Bongo: A BGP Speaker Built for Defending Against Bad Routes

Kevin Benton
Indiana University
School of Informatics and Computing
Bloomington, IN, USA
KTBenton@Indiana.edu

Dr. L. Jean Camp
Indiana University
School of Informatics and Computing
Bloomington, IN, USA
LJCamp@Indiana.edu

Dr. Martin Swany
Indiana University
School of Informatics and Computing
Bloomington, IN, USA
Swany@Indiana.edu

Dec 2015

Abstract

Hijacks, outages, route leaks, and AS path spoofing are cases where network operators may want to influence the way routes are accepted and propagated from BGP neighbors in ways not supported by traditional BGP speakers. In this paper, we introduce *Bongo*, a software-based BGP speaker than can selectively filter out or extend the path of BGP updates received from other peers based on arbitrary operator-defined policies. Additionally, we show how the modularity of this system makes it easy to integrate with existing routers as well as other network devices such as OpenFlow switches or firewalls.

1 Introduction

The Border Gateway Protocol (BGP) is responsible for the distribution of routes between autonomous systems on the Internet. The current version of the protocol (BGPv4) was first published in 1995 [19] and the most recent revisions were

published in 2006 [20]. Due to the trusted nature of the Internet at the time of the protocol's creation, BGP participants were assumed to behave well and were trusted to only advertise routes to networks they owned or could reach. Therefore, the protocol contains no protection against BGP participants from advertising false routes.

The trusting nature of BGP means that simple misconfigurations can take down large chunks of the Internet. In June of 2015, ISP *Telekom Malaysia* advertised preferable paths to 179,000 prefixes to the Tier 1 ISP *Level 3*, which redistributed them widely resulting in a major portion of the Internet traffic in Asia routing through, and saturating, Telekom Malaysia's network [1]. While these route leaks are quickly identified as errors by network operators, the standard BGP protocol does nothing to stop it.

In addition to route leaks, there are more focused outages caused by specific prefixes being advertised by ISPs which do not actually own them (a.k.a prefix hijacking). A famous example of this is when Pakistan Telecom acted on a government order to block YouTube in the country by setting up a blackhole route [1] for YouTube's prefixes. While this is a common method of blocking traffic to certain addresses, they made the mistake of redistributing the route to their BGP peers outside of the country, which caused a YouTube outage around the world. [14]

In the two cases above, the result of the misconfiguration was immediately noticeable to network operators so steps could be quickly taken to manually intervene (e.g. drop BGP peering sessions with broken neighbors). However, when prefixes are hijacked in an intentional and targeted manner, attacks can go unnoticed for weeks and/or until well after they have finished.

In one case, an attacker successfully used a Canadian ISP to hijack prefixes to control-nodes of Bitcoin[2] miners just long enough to issue commands to the miners to change to a different control-node. With the control of the miners, the attacker was able to steal about $83,000 of Bitcoin. Due to the short nature of the hijacks, none of the operators for the hijacked networks (mainly hosting providers) even noticed the attacks. [2]

The final type of attack is more subtle than short hijacks like the Bitcoin attack above. Instead of the attacker hijacking the prefix in a way that causes all ISPs to send traffic destined for the prefix to the attacker, the attacker only advertises the bad prefix out one of its peer links while maintaining the valid route via another link. This forces the traffic to travel to the attacker like a normal hijack. The attacker can then forward it via the valid route to its original destination after recording or making modifications to the traffic. This type of attack was observed to force traffic between two ISPs in Denver to travel all of the way via the UK and Iceland. [12, 10]

The goal of our work is to present a system, Bongo, that can selectively filter out or extend the path of BGP updates received from other peers based on arbitrary operator-defined policies. (Following the tradition of Quagga, we named

[1] A blackhole route is a route that just drops packets which match it.
[2] Bitcoin is a digital currency

this systems after an ungulate: Bongo. Our proposal adds defensive features; analogously a Bongo has horns but a Quagga does not.) Before introducing Bongo, we discuss research on detecting bad routes. Any of the detection mechanisms could be used with Bongo; it is agnostic about its data sources. After describing the efforts in route detection, then filtering, we describe the architecture of Bongo. We discuss the performance in the lab, and close with future work.

1.1 Detecting Bad Routes

There has been a significant body of work dedicated to detecting prefix hijacking and route leaks. Most use either dataplane observations, control plane observations, or a combination of the two.

Zhang et al. proposed a system based purely on dataplane observations by sending traceroute probes from the network containing the prefix in question and observing variances in the return path [22]. This approach only detects unexpected changes for the network running the probes. So global detection would need require that the system be deployed at every prefix origin.

Hiran et al. proposed a crowd-sourced approach to measuring dataplane round-trip-time anomalies to different addresses caused by routing updates [8]. However, it didn't take into account the common practice of BGP anycast for large services that legitimately announce routes from many locations, leading to a wide RTT variance that looks the same as route hijacks. This system alone may not be enough to identify route leaks.

Qiu et al. proposed an approach based purely on observing changes in the BGP topology over time [17] and flagging previously unseen topologies that violate known routing policies as suspicious. A disadvantage to this approach is that it had to be augmented with heuristics to try to account for legitimate updates that were being incorrectly classified.

Qui et al. proposed a distributed monitoring system that would monitor route updates on the control-plane and would send traffic on the dataplane to monitored prefixes to locate the origin of a hijack [18]. However, the active probing of the dataplane in this case would become expensive if every prefix was monitored.

Hu et al. proposed a system based on route update collection and data plane fingerprinting [9] based on checking responses to probes sent to endpoints in each prefix. This approach requires an expensive amount of network traffic and processing to probe every prefix on the Internet.

Chang et al. proposed a reputation system called AS-CRED that tracks which ASes have bad BGP behavior (e.g. route leaks and rapid route announcement withdrawals) and they show that ASes tend to have repetitive behavior that makes their reputation system a useful indicator [4]. While this is not a method to detect route leaks in itself, it provides a useful historical aggregation method for leaks detected by other methods that can be used as a feedback mechanism.

McArthur et al. have shown that even with many of the proposed detection systems, hijacks limited in scope will still go undetected [15], indicating there is still more room for improvement.

In addition to these there is a plethora of sources of reputation. The original real time black hole list by Spamhaus continues as lists of exploits and malicious IP addresses which can be used to create AS reputation. Other providers of blocking information include APWG, anti-virus vendors, Microsoft, Google, and network operators associations.

1.2 Filtering Bad Routes

Once suspicious routes have been identified, the next open research problem is determining what actions can be taken to prevent these from harming the network.

Zhang et al. showed that once bad routes are identified, as few as 20 well-connected ISPs can reduce a hijacks impact from 50% of the Internet down to as low as 25% by refusing to propagate the bad route [21]. This shows that even a few participants in a hijack prevention scheme can provide significant benefit to the Internet.

Karlin et al. have proposed a system that delays the acceptance of routes that change the origin of a prefix if the original route is still being announced [11]. One of the issues with this short-term historical approach is that it can delay legitimate updates where a prefix owner legitimately started announcing from another provider due to capacity problems (e.g. in response to a DDoS).

Gersch et al. proposed a system that stores information about where prefixes belong in reverse DNS records for IP addresses in the prefix [7]. Then any updates that violate these constraints should be dropped.

Qi et al. proposed a system where all routers would perform attestations on neighboring routing software before accepting routes from them [13]. However, this requires deployment inside the routing software of many core routers so it's not likely a solution that can get widespread adoption.

The description of the architecture below illustrates that Bongo provides the flexibility to provide different responses to different levels of suspicion and different types of risk. The responses can vary by the sensitivity of the network operator and the risk itself. For example, with Bongo, a small network of local attestations could be used as an overlay so that [13] could be used in a limited manner. Building on [11] an operator may choose route padding, so that the original route would still be used but another would be diffused and available should it be needed. Conversely, should suspicion prove founded, the route may be dropped without the the disruption of it having been widely deployed.

2 Bongo Architecture

Bongo is based on the open source ExaBGP [5]. Figure 1 shows the overall architecture of Bongo and its integration with other network devices. The main

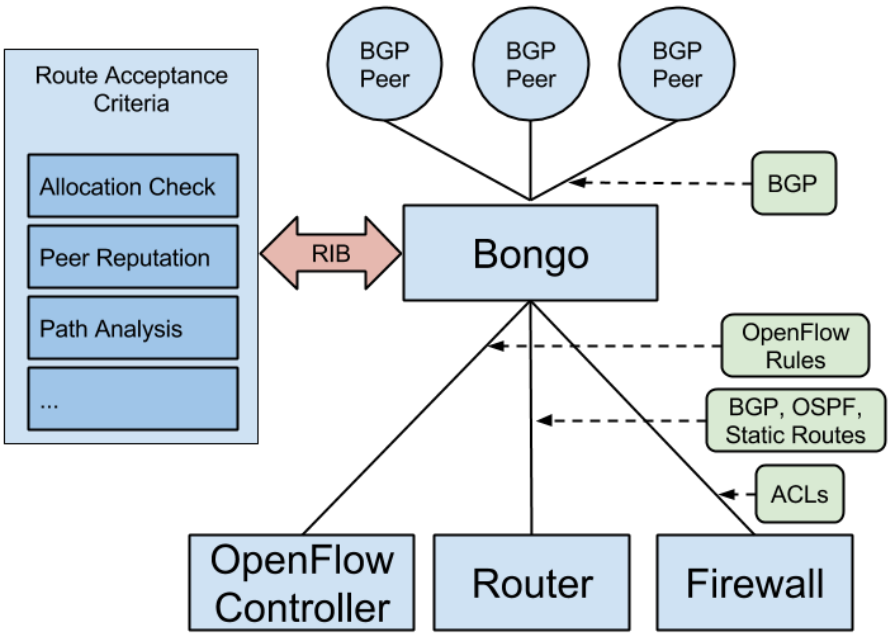

Figure 1: Bongo High-Level Architecture

component that Bongo adds beyond the capabilities of ExaBGP is the route acceptance engine. As routes are received, they are passed into the acceptance engine, which will apply any arbitrary logic we want to leverage to determine the action that should be taken for the route.

We currently have implemented three actions. The first is to simply drop the update so it neither populates the forwarding information base nor propagates to other peers. The second is to extend the path of the update so much that it would be the least preferable route to that prefix. The final one is to make a call to an external program with the details of the suspicious route so some other action could be taken. Our model of the third action was that the route would be flagged and delayed. That is, the route would be delayed and immediately flagged for manual review and intervention by a network operator. After some delay, if no other action is taken then the route would be diffused.

Writing the filtering modules for Bongo is relatively straight-forward. The filtering module was implemented as a python module that must implement a single filter function that receives a route and a reference to the current topology. The filter can then perform whatever arbitrary calculations and lookups are necessary and then it is then expected to return one of the 3 actions outlined

above.

The placement of Bongo is like any other route reflector or BGP speaker. It must be configured to peer with neighboring routers using standard BGP peering configurations. In addition to regular BGP peering, several other methods of installing forwarding entries can be leveraged, which we discuss in the next section.

2.1 Dataplane Forwarding Control

One of the advantages of building Bongo on top of ExaBGP is that the flexibility it offers in what to do with the resulting forwarding information base (FIB). Figure 1 shows three uses: interactive with SDN flows, integrating with standard BGP, and blocking certain types of traffic with a firewall.

The simple case is integrating with a normal BGP-speaking router. To enable this, the router can be configured as a standard BGP peer to Bongo like the upstream routers. In this case Bongo will just be acting as route reflector but with defensive capabilities[3].

ExaBGP also offers simple configuration options to call external software with the route updates so forwarding or filtering entries can be setup using varying underlying technologies. We have leveraged this approach in other work to implement feasible path reverse path forwarding [6] using OpenFlow [16] switches based on the known valid routes. In that case, we found that focusing on Tier 2 providers and dropping paths in SDN could have a high degree of efficacy in defeating amplification attacks [3]. Here we present the generalizable architecture of which that was a specific application.

In addition to controlling the forwarding behavior of the underlying infrastructure, the knowledge about bad routes can be used to generate filtering rules to protect sensitive devices from sending traffic down bad routes. If Bongo is placed in a location at the edge of the network where it can recognize a bad route but it cannot prevent it from propagating, the identified bad routes can be converted into ACL entries for a firewall that prevent traffic destined to any of the prefixes affected by the hijack. While this will result in a loss of connectivity, it will prevent sensitive data from being sent to bad actors. As the Bongo architecture allows for any set of rules, organizations can select particularly sensitive domains where confidentiality outweighs availability. The resulting rules from Bongo could limit the risk of exposure for those machines or subnetworks while allowing communication that is either less sensitive in terms of confidentiality or more timely (in terms of availability) to continue as usual.

2.2 Performance

The performance of Bongo is bound by two factors: the performance of the underlying ExaBGP framework and the types of filters in the route acceptance engine.

[3] A Bongo has horns but a Quagga does not

The underlying ExaBGP framework is performant enough to operate at Internet exchange points, where it can process the global Internet IPv4 and IPv6 routing table from 6 different peers in just over 3 minutes[4].

In local testing, the performance degrades significantly if the filters we have defined in the route acceptance engine have to do many database lookups. In one case, it took 5 hours to process approximately 3 million updates (full routing table * 6 peers). While this performance is poor during the initial peering phase, it's still fast enough to handle the updates during normal BGP topology changes [5]).

In order to improve the performance, we are working on adding an asynchronous filtering mode where we allow ExaBGP to participate at its full speed and then examine the forwarding information base it generates with the Bongo route acceptance engine in a separate task. Then whenever it identifies a bad route, Bongo informs ExaBGP to withdraw or modify the route.

Adding the asynchronous model will give us a trade-off between full performance with small windows where bad routes may be accepted versus slower performance where every route is examined as it arrives.

3 Future Work

3.1 Exploring Additional Indicators for Bad Routes

While there has been a good body of work identifying bad routes purely from control plane observations (e.g. AS topology history) and dataplane measurement (e.g. RTT observations to destinations), there has been a limited amount of work annotating BGP updates with additional externally generated indicators.

We are evaluating the addition of economic data about the country the ISP operates in to look for correlations between the strength of the country's tech sector and the propensity for being an originator of route leaks and prefix hijacking. Additionally, each government's policy on blocking entire networks for the purpose of censorship may have a correlation with route leaks.

We are also examining indicators based on the history of the autonomous systems. These include the type of organization operating the AS (e.g. ISP vs. hosting provider) as well as cross-references to other lists that indicate poor Internet participant behavior (e.g. known to be a heavy source of spam).

By building a holistic reputation of both the autonomous systems and the prefixes they originate based on the indicators outlined above, we believe we can improve the accuracy of additional detection systems based purely on topology analysis.

[4] https://gixtools.net/gix/gix-route-collector-performance/

[5] 10s per second see https://labs.ripe.net/Members/vastur/the-shape-of-a-bgp-update

3.2 Impact of Filtering Routes

One of the major concerns with dropping route updates or modifying them based on arbitrary policy decisions setup by individual network operators is that it could impact the stability and resiliency of BGP. Previous work has leveraged simulations to estimate the impact of route manipulation [21, 11] and we intend to do the same.

Additionally, we will simulate strategies that are less invasive than completely dropping updates. For example, if a shorter path to a prefix is advertised via a path that is identified as suspect, the path could be significantly extended to prevent any downstream peers from leveraging it unless all other routes are retracted.

4 Conclusions

Detection of a malicious route is now as much art as science. Dedicated professionals identify leaks and hijacks as soon as possible. This is all done while routes are accepted and diffused across the network. Yet despite dedicated cores of engineers, the current approach allows route anomalies and then recovers from them. By combining different reputation and risk metrics, Bongo allows institutions to slow down changes and examine them before accepting the risks that are most concerning. Because different organizations have different concerns, for example fraud in banking versus classified materials in the military, Bongo allows each organization to fine tune its own concerns without requiring any changes in or additional information from the network as a whole. By providing an open API to respond at the flow or firewall level, Bongo can support organizations where only a small part is highly sensitive. Bongo approaches route updates as risk decisions, and can estimates the risk of adopting a route based on the any filter the programmer deems appropriate. Bongo allows an organization to decide exactly how much and what kind of risk it wants to accept from the control plane.

5 Acknowledgments

Research was sponsored by the Army Research Laboratory under Cooperative Agreement Number W911NF-13-2-0045 (ARL Cyber Security CRA). This material is based upon work supported, in part, by the DHS BAA 11-02-TTA 03-0107 Contract N66001-12-C-0137, Cisco Research Support Proposal 591000, and the Google Privacy & Security Focused Research Program. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the DHS, DoD, Google, Cisco, or Indiana University. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research

References

[1] B. Andree Toonk. Massive route leak causes internet slowdown. `http://www.bgpmon.net/massive-route-leak-cause-internet-slowdown/`, 2015.

[2] W. Andy Greenberg. Hacker redirects traffic from 19 internet providers to steal bitcoins. `http://www.wired.com/2014/08/isp-bitcoin-theft/`, 2014.

[3] K. Benton, L. J. Camp, T. Kelley, and M. Swany. Filtering ip source spoofing using feasible path reverse path forwarding with sdn. In *5th International Conference on Communication and Network Security*. IEEE, 2015.

[4] J. Chang, K. K. Venkatasubramanian, A. G. West, S. Kannan, I. Lee, B. T. Loo, and O. Sokolsky. As-cred: Reputation and alert service for interdomain routing. *Systems Journal, IEEE*, 7(3):396–409, 2013.

[5] Exa-Networks. exabgp. `https://github.com/Exa-Networks/exabgp`, 2015.

[6] F. Baker and P. Savola. Ingress Filtering for Multihomed Networks. RFC 3704, IETF, March 2004.

[7] J. Gersch and D. Massey. Rover: Route origin verification using dns. In *Computer Communications and Networks (ICCCN), 2013 22nd International Conference on*, pages 1–9. IEEE, 2013.

[8] R. Hiran, N. Carlsson, and N. Shahmehri. Crowd-based detection of routing anomalies on the internet. 2015.

[9] X. Hu and Z. M. Mao. Accurate real-time identification of ip prefix hijacking. In *Security and Privacy, 2007. SP'07. IEEE Symposium on*, pages 3–17. IEEE, 2007.

[10] R. Jim Cowie. The new threat: Targeted internet traffic misdirection - dyn research. `http://research.dyn.com/2013/11/mitm-internet-hijacking/`, 2013.

[11] J. Karlin, S. Forrest, and J. Rexford. Pretty good bgp: Improving bgp by cautiously adopting routes. In *Network Protocols, 2006. ICNP'06. Proceedings of the 2006 14th IEEE International Conference on*, pages 290–299. IEEE, 2006.

[12] W. Kim Zetter. Someones been siphoning data through a huge security hole in the internet. `http://www.wired.com/2013/12/bgp-hijacking-belarus-iceland`, 2013.

[13] Q. Li, M. Xu, J. Wu, X. Zhang, P. P. Lee, and K. Xu. Enhancing the trust of internet routing with lightweight route attestation. *Information Forensics and Security, IEEE Transactions on*, 7(2):691–703, 2012.

[14] R. Martin Brown. Pakistan hijacks youtube. `http://research.dyn.com/2008/02/pakistan-hijacks-youtube-1/`, 2008.

[15] C. McArthur and M. Guirguis. Stealthy ip prefix hijacking: don't bite off more than you can chew. In *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, pages 1–6. IEEE, 2009.

[16] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. Openflow: enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, 38(2):69–74, 2008.

[17] J. Qiu, L. Gao, S. Ranjan, and A. Nucci. Detecting bogus bgp route information: Going beyond prefix hijacking. In *Security and Privacy in Communications Networks and the Workshops, 2007. SecureComm 2007. Third International Conference on*, pages 381–390. IEEE, 2007.

[18] T. Qiu, L. Ji, D. Pei, J. Wang, J. J. Xu, and H. Ballani. Locating prefix hijackers using lock. In *USENIX Security Symposium*, pages 135–150, 2009.

[19] Y. Rekhter and T. Li. A Border Gateway Protocol 4 (BGP-4). RFC 1654, IETF, July 1995.

[20] Yakov Rekhter, Tony Li, and S. Hares. A Border Gateway Protocol 4 (BGP-4). RFC 4271, IETF, January 2006.

[21] Z. Zhang, Y. Zhang, Y. C. Hu, and Z. M. Mao. Practical defenses against bgp prefix hijacking. In *Proceedings of the 2007 ACM CoNEXT conference*, page 3. ACM, 2007.

[22] Z. Zhang, Y. Zhang, Y. C. Hu, Z. M. Mao, and R. Bush. Ispy: detecting ip prefix hijacking on my own. In *ACM SIGCOMM Computer Communication Review*, volume 38, pages 327–338. ACM, 2008.