Border Gateway Protocol (BGP) Path Poisoning

Arizona State University CSE 548

Leyba, Kirtus
kleyba@asu.edu

Schmidt, Ryan
raschmi4@asu.edu

Derbique, Austin
aderbiqu@asu.edu

February 2021

Abstract

Internet security at the Border Gateway Protocol (BGP) level is a tug-of-war between BGP attacks and BGP defense mechanisms, often relying on similar technologies. We propose to take this back and forth struggle in a new direction, by using *path poisoning* as a line of defense against censorship and surveillance actions on the BGP network. We propose to implement a technique for path engineering on BGP networks called Fraudulent Route Reverse Poisoning (FRRP) by designing poisoned path announcements on a test network. FRRP was used previously in the Nyx framework to mitigate DDoS attacks. We will test our implementation and verify that FRRP can successfully mitigate DDoS attacks. Additionally, we will extend this work by investigating how FRRP can route around censorship and surveillance on the network. In order to determine if this tool can be used to avoid censoring nodes on the autonomous system (AS) network and prevent surveillance of packets as they move across the network we propose an experimental evaluation of our solution on a realistic network test-bed.

Keywords— SDN, BGP, Censorship, Path Poisoning

I. Introduction

In the Border Gateway Protocol (BGP), autonomous systems (ASes) autonomously gather routes to regions of IP space known as prefixes. These routes are stored within BGP routers in *routing tables* [5]. Together, the routing tables of ASes around the world represent the global routing topology of the Internet. For a variety of reasons network operators are interested in controlling the routes that other ASes adopt. Sometimes this is for economic reasons [2], for repairing outages [2], or as part of a man in the middle attack [3]. Naively, this isn't possible because each AS can only choose where to send packets next, and cannot control the routes that other ASes adopt. Despite this, network operators have discovered a method to conduct traffic engineering on BGP networks: *path poisoning*.

Path poisoning is a method by which an AS announces false paths with a particular structure that influences other ASes to drop routes from their routing tables. This exploits a protection in BGP known as loop detection, where an AS will discard any received routes it receives that already contain that AS [5]. An AS will drop routes containing cycles and will replace them with other valid routes to the same destinations as they are announced.

While path poisoning is an attack on BGP, it can be used in a defensive manner, such as routing around congested links due to an ongoing DDoS attack [7]. The authors of [7] propose a system known as **Nyx** to detect and perform BGP path poisoning to eliminate ASes from routing considerations should they become congested. We aim to implement Fraudulent Route Reverse Poisoning (FRRP), the primary algorithm in Nyx, and apply it to the domain of censorship, where some node is known *a priori* to be monitoring or censoring traffic that flows through it. With application of path poisoning, we can cause incoming and outgoing traffic between us and a target prefix to avoid these censorship nodes.

The internet is increasingly used to express political opinions and organize protests [1], and as such the control of the Internet has become a focus of national actors. Many nation-states make use of BGP and ASes to capture traffic and censor it by injecting spurious TCP RST packets or by returning block pages [1]. On other occasions, entire national networks are shut down by the deactivation of key infrastructure [6]. By routing around such ASes by utilizing path poisoning, we can bypass such censorship and even maintain connectivity when critical ASes are shutdown to prevent communication.

By conducting this project, we expect to demonstrate the effectiveness of BGP path poisoning strategies in avoiding censorship ASes where possible, and detecting cases where avoiding such ASes is not possible. The poisoning method will not require coordination between us and any other AS to be effective. We will build a test network consisting of multiple simulated ASes, each connected together via a topology representing actual internet topologies between core routers. BGP will be used to establish routes between these simulated ASes, and we will then conduct our path poisoning attacks in this network.

II. System Models

A. System Model

Due to BGP's autonomous nature, given a particular network structure it is not trivial to estimate the routing tables for each AS, or even if the routing tables will converge to a stable state. Thus, we require a real test environment to conduct useful experiments. We propose the use of two widely used test-bed environments for network security experiments: GENI and Cloudlab.

1. GENI, otherwise known as Global Environment for Network Innovation, is a large scale experiment infrastructure used to carry out realistic network testing. Not only does GENI allow you to connect compute resources using Layer 2 networks, it also lets you execute custom software or even operating systems on these compute resources [10]. Supported by the National Science Foundation, GENI is a platform that could provide a real test environment for our deployed ASes.

2. The second option was developed more recently and provides some extra features compared to GENI. This platform is called Cloudlab and is considered a "meta-cloud", which means it is not a cloud itself; rather, a facility for building clouds. In fact, Cloudlab interoperates with GENI and may prove useful in providing hybrid functionality of new features in addition to all of GENI's offerings [12].

Using these network testbed environments, we will configure several AS topologies and censorship scenarios. The proposed model will include a sufficient number of ASes to be able to provide multiple path options for a route such that the critical, censoring AS will be avoided. Additionally, we will experiment with the impact of topology and attack locations on the efficacy of our methods.

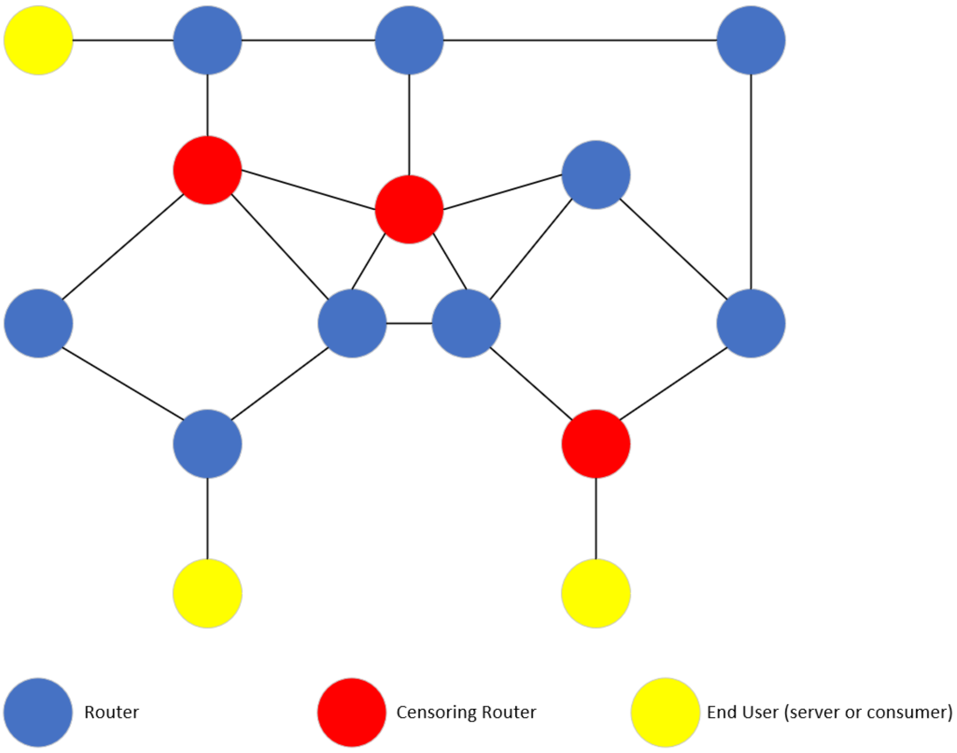


Figure 1: This diagram illustrates a network topology that we will implement to conduct tests on

B. *Software*

Various tools we propose to use are as follows:

- A test network used as an environment to conduct our experiments. This will provide networking resources, compute, and generally a place for our other software applications to run. our test network will be implemented on a CloudLab virtual network and we will implement BGP with the FRRouting controller [4].
- Another software framework worthy of exploration for a test network is Mininet [9]. This software creates a virtual network capable of creating ASes with links and other resources required for testing our algorithm.
- Multiple Autonomous Systems, along with a critical AS which will be the AS to be poisoned. These can be virtualized using FRRouting.
- A traffic generator to send packets to a destination for ASes to route around the critical AS. One such solution is SolarWinds WAN Killer Network Traffic Generate [8].
- A monitor to observe traffic as it passes from AS to AS using BGP such as BGP Stream [11].
- We will write the implementation of FRRP using Python.

III. Project Description

The first phase of the project involves setting up a test network. The test network will contain multiple links between simulated ASes, some of which are pre-designated as "censoring" ASes. In some cases, alternative routes between a source prefix and destination prefix that do not travel through a censoring AS exist. In other cases, it is not possible for packets to route between the source and destination without travelling through a censoring AS. This will allow us to examine the performance of our utility in a variety of realistic cases.

Next, we will build a utility to implement the FRRP framework described in [7]. This framework allows for BGP path poisoning in both the incoming and outgoing direction from our source AS, without requiring coordination from other ASes on the network. We will verify that after the poisoning is performed, that no packets between the source and destination route through a censoring AS.

A. *Project Overview*

Broadly speaking, there are three areas of tasks in this project: writing the paper, setting up the test network, and creating the BGP path poisoning framework. These areas can all be started independently, although we cannot test the path poisoning framework without first having completed the test network. In the task breakdown below, dependencies on previous tasks will be highlighted.

B. *Task 1: Create Proposal Document*

This task involves researching related literature to BGP path poisoning and means of internet censorship. We have identified [7] as the paper to implement after consulting with Dr. Huang, and will be taking that paper in a new direction with avoiding censoring ASes instead of routing around congestion caused by DDoS.

C. *Task 2: Provision Network Resources*

Using research clouds such as CloudLab or GENI, or by using SDN infrastructure elsewhere, we will need to spin up enough nodes to provide a good test-bed for our simulated network. Each node will represent one AS (router), responsible for some /24 prefix on the 10.0.0.0/8 subnet. We will also provision nodes on some ASes to represent users or servers running on those networks.

D. *Task 3: Create Test Network*

We will establish static routes between certain nodes to represent physical interconnects between ASes, and then use FRRouting or similar software to implement the BGP protocol on these nodes to propagate information of these interconnects. This will define a simulated internet between the ASes we set up. Then, we will nominate certain nodes as "censoring" ASes, and install some sort of network monitoring software on them to determine whether or not data is flowing through those nodes at any given point in time. These "censoring" ASes are assumed to be known *a priori* in the actual internet. On the internet, they can be detected using methods described in [1], which is outside of the scope of this project. We will explore using Mininet as a way of deploying said test network[9].

This task depends on Task 2 being completed first.

E. Task 4: Run Services on Test Network

Before implementing any BGP path poisoning attacks, we will first test various user and server nodes to ensure that traffic flows between them as expected, and the users can talk to the services. Some of these flows will travel through censoring ASes, and we will be able to see that traffic logged in our traffic monitoring tool to verify that our "censorship" is working. Example servers would include a webserver providing content over HTTP, a DNS server, and an SSH server. This provides a mixture of TCP and UDP traffic flows to show that our test network is robust and a decent simulation of the real internet.

This task depends on Task 3 being completed first.

F. Task 5: Create Path Poisoning Utility

We will follow the framework given in [7] to create a program that performs BGP Path Poisoning from a given source AS to a given target AS, while avoiding the known censoring ASes. The paper is only concerned with one "critical network" to connect to our source AS, so this would represent the user attempting to bypass censoring for a single website or other online resource rather than a general bypass.

G. Task 6: Test Path Poisoning Utility

To ensure the poisoning attack is successful, we will then test the service from the user to the target server, and verify that the censoring ASes do not log anything in their traffic monitoring tool. If the censoring AS was able to log packets before the poisoning but not afterwards, this indicates the poisoning was successful and the censoring AS is no longer part of the route between the user and server.

This task depends on tasks 4 and 5 being completed first.

H. Task 7: Enhance Path Poisoning Utility

As an enhancement over the paper, we will extend the utility to allow bypassing the censoring ASes for more than just one target AS. When a user wishes to not be censored or monitored, they likely wish that to apply to all of their internet traffic instead of just one destination. The authors of [7] describe extending their framework to multiple critical networks as future work, and we will provide an implementation of that.

This task depends on Task 6 being completed first.

I. Task 8: Create Midterm Report

With our current timeline, we anticipate that the midterm report will be due after we complete Tasks 2, 3, and 5. We will see if we are still on track and what modifications we needed to make to this proposal for the project to move forwards.

This task depends on Tasks 1, 2, 3, and 5 to be completed first.

J. Task 9: Create Final Report and Demo

Upon completion of Task 7, our project is complete and we will work on making the final report and demonstration video.

This task depends on all other tasks being completed first.

K. Project Task Allocation

Our team will work together on all tasks, however, we will allocate the following tasks to task leaders to lead the effort and find solutions to obstacles in the way of getting the task completed. The overall workload breakdown for each task is given in 1. We have elected **Kirtus Leyba** as the project lead.

Task	Task Leader
Create Proposal Document	Austin Derbique
Provision Network Resources	Ryan Schmidt
Create Test Network	Austin Derbique
Run Services on Test Network	Kirtus Leyba
Create Path Poisoning Utility	Kirtus Leyba
Test Path Poisoning Utility	Ryan Schmidt
Enhance Path Poisoning Utility	Austin Derbique
Create Midterm Report	Kirtus Leyba
Create Final Report and Demo	Ryan Schmidt

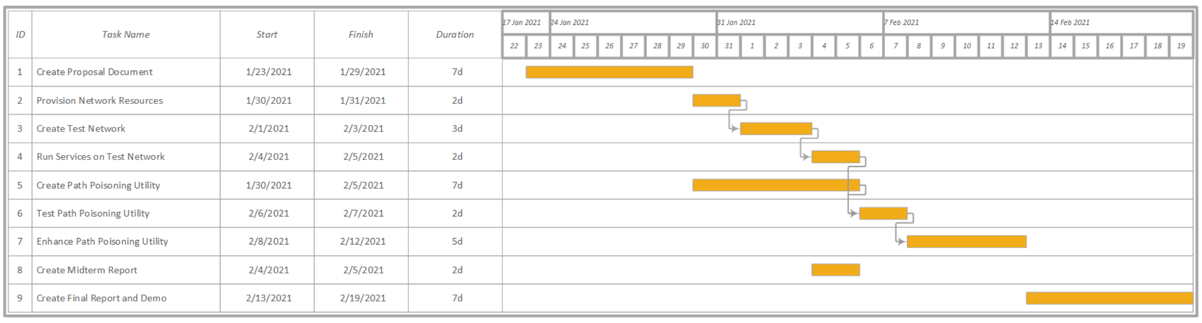
Table 1: The workload distribution of the project

L. *Deliverables*

The main deliverables of the project will be the utility to perform BGP path poisoning given known censoring ASes, a demonstration video showing this utility working on our test network, and a final report detailing our work and the research that went into creating the utility. All of these deliverables will be accessible via the class GitLab repository.

At the time of the midterm report, we are currently working on provisioning the test resources and creating the path poisoning utility. Work on each proceeds independently. CloudLab has not yet approved our account request, so we are looking into alternatives to host our test network should they not come through in the next few days.

M. *Project Timeline*



ID	Task Name	Start	Finish	Duration
1	Create Proposal Document	1/23/2021	1/29/2021	7d
2	Provision Network Resources	1/30/2021	1/31/2021	2d
3	Create Test Network	2/1/2021	2/3/2021	3d
4	Run Services on Test Network	2/4/2021	2/5/2021	2d
5	Create Path Poisoning Utility	1/30/2021	2/5/2021	7d
6	Test Path Poisoning Utility	2/6/2021	2/7/2021	2d
7	Enhance Path Poisoning Utility	2/8/2021	2/12/2021	5d
8	Create Midterm Report	2/4/2021	2/5/2021	2d
9	Create Final Report and Demo	2/13/2021	2/19/2021	7d

Figure 2: Gantt chart of project timeline

IV. Risk Management of the Project

The main risk we are facing right now is related the provisioning of network resources. Our account request at CloudLab is still pending approval, and we cannot provision anything on CloudLab until that is approved. To mitigate this risk and unblock future tasks that depend on the test network being provisioned, we are evaluating other places we can host the network. Ryan Schmidt has a server at home that could additionally be used to host virtual machines for all of these nodes.

Another risk relating to the network setup is if the path poisoning attack disables network connectivity to the point that we are no longer able to log into nodes to determine if packets are flowing properly. This can be mitigated by using out-of-band management such as virtual machine console access to log into nodes and restore proper connectivity.

V. Discussion

Network censorship is another element of network security that isn't often the first concern of network security approaches. Anonymity and data privacy are subjects that are both controversial and of large interest to powerful Internet organizations. These include national networks and Internet companies such as social networks. Our approach proposed here reflects the growing interest in the Internet research community in anonymity and Internet censorship.

Our approach isn't necessarily limited to BGP and the AS network. A potential future research outcome would be developing similar technologies for circumventing nefarious nodes in a variety of networks. Small scale autonomous networks formed by Internet-of-things (IoT) devices have the potential to create a large amount of useful functionality while also presenting new security challenges. Can networks be robust to the style of path engineering proposed here? This is a potential future research interest for the IoT community.

VI. Conclusion

In this proposal we have introduced are experimental plan to evaluate path poisoning as a defense mechanism instead of the attack that it historically has been used for. We will re-implement previous work that mitigated the impact of DDoS attacks and we will extend that work to a new domain: avoiding censorship and surveillance on the AS network.

Acknowledgement

We thank Dr. Dijiang Huang for his guidance on selecting a novel research topic.

References

[1] Shinyoung Cho et al. "A Churn for the Better: Localizing Censorship Using Network-Level Path Churn and Network Tomography". In: *Proceedings of the 13th International Conference on Emerging Networking EXperiments and Technologies*. CoNEXT '17. Incheon, Republic of Korea: Association for Computing Machinery, 2017, pp. 81–87. ISBN: 9781450354226. DOI: 10.1145/3143361.3143386. URL: https://doi.org/10.1145/3143361.3143386.

[2] Matthew Luckie. "Spurious Routes in Public BGP Data". In: *SIGCOMM Comput. Commun. Rev.* 44.3 (July 2014), pp. 14–21. ISSN: 0146-4833. DOI: 10.1145/2656877.2656880. URL: https://doi.org/10.1145/2656877.2656880.

[3] Alex Pilosov and Tony Kapela. "Stealing the Internet: An Internet-scale man in the middle attack". In: *NANOG-44, Los Angeles, October* (2008), pp. 12–15.

[4] FRRouting Project. *FRRouting*. 2021. URL: https://frrouting.org/ (visited on 02/06/2021).

[5] Y. Rekhter, T. Li, and S. Hares. *A Border Gateway Protocol 4 (BGP-4)*. RFC 4271. http://www.rfc-editor.org/rfc/rfc4271.txt. RFC Editor, Jan. 2006. URL: http://www.rfc-editor.org/rfc/rfc4271.txt.

[6] Matt Richtel. "Egypt cuts off most internet and cell service". In: *New York Times* 28 (2011).

[7] James Smith and Max Schuchard. "Routing Around Congestion: Defeating DDoS Attacks and Adverse Network Conditions via Reactive BGP Routing". In: *2018 IEEE Symposium on Security and Privacy (SP)*. 2018, pp. 599–617. DOI: 10.1109/SP.2018.00032.

[8] LLC SolarWinds Worldwide. *Best Network Traffic Generator and Simulator Stress Test Tools*. 2020. URL: https://www.dnsstuff.com/network-traffic-generator-software (visited on 01/28/2021).

[9] Mininet Team. *Mininet*. 2018. URL: http://mininet.org/ (visited on 02/06/2021).

[10] Raytheon BBN Technologies. *What is GENI?* 2021. URL: https://www.geni.net/about-geni/what-is-geni/ (visited on 01/28/2021).

[11] The Regents of the University of California. *BGPStream: A Software Framework for Live and Historical BGP Data Analysis*. 2016. URL: https://bgpstream.caida.org/pubs#bgpstream-tech-rep (visited on 01/28/2021).

[12] The University of Utah. *CloudLab Technology*. 2021. URL: https://www.cloudlab.us/technology.php (visited on 01/28/2021).