ARTEMIS: Real-Time Detection and Automatic Mitigation for BGP Prefix Hijacking

Gavriil Chaviaras, Petros Gigis, Pavlos Sermpezis, and Xenofontas Dimitropoulos
FORTH / University of Crete, Greece
{gchaviaras, gkigkis, sermpezis, fontas}@ics.forth.gr

ABSTRACT

Prefix hijacking is a common phenomenon in the Internet that often causes routing problems and economic losses. In this demo, we propose ARTEMIS, a tool that enables network administrators to *detect and mitigate* prefix hijacking incidents, against their own prefixes. ARTEMIS is based on the real-time monitoring of BGP data in the Internet, and software-defined networking (SDN) principles, and can completely mitigate a prefix hijacking within a few minutes (e.g., 5-6mins in our experiments) after it has been launched.

CCS Concepts

•Networks → Network management; Network monitoring; •Security and privacy → *Network security;*

1. INTRODUCTION

The Internet is composed of thousands of Autonomous Systems (ASes), whose inter-domain traffic is routed with the Border Gateway Protocol (BGP). Due to the distributed nature and lack of authorization in BGP, an AS can advertise illegitimate paths or prefixes owned by other ASes, i.e., hijacking their prefixes. Prefix hijacking can cause serious routing problems and economic losses. For instance, YouTube's prefixes were hijacked in 2008 disrupting its services for more than 2 hours [1], whereas China Telecom hijacked 37000 prefixes (about 10% of the BGP table) in 2010 causing routing problems in the whole Internet for several minutes [2].

Prefix hijacking (due to an attack or misconfiguration) is a common phenomenon in the Internet, and since its prevention is not always possible, mechanisms for its detection and mitigation are needed. To this end, several methodologies for detecting prefix hijackings have been proposed, e.g., [3, 4]. However, most previous works focus on *alert systems* that are not controlled by the AS itself [3, 4], but offer BGP prefix hijacking detection as a service to ASes. In addition, previous research focuses primarily on *accurately* detecting BGP hijacks, rather than *timely* detecting *and* mitigating them. The whole detection/mitigation cycle presently has significant delay: (i) aggregated BGP data from RouteViews [5] or RIPE RIS [6], which are commonly-used for detection, become available approximately every 2 hours (BGP full RIBs) or 15mins (BGP updates); (ii) a network administrator that receives a notification from a third-party alert system needs to *manually* process it to verify if the notification corresponds to a hijacking or is a false alarm; and (iii) for mitigation, administrators often need to manually reconfigure routers or contact administrators of other ASes to filter announcements. YouTube, for example, reacted about 80min after the hijacking of its prefixes. These problems render existing mechanisms inefficient especially for a large percentage of hijacking events that last only for a short time (cf., more than 20% of hijacks last $<$ 10mins [3]).

In this work, our goal is to enable network administrators to *timely* detect and mitigate prefix hijacking incidents, e.g., in 5-6 mins, against their *own* prefixes. To accelerate detection, our approach exploits real-time BGP data from: (i) Looking Glass (LG) servers; and (ii) BGP collectors with live data streaming capabilities, which are provided by the RIPE RIS [6, 7] and BGPmon [8] projects. LGs provide a view directly from operational BGP routers, without intermediate collectors, while (the recent) RIPE RIS streaming service [7] and BGPmon [8] provide real-time feeds of the collected BGP data. Furthermore, we *automatically* mitigate hijackings of prefixes owned by an AS by announcing de-aggregated BGP prefixes. We combine these in a tool we call ARTEMIS (*Automatic and Real-Time dEtection and MItigation System*), which can detect a prefix hijack in near real-time, and mitigate it without any manual intervention.

We evaluated ARTEMIS in real settings, by deploying it to detect and mitigate prefix hijackings performed against our own prefixes from an actual AS in the Internet. We found that we can detect hijacks in <1min, start the mitigation in a few seconds, and completely solve the problem in around 5mins. To our best knowledge, this is the first time that we can *detect and mitigate* hijacks within a few minutes.

2. ARTEMIS OVERVIEW

ARTEMIS consists of three components: a *detection*, a

SIGCOMM '16, August 22–26, 2016, Florianopolis, Brazil

DOI: http://dx.doi.org/10.1145/2934872.2959078

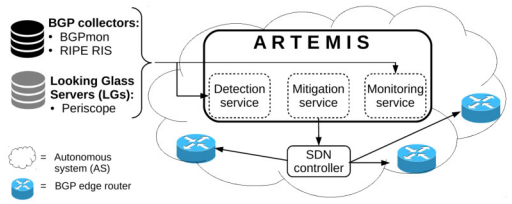

Figure 1: ARTEMIS overview.

mitigation, and a *monitoring* service as shown in Fig. 1. The detection service runs continuously and combines *control plane* information from Periscope [9] (an LG API), the streaming service of RIPE RIS [7], and BGPmon [8], which return in near real-time BGP routes/updates for a given list of prefixes and ASNs. By combining multiple sources, the delay of the detection phase is the min of the delays of these sources. The system can be parametrized (e.g., selecting LGs based on location or connectivity) to achieve trade-offs between monitoring overhead and detection efficiency/speed.

When a prefix hijacking is detected, ARTEMIS launches the mitigation service, which changes the configuration of BGP routers to announce the de-aggregated sub-prefixes of the hijacked prefix. Therefore, ARTEMIS assumes permissions for sending BGP advertisements for the owned prefixes from the BGP routers of the network. This can be effectively accomplished by running ARTEMIS, as an application-level module, over a network controller that supports BGP, like ONOS [10] or OpenDayLight [11]. Prefix de-aggregation is effective for hijacks of IP address prefixes larger than /24, but it might not work for /24 prefixes, as BGP advertisements of prefixes smaller than /24 are filtered by some ISPs.

In parallel to the mitigation, a monitoring service is running to provide real-time information about the mitigation process. This service uses again data from Periscope, RIPE RIS, and BGPmon to monitor/visualize the mitigation.

3. EXPERIMENTS WITH A REAL AS

To evaluate ARTEMIS, we conduct prefix hijackings against our own prefixes in the Internet. We use the PEERING testbed [12], which owns actual AS numbers (ASNs) and IP prefixes, and is connected to the Internet at multiple sites (university networks and IXPs). Through PEERING, we run a virtual AS, which announces a prefix and uses ARTEMIS to detect and mitigate hijackings for this prefix. We then announce the same prefix from another virtual AS of PEERING, emulating effectively a prefix hijacking attack. We associate different sites with the two ASes, and denote them as `ASN-1` and `ASN-2`. Each experiment consists of the following phases.

(Phase-1) Setup. We announce an IP prefix, say 10.0.0.0/23, from the legitimate owner of the prefix (`ASN-1`), and wait until the announcement becomes visible to all the LGs in our arsenal, i.e., for BGP convergence.

(Phase-2) Hijacking and Detection. Then, from a different site of PEERING, `ASN-2` hijacks the prefix 10.0.0.0/23, announcing it with `ASN-2` as the origin AS number. The new announcement disseminates in the Internet as well, and the ASes that are "closer", change their preferred path for the prefix to `ASN-2`. We measure the time until ARTEMIS detects the prefix hijacking by observing an announcement with an illegitimate origin AS in the data it processes from Periscope, RIPE RIS, and BGPmon.

(Phase-3) Mitigation. Immediately after the detection, ARTEMIS triggers prefix de-aggregation to mitigate the attack: it splits the hijacked prefix 10.0.0.0/23 into two more specific sub-prefixes, i.e., 10.0.0.0/24 and 10.0.1.0/24, and announces them. The announcements for the /24 sub-prefixes disseminate in the Internet, and the routes change back to `ASN-1`, since the more specific /24 prefixes are preferred over the initial /23 prefix. We measure the time from the moment prefix de-aggregation is triggered until all the vantage points in our data have switched to the legitimate `ASN-1`.

Our preliminary results over a few dozen experiments show that ARTEMIS needs (on average) 45secs to detect the hijacking, 15secs to announce the de-aggregated /24 prefixes (through the controller), and, after that, the mitigation is completed within 5mins. In total, the hijacking is completely mitigated around 6mins after it has been launched (which is smaller than the duration of $> 80\%$ of the hijacking cases observed in [3]). The detection is faster because it needs *at least one* observation of the bogus route, while the mitigation is completed when *every router* has the legitimate route.

4. DEMO

The goal of the demo is to show that it possible to detect and mitigate BGP prefix hijackings in near real-time on the actual Internet. We will use ARTEMIS over the PEERING testbed to perform hijacking experiments, like in Section 3. Using the monitoring service of ARTEMIS, we will visualize in real-time how the hijacking incident propagates in the Internet, turning affected networks into the illegitimate AS. This, as well as the effect of the mitigation, will be demonstrated with a geographical visualization of vantage points around the globe that select the (il-)legitimate origin-AS.

Acknowledgements. This work has been funded by the European Research Council Grant Agreement no. 338402.

5. REFERENCES

[1] www.ripe.net/publications/news/industry-developments/youtube-hijacking-a-ripe-ncc-ris-case-study.
[2] www.bgpmon.net/chinese-isp-hijacked-10-of-the-internet/.
[3] X. Shi, et al., "Detecting prefix hijackings in the Internet with Argus," in *Proc. ACM IMC*, 2012.
[4] M. Lad, et al., "Phas: A prefix hijack alert system.," in *Usenix Security*, 2006.
[5] "The Route Views Project." http://www.routeviews.org/.
[6] "RIPE RIS." http://ris.ripe.net/.
[7] "RIPE RIS - Streaming Service." https://labs.ripe.net/Members/colin_petrie/updates-to-the-ripe-ncc-routing-information-service.
[8] "BGPmon." http://www.bgpmon.io.
[9] V. Giotsas, A. Dhamdhere, and K. Claffy, "Periscope: Unifying looking glass querying," in *Proc. PAM*, 2016.
[10] http://onosproject.org/.
[11] https://www.opendaylight.org/.
[12] B. Schlinker, K. Z. I., Cunha, N. Feamster, and E. Katz-Bassett, "Peering: An as for us," 2014.