installation even were it to be well formed. The next level of BGP response is to treat the update containing a malformed attribute as an implicit withdraw, and just not process the entire update. There are two further levels of escalation in error response in RFC 7606. The first is to disable the address family from further processing in that session, by ignoring this and all subsequent updates with the same AFI/SAFI as that used in the malformed update. And the ultimate response is to perform a session termination. (The RFC refers to this as both as a session "reset" and a session "termination" as these are interchangeable terms, which strikes me as a poor use of the term "reset". For me "reset" means erase the current state of the session and restart it from scratch.) This revision to BGP is now almost 10 years ago. We've fixed this. Right?

As Ben Cartwright-Cox described in a recent presentation at NANOG 89 a small Brazilian network (AS264366, Evaldo Sousa Carvalho-ME) originated a BGP route object that contained an attribute that was still in the process of being incorporated into the standard BGP specification (the BGP Entropy Label Capability Attribute). Unsurprisingly, many deployed routers did not understand this attribute value, so they simply passed in onward as part of standard BGP propagation of opaque transitive attribute handling. However, Juniper's JUNOS platforms had some level of support for this attribute, but it appears that the implementation in deployed versions of JUNOS were incomplete, or incompatible with the attribute contained in this update. While JUNOS recognised the attribute in a BGP update, the update was flagged as an error.

In and of itself this should not be a disaster. In any case the compounding problem for the JUNOS implementation is that they evidently hadn't received this 8-year-old RFC7606 message about graceful error recovery options and they performed a session termination. It's hard to be sympathetic here about Juniper's BGP implementation. It's not as if BGP is as convoluted as the DNS, and it's not as if the RFC set for BGP is constantly growing on a weekly basis. Assuming that a major router vendor has a BGP implementation that includes conformance to 8-year-old Standards Track RFCs is an entirely warranted assumption on the part of any network operator.

We now have an "interesting" set of circumstances here. BGP implementation is deployed over a lot of the Internet, but not so widely that it dominates the deployment on all transit routes. And this is an implementation that performs a full session termination whenever it encounters a particular update for an individual malformed route object. At this point the injection of a flawed update can effectively isolate large parts of the network.

Juniper have reportedly applied a patch to JUNOS to correct this behaviour (JSA72510). There is also a more general commentary on attribute propagation and the issue of whether or not to propagate an update containing an unrecognised attribute. As the commentary draft points out "This document highlights properties of the BGP protocol and situations where its defined behavior for propagating Path Attributes may lead to inadvertent disclosure of information, improper routing, or even session resets and crashes. Such behaviors can be maliciously exploited."

In any case, this leads to the obvious followup question: Just how well do other BGP implementations cope with malformed attributes?

BGP Path attributes are enumerated in a IANA protocol parameter registry. BGP Attribute types use an 8-bit wide field and of the 256 possible values some 32 are assigned, 13 are deprecated, 2 codes are reserved and the remaining 209 are unassigned at present.

The question here is, what happens when the implementation is presented with any of the 256 possible attribute codes? Ben has tested a number of BGP implementations in this way and has written up his experiences in his blog as well as his NANOG 89 presentation.

BGP implementations from MicroTik, Ubiquiti, Arista, Cisco IOS-XE and IOS-XR had no observed errors in his tests. JUNOS did encounter the issues already described, which can be mitigated with the "bgp-error-tolerance" configuration objective. Nokia SR-OS defines "update-fault-tolerance" as an error handling directive to avoid session termination. A number of other implementations, including FRR, Pica8, SONIC, and OpenBGPd have released fixes for this problem.

Ben has called out Extreme EXOS as an outlier here. Session termination was observed to occur on Attribute 21 (AS_PATHLIMIT) and Attribute 25 (IPv6 Address Specific Extended Community) and Extreme apparently have no configuration option to mitigate this behaviour of their BGP implementation.

Extensible protocols always present issues for BGP vendors. New features may interfere in unforeseen ways with existing behaviours and so these features often take time to be integrated into the code base. But even so, it does seem surprising that some implementations are based on code that does not conform to behaviours defined in an 8-year-old RFC. It seems to be a poor outcome where it is left to the end user to detect such instances of non-conformance, and even poorer when the vendor response indicates no plans to remedy the situation.

There are no BGP protocol police and no third-party agency that tests BGP implementations for standards-conformant behaviour in an exhaustive manner. It's not even clear what would constitute such a collection of exhaustive tests that would stretch a BGP limitation through every possible case. It may have been acceptable a few decades ago to release Internet products on a "suck it and see" basis, but using the network itself as the beta testers for product quality in today's Internet seems to be entirely irresponsible. This is now all that we have for communications, and when the Internet breaks these days then it's broken for everyone.

Other industries have had to adapt to increasingly stringent requirements for robustness testing prior to release, and the airline industry is perhaps a good example of both such processes and their rationale. The Internet has managed to evade such requirements since its inception. The market-driven model that underlies much of the Internet's technology base does not necessarily properly account for risk and as a result we undervalue robustness of products.

How we might want to respond this, and how we might want to structure incentives in this industry to increase the level of investment in the quality and robustness of the products we all rely on, such as the BGP protocol and its implementations, remains an open question for this industry. We are acutely aware that tolerating such vulnerabilities is tantamount to tolerating the addition of yet another DOS attack vector to already brimming bucket of DOS attack vectors, and this is not acceptable. But how, and who, should be co-opted to work in this space and how we should resource this activity remains part of the set of open issues in this space. Is it a case of increasing the level of requirements in applicable regulations? Or increasing the liabilities for the consequences of faulty products? Or the use of compliance certification with independent test laboratories? I suspect it's not a case of a paucity of potential methods to achieve this, but a marked reluctance by individual industry actors to take the first step here. A cautious option is to do what everyone else does, even if what they are doing is simply not good enough!

I really hesitate to say that this is yet another instance of an Internet Governance issue, but in other industries where their market was heading in a downward spiral where product price was more important than product quality and safety, a common response was regulatory intervention through the definition

of minimum levels of acceptable quality in these fundamental technologies. The routing space, and BGP in particular, could benefit from the imposition of such minimum standards of quality and robustness of routing protocol implementations.

Ben's slides and a recording of his presentation are on the NANOG 89 web site.

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