Poster: Smart BGP Hijacks that Evade Public Route Collectors

Alexandros Milolidakis
KTH Royal Institute of Technology

Tobias Bühler
ETH Zurich

Marco Chiesa
KTH Royal Institute of Technology

Laurent Vanbever
ETH Zurich

Stefano Vissicchio
University College London

ABSTRACT

Routing hijack attacks have plagued the Internet for decades. Internet-wide BGP monitoring infrastructures have recently received great attention as they promise to quickly *detect* hijack attacks and, ultimately, mitigate them.

In this poster, we investigate the robustness of monitor-based detection systems with respect to so-called "smart" attackers who engineer their hijacks to evade detection. Our preliminary simulations show that monitor-based systems may be unable to detect many carefully crafted hijacks diverting traffic from thousands of ASes.

1 INTRODUCTION

Routing hijacks are common on the Internet. The Internet consists of thousands of Autonomous Systems (ASes) deployed in multiple geographic regions. The Border Gateway Protocol (BGP) allows ASes to advertise IP prefixes to their neighbors as well as routes towards the IP prefixes of other networks. In this way, networks in different regions learn how to forward traffic to a destination prefix.

Yet, BGP has not been designed with security in mind. Any network can advertise routes containing bogus information (*e.g.*, reachability of any IP prefix). These attacks, known as BGP hijacks, corrupt the routing table entries of networks that trust them, causing them to redirect traffic destined to the victim towards the hijacker. In 2020, over 2000 BGP hijacks were reported, a 30% increase from the year 2019 [3].

To thwart hijacks, networks rely on multiple solutions such as *i)* monitoring routes at "route collector" locations, *ii)* signing BGP information using cryptography, as in RPKI, or *iii)* data-plane signals such as pings and traceroutes. Solutions such as [2, 4, 8, 9] rely on BGP *route collectors*, systems which collect the BGP *best routes* from any network they are peering with. We call these peering networks *BGP monitors* in this poster. Examples are the Public BGP route collector infrastructure, namely RouteViews and RIPE-RIS [6, 7].

In this work, we investigate the effectiveness of today's public route collectors in the face of smart hijackers that react to the deployment of monitor-based systems. Hijackers can indeed exploit well-known techniques, such as AS-path poisoning or prepending, to avoid that their BGP messages reach the monitors. These techniques however tend to reduce the hijacks' impact because the attackers' routes become longer and hence generally less preferred by networks without monitors too. Past work [8] has shown using simulations that hijacking events that affect more than 2% of the Internet are *always* visible by the public Public route collectors. In contrast, our simulations show that smart hijackers that *deliberately* react to the location of the monitors can stealthily poison up to 25% of the Internet.

2 ATTACK STRATEGIES

We note that the way malicious routes propagate over the Internet depends on the routing policies of other ASes – something outside the control of the hijacker. Those policies depend on how each AS has configured its local BGP decision process and its ability to filter out illegitimate advertisements (*e.g.*, using RPKI). Yet, even though the hijacker has no control over other's policies, it can still influence how its own illegitimate advertisements propagate by carefully adjusting:

- *The AS-path length of the malicious route:* The second step in the BGP path selection process prefers routes with shorter AS paths. Therefore, a hijacker can craft routes with longer AS paths that are less likely to propagate far and be reported by Route collectors. We classify hijacks as *type-N*, where N denotes the AS-hop distance of the hijacker from the origin in the crafted path [8].
- *The ASes included in the AS-path:* The so-called BGP "loop-prevention mechanism" drops routes containing the ASN of the AS processing it. An attacker can exploit this behavior to prevent the propagation of a route to an AS by adding a specific ASN to the path (AS path poisoning).
- The neighbor to which to export the malicious route: Commonly, only some neighbors cause hijacks to propagate to route collectors. Carefully crafting the best per-neighbor announcement allows the creation of lower type hijacks that can propagate further while avoiding detection.

Smart hijacker's goal: Compute a set of crafted *per-neighbor* BGP messages that maximize the number of poisoned networks while being invisible (stealthy) to the route collectors.

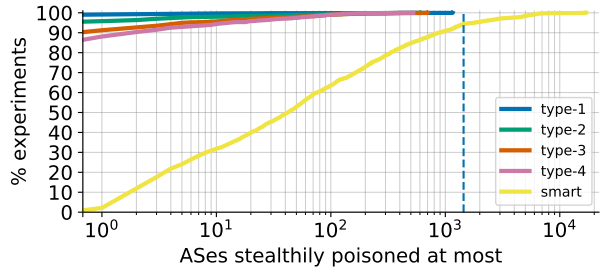

Fig. 1: Smart vs baseline hijacker impact. The dashed line illustrates 2% of the Internet networks.

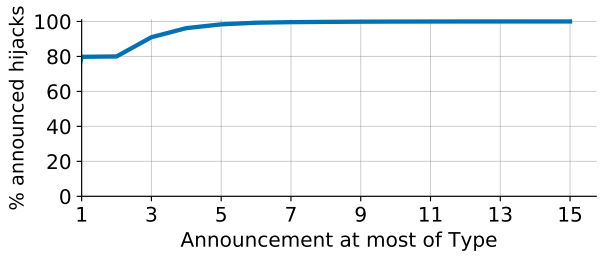

Fig. 2: Per-neighbor announced hijack-types crafted by the smart hijacker.

3 DATASETS

Internet topology and routing policies. We rely on the monthly Internet topology inferred by CAIDA gathered from the RouteViews and RIPE-RIS route collectors [1]. As of July 2021, the topology includes 72K ASes and 509K links each one characterized over its Gao-Rexford relation type (p2p or c2p). This topology has been utilized by multiple papers seeking to simulate the Internet, therefore we also utilize it and assume Gao-Rexford routing policies.

BGP route collectors and monitors. We identify the monitor ASes by using BGPStream [5], observing for which peer ASes of RIPE-RIS and RouteViews the collectors actively report any of their routes. From one day worth of BGP UPDATES (July 1*st*, 2021), we identify 483 unique monitor ASes, 367 actively reporting paths within 5 minutes. We mark those 367 ASes as monitors in our topology. We report our designed smart hijack as stealthy if none of these monitors is poisoned (*i.e.*, the attack does not reach the route collectors).

4 SIMULATION SETUP AND FINDINGS

Selecting victim and hijacker networks. We perform 2000 simulations selecting random pairs of hijacker and victim networks. In each simulation, the victim announces one of its prefixes to all its neighbors. Then, the hijacker stealthily hijacks that prefix by crafting *per-neighbor* hijack announcements based on the strategies of section 2. We compare this smart hijacker with a *baseline* hijacker that always announces the same attack type to all neighbors. When selecting the victim, we verify that it is well-connected and its prefix propagates to more than 95% of the Internet. When selecting the hijacker, we make sure that it is not a direct provider of the victim, with which we assume the victim shares a more trusted relation. Furthermore, we do not select hijackers among the monitor ASes.

Smart hijackers are able to stealthily poison thousand of networks. Figure 1 shows the % of simulations (Y-axis) in which the hijacker was able to stealthily poison at most a certain number of networks (X-axis). As the figure illustrates, along the Y-axis all hijacker types, even baseline ones, could produce a stealthy attack that is not visible to the Public route Collectors. Out of the 2000 simulations, only 1% of Type-1 hijacks were invisible, increasing to 4.5%, 9.7%, and 13.5% for hijacks of Type-2, 3, and 4, respectively. As expected, the lower the hijack type the further the hijack propagates in the topology, and the more likely it is for a monitor to report it. Baseline hijacks could not poison more than 1440 ASes, i.e. 2% of the topology. On the contrary by announcing neighbor-specific hijacks, our smart hijacker was able to almost always generate a stealthy hijack affecting more than 2% of the topology in 5.6% of our simulations. At worst, this increases up to a maximum of 25% *i.e.*, 18K affected networks.

Figure 2 illustrates the neighbor-specific hijack types produced by our smart hijacker (84K in total during the 2000 simulations). As we see, 80% of these hijacks are of type-1 increasing up to type-15 for our largest announcement. We do not announce type-0 attacks since they are not RPKI valid.

As future work, we plan to conduct extensive tests of our attacks in the real-world using the PEERING testbed.

REFERENCES

[1] 2021. The CAIDA AS Relationships Dataset, <2021-07-01>. http://www.caida.org/data/active/as-relationships/.

[2] Cisco. 2021. Cisco BGP alerter. https://bgpstream.com/.

[3] Internet Society. 2020. A Regional Look into BGP Incidents in 2020. https://www.manrs.org/2021/03/a-regional-look-into-bgp-incidents-in-2020/. Accessed: 2021-05-25.

[4] Massimo Candela. 2020. Easy BGP Monitoring with BGPalerter. https://labs.ripe.net/author/massimo_candela/easy-bgp-monitoring-with-bgpalerter/.

[5] Chiara Orsini, Alistair King, Danilo Giordano, Vasileios Giotsas, and Alberto Dainotti. [n.d.]. BGPStream: a software framework for live and historical BGP data analysis. In *IMC'16*.

[6] NCC Ripe. 2006. Routing information service. *http://www. ripe. net/projects/ris/docs/peering. html* (2006).

[7] Oregon RouteViews. 2013. University of oregon routeviews project. *Eugene, OR.[Online]. Available: http://www. routeviews. org* (2013).

[8] Pavlos Sermpezis, Vasileios Kotronis, Petros Gigis, Xenofontas Dimitropoulos, Danilo Cicalese, Alistair King, and Alberto Dainotti. 2018. ARTEMIS: Neutralizing BGP hijacking within a minute. *IEEE/ACM Transactions on Networking* 26, 6 (2018), 2471–2486.

[9] ThousandEyes. 2020. ThousandEyes bgp monitoring service. https://www.thousandeyes.com/solutions/bgp-and-route-monitoring/. Accessed: 2021-05-25.