BGPmon: Towards a More Robust Border Gateway Protocol (BGP) Monitoring System

Miguel Cazares
Texas State University
mc1512@txstate.edu

Jason Bartlett
Colorado State University
bartletj@cs.colostate.edu

Cathie Olschanowsky
Colorado State University
cathie@cs.colostate.edu

Dan Massey
Colorado State University
massey@cs.colostate.edu

I. Introduction

The Internet and all of the functionality that depends on it requires a set of rules and policies in the form of protocols. One such protocol, BGP, enables large independent networks within the Internet to connect to each other. However, BGP is susceptible to malicious attacks, such as prefix hijacking. Defending from these attacks requires that internet operators monitor BGP traffic and analyze the data. To fill this need, we introduce BGPmon - a real-time, scalable BGP monitoring tool that enables operators and researchers to monitor and analyze BGP routing data.

II. Background and Related Work

A prefix hijack occurs when a route is falsely announced between Autonomous Systems (ASes) causing neighboring ASes to redirect traffic to the hijacker AS. On April 8th 2010, China Telecom announced 37,000 unique prefixes. This caused very large service outages across the globe because legitimate traffic to numerous ASes was re-routed to China Telecom. This hijack illustrates the necessity of monitoring Internet routing data on the global scale (i.e. BGP data).

Existing BGP data collectors such as RouteViews and RIPE RIS do not provide data in real-time. RouteViews uses only the monitoring subset of software designed to fully implement a BGP router, thus hindering real-time data delivery.

III. Scalable, Real-time Monitoring

Effective attack detection requires that a monitoring system scale to cover a large portion of the Internet. The coverage should include ASes that are both numerous and geographically distant. This enables the dataset to be much larger in volume and will be more useful to accurate analysis and mitigation of attack. The system must also be able to provide this data in real-time. Usefulness of data is directly related to how soon it can be accessed for handling and correction of attacks.

IV. BGPmon Design

BGPmon utilizes a publish-subscribe model to achieve real-time data delivery. In this model, there exist three entities: publishers, subscribers, and brokers. Publishers are the router peers (direct or MRT), subscribers are clients that connect to BGPmon to receive a live XML stream of data, and brokers are BGPmon systems that deliver this stream of data. The XML format contains representation of BGP data in human-readable format and also contains the original BGP data in machine-readable format. BGPmon generates this stream by pushing data through queues so that clients can pull this data from the stream.

Scalability is achieved through chaining. Chaining of two BGPmons is defined as one BGPmon receiving XML output of another BGPmon. BGPmon systems can be arbitrarily chained together to distribute services and span a larger subset of BGP traffic.

V. Implementation and Evaluation

The 7.2 release of BGPmon included the addition of corrupt message handling of Multi-threaded Routing Toolkit (MRT) collector input. The corruption could either occur due to faulty configuration of the collector or incorrect parsing and processing of MRT data. The former indicates actual corrupt data while the latter indicates merely possibly corrupt data. A 5-minute capture of MRT data was collected and stored in a file. The data was then sent to BGPmon to take as input so that possibly corrupt data could be identified. There were 5 possibly corrupt messages in this capture. The messages were then parsed manually and were subsequently identified as actually corrupt.

The ability to send real-time data was evaluated by a client system that read and processed the output XML stream of BGPmon. For each peer, MRT collector, and chain, several attributes were stored that are contained in a 24-hour sliding window; the collection of data was written as a web page for easy accessibility and visualization. A graph is generated every two minutes displaying magnitude of 6 message types for each peer within the past 24 hours. In a typical day, BGPmon receives and processes more than 2 million update messages from direct BGP peers and more than 10 million update messages from MRT peers. This is done while consuming an average of 6.42 GB of memory.

VI. Conclusion

A real-time and scalable BGP monitoring system was presented that enables continuous monitoring of worldwide Internet routing traffic. BGPmon achieves scalable and real-time data through chaining and publish-subscribe models. BGPmon successfully recovers from corrupt BGP messages and handles a high number of BGP update messages. This system can be used by Internet operators for real-time data analysis and by Internet researchers to better understand how Internet traffic is routed on a global scale. Thus, BGPmon enables a strong defense against prefix hijack attacks.