Computer Networks 82 (2015) 135–155

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Self-reliant detection of route leaks in inter-domain routing

M.S. Siddiqui *, D. Montero, R. Serral-Gracià, M. Yannuzzi

Networking and Information Technology Lab (NetIT Lab), Technical University of Catalonia (UPC), Spain

ARTICLE INFO

Article history:
Available online 7 March 2015

Keywords:
BGP
Reliability
Security
Routing
Internet
Inter-domain

ABSTRACT

Route leaks are among the several inter-domain routing anomalies that have the potential to cause large scale service disruptions on the Internet. The reason behind the occurrence of route leaks is the violation of routing policies among Autonomous Systems (ASes). There exist a few rudimentary solutions that can be used as a first line of defense, such as the utilization of route filters, but these palliatives become unfeasible in large domains due to the administrative overhead and the cost of maintaining the filters updated. As a result, a significant part of the Internet is defenseless against route leak attacks. In this paper, we examine the different types of route leaks and propose detection methodologies for improving the reliability of the routing system. Our main contributions can be summarized as follows. We develop a relatively basic theoretical framework, which, under realistic assumptions, enables a domain to autonomously determine if a particular route advertisement received from a neighbor corresponds to a route leak. Based on this, we propose three incremental methodologies, namely Cross-Path (CP), Benign Fool Back (BFB), and Reverse Benign Fool Back (R-BFB), for autonomously detecting route leaks. Our strength resides in the fact that these detection techniques solely require the analysis of control and data plane information available within the domain. We analyze the performance of the proposed route leak identification techniques both through real-time experiments as well as simulations at large scale. Our results show that the proposed detection techniques achieve high success rates for countering route leaks in different scenarios.

1. Introduction

The security and reliability of the Border Gateway Protocol (BGP) [1] have been actively investigated since its adoption as the standardized inter-domain routing protocol among Autonomous Systems (ASes) in the Internet. The implicit trust model among ASes for exchanging reachability information using BGP, along with the lack of in-built security mechanisms in the protocol itself make the inter-domain routing system vulnerable to a number of security threats, such as false IP prefix origination and false route advertisements. As evident from the Youtube incident in 2008 [2] and alleged Chinese Telecom traffic hijacking event in 2010 [3], even non-sophisticated attacks have the potential to globally disrupt the Internet. Another inter-domain routing anomaly with the potential to produce large scale service disruptions is the "route leak" problem. Route leaks occur due to policy violations while exporting routes to a neighbor AS. The ASes typically set their policies for exporting or importing routes from a neighbor AS according to the business relationship that they have with that specific neighbor on a given inter-domain link. There are three types of business relationships between any two ASes: (1) customer–provider; (2) peer–peer; and (3) sibling–sibling relation. In a customer–provider relation, the provider AS offers transit to the customer AS. The ASes in a peer–peer relation usually

* Corresponding author. Tel.: +34 938967294.
E-mail addresses: siddiqui@ac.upc.edu (M.S. Siddiqui), dmontero@ac.upc.edu (D. Montero), rserral@ac.upc.edu (R. Serral-Gracià), yannuzzi@ac.upc.edu (M. Yannuzzi).

http://dx.doi.org/10.1016/j.comnet.2015.02.029
1389-1286/

exchange only their customers' traffic between each other up to an agreed upon threshold. A sibling–sibling relation exists between two ASes which belong to the same organization and the ASes typically offer customized transit to each other. A peer–peer relation is different from a sibling–sibling relation in the sense that the ASes, in the latter case, are owned by the same organization whereas, in the former case, the two ASes belong to two distinct organizations. This difference leads to different type of AS polices among the ASes (cf. Section 7).

A route leak occurs when an AS advertises a route toward a neighbor AS that does not respect the agreed business relationship between them. For instance, if a customer AS starts offering transit between two of its providers, then it is a route leak. Similarly, a route leak will occur if an AS advertises routes learned from one provider toward a peer AS. We will delve into these aspects later on, but in general terms, a route leak entails a violation of the business relationship that rules the interconnection of domains.

The main concern about route leaks is that they are a common occurrence, and regardless if they are due to misconfigurations or deliberate attacks, they can lead to traffic loss, sub-optimal routing, and more importantly, traffic hijacking. For instance, in 2012, a multi-homed ISP leaked routes learned from one of its providers to another provider, causing a national level disruption in Internet service in Australia [4]. Another major route leak incident occurred the same year, when one of Google's peers improperly advertised Google routes to its provider, knocking out Google services for around half an hour [5]—we shall describe these two incidents in more detail later in Section 2.

Route leaks are apparently simple but hard to solve. This is because the ASes keep the information regarding their relationships and policies with other ASes confidential, which makes the identification of policy violations a challenging problem. Although there are orthodox countermeasures for the route leak problem, including route filters, Internet Route Registries (IRRs), and several BGP monitoring tools, they become impotent or unreliable in face of scalability, due to the high cost of maintenance and dependence on third party information.

In this paper, we extend our work presented in [6] where we formally analyzed and developed the route leak problem. In [6], we described different types of route leaks and explained how, where, and why they occur with the help of example scenarios. More importantly, we showed that, under realistic assumptions and routing conditions, a single AS can detect route leaks utilizing only the standard routing information available at hand, and without needing any vantage point deployed in the internetwork. Our approach targets inference and route leak detection requiring neither changes nor extensions to the BGP protocol. Based on the theoretical framework presented in [6], in this paper we develop three incremental route leak identification techniques, namely Cross-Path (CP), Benign Fool Back (BFB) and Reverse Benign Fool Back (R-BFB). The first two techniques are based on the analysis of BGP's control-plane information, i.e., our mechanisms are able to counter a considerable fraction of route leaks utilizing only the information available from the Routing Information Base (RIB) of the BGP routers in the AS—and obviously the knowledge of the AS relationships with direct neighbors as well. The third technique, R-BFB, also takes advantage of data-plane traffic to provide additional information to the analytics performed to the BGP RIBs. The CP, BFB and R-BFB techniques are described in detail in Sections 4–6, respectively. Furthermore, we evaluate the proposed techniques both experimentally as well as through event-driven simulations at large scale. For the latter, we utilized a sub-graph of the Internet graph extracted from ARK [7], and we performed simulations using NS2 [8] and BGP++ [9] on a topology composed of more than 1600 ASes. For the experimental part, we deployed an inter-domain network topology consisting of almost 1000 ASes using Linux Containers (Docker [10]), with the aim of testing our route leak identification techniques in a scenario that can realistically support the data-plane part. The results from our tests, which include more than 20,000 event driven simulations and 1930 real-time experiments, show that an AS is able to autonomously detect route leaks in different scenarios with a high success rate using the CP, BFB and R-BFB, especially, when the three techniques are combined and used together. As far as our knowledge goes, our work introduces the first theoretical and experimental analysis for autonomously detecting route leaks in the Internet.

The rest of the paper is organized as follows. Section 2 describes two real world examples of route leaks. The theoretical framework for detecting route leaks including, definition and description of different types, hypotheses and formalization for their detection, is explained in Section 3. Sections 4–6, introduce the three Route Leak Detection (RLD) techniques, CP, BFB and R-BFB, respectively. The simulations and experimental tests and their results are covered in their respective sections. Section 7 discusses the route leak problem and its detection in sibling–sibling relationship and Section 8 highlights open issues. The related work along with the comparison with our proposed solution is provided in Section 9, and finally, Section 10 concludes the paper.

2. Route leaks in real world

Internet service outages by virtue of the BGP shortcomings are frequent [11], but only a few succeed to get mass attention—in practice this typically depends on the scale of the service disruption and the profile of the victims. In this section, we illustrate two major Internet disruption incidents, that we refer to as Telstra-Dodo [4] and Google-Moratel [5]. The apparent causes behind the disruptions point out to incidents that involuntary produced route leaks. More specifically, these incidents were thoroughly analyzed, and the collected evidence boils down to the violation of routing policies between ASes. However, what could not be clarified, is if they were due to intentional (e.g., a traffic hijack attack) or unintentional misconfiguration (e.g., a fat-finger problem) over the export policies of an AS. Despite the traces and evidence left, we found that some service providers involved in these cases claimed that the issues were due to hardware failures, thereby avoiding to mention the possible case of route leaks [12].