A FRAMEWORK FOR DEFENDING AGAINST PREFIX HIJACK ATTACKS

A Thesis

by

KRISHNA CHAITANYA TADI

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

May 2009

Major Subject: Computer Engineering

A FRAMEWORK FOR DEFENDING AGAINST PREFIX HIJACK ATTACKS

A Thesis

by

KRISHNA CHAITANYA TADI

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved by:

Chair of Committee,	Narasimha Reddy
Committee Members,	Alexander Sprintson
	Riccardo Bettati
Head of Department,	Costas N. Georghiades

May 2009

Major Subject: Computer Engineering

ABSTRACT

A Framework for Defending against Prefix Hijack Attacks. (May 2009)

Krishna Chaitanya Tadi, B.E., Jawaharlal Nehru Technological University

Chair of Advisory Committee: Dr. Narasimha Reddy

Border Gateway Protocol (BGP) prefix hijacking is a serious problem in the Internet today. Although there are several services being offered to detect a prefix hijack, there has been little work done to prevent a hijack or to continue providing network service during a prefix hijack attack.

This thesis proposes a novel framework to provide defense against prefix hijacking which can be offered as a service by Content Distribution Networks and large Internet Service Providers. Our experiments revealed that the hijack success rate reduced from 90.36% to 30.53% at Tier 2, 84.65% to 10.98% at Tier 3 and 82.45% to 8.39% at Tier 4 using Autonomous Systems (ASs) of Akamai as Hijack Prevention Service Provider. We also observed that 70% of the data captured by Hijack Prevention Service Provider (HPSP) can be routed back to Victim. However if we use tunneling, i.e. trying to route data to neighbors of Victims which in turn sends the traffic to Victims, we observed that data can be routed to Victim 98.09% of the time. Also, the cost of such redirection is minimal, since the average increase in path length was observed to be 2.07 AS hops.

To my Parents

ACKNOWLEDGEMENTS

I would like thank my advisor, Dr. A. L. Narasimha Reddy, for accepting me into his research team. He has been an excellent guide and mentor throughout my research. I would also like to thank Dr. Alexander Sprintson and Dr. Riccardo Bettati for their willingness to serve in my committee. I will always be grateful to Dr. John D. Oswald for his efforts to provide me a Graduate Assistantship throughout my course of study at Texas A&M University. I would also like to thank my parents who have been a continuous source of encouragement throughout my thesis work. Finally, I would like to thank other members of Dr. Reddy's research group and staff of the Computer Engineering group who directly or indirectly contributed to and had an influence on my research.

NOMENCLATURE

ARIN	American Registry for Internet Numbers
AS	Autonomous System
ASN	Autonomous System Number
ASPP	Autonomous System Path Prepending
BGP	Border Gateway Protocol
CAIDA	Cooperative Association for Internet Data Analysis
CDN	Content Distribution Networks
CIDR	Classless Inter Domain Routing
EBGP	External Border Gateway Protocol
HPSP	Hijack Prevention Service Provider
IANA	Internet Assigned Numbers Authority
IBGP	Interior Border Gateway Protocol
ISP	Internet Service Provider
MOAS	Multiple Origin Autonomous Systems
OSPF	Open Shortest Path First
PHAS	Prefix Hijack Alert System
RIP	Routing Information Protocol
RIR	Regional Internet Registries
WHSR	Weighted Hijack Success Rate

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

CHAPTER I

INTRODUCTION

A prefix hijack involves a hijacker announcing IP prefixes that it does not own into the global routing system. It is a serious security threat in the Internet today. Potentially, a prefix hijack can be launched from any part of the Internet and can target any prefix belonging to any network because Border Gateway Protocol (BGP), which is the major inter domain routing protocol used in Internet, is completely insecure and uses no authentication mechanism. There have been several prefix hijack incidents reported in recent years and they are on the rise. There are several flavors of prefix hijacking such as exact prefix hijack, sub prefix hijack and interception. In both exact prefix and sub prefix hijacks, data is blackholed / dropped at the hijacker. In an interception based attack, hijacker routes the data to the victim after eavesdropping on the information. Such interception based attacks are both difficult to achieve and detect. Attackers hijack IP addresses for the purpose of conducting malicious activities such as spamming and Denial of Service (DoS) without worrying about their identity getting disclosed. Sometimes the attackers want to disrupt the reachability to legitimate hosts or spoof them. Such hijacked IP prefixes were also found to be sold on eBay [1]. Imagine typing www.citibank.com in your browser, which in turn sends/fetches information from the machines belonging to hijacker instead of authentic Citi bank machines.

This thesis follows the style of *IEEE Transactions on Dependable and Secure Computing.*

There are several tools currently available to detect a prefix hijack. Such detection usually involves monitoring changes in the origin Autonomous System set for a given prefix. However the response to thwart such hijack is mostly through NANOG mailing lists or contacting the administrators of hijacking Autonomous System. Considerable time is wasted in such process during which the stability and security of Internet is severely disrupted. If administrators of hijacking Autonomous System fail to cooperate, such hijacking can continue for long periods of time. We propose a mechanism which wastes no time in restricting the hijack and ensures reachability to the authentic prefix owner without any manual intervention.

CHAPTER II

BGP ROUTING

The internet is a vast collection of networks that interact with each other using the TCP/IP protocol suite. Routing protocols are responsible for determining the connectivity in the Internet and for generating routing tables that direct packets to their destinations. Thus routing protocols provide connectivity between every pair of routers in the Internet. This poses a huge scalability challenge with the explosive growth of Internet. The Autonomous System (AS) structure introduces a two level hierarchy that decomposes the problem of determining Internet connectivity into two parts [2]:

(i) Intra Domain Routing: Routing within the AS.

(ii) Inter Domain Routing: Routing between ASs.

An AS is defined by Wikipedia [3] as a collection of connected IP routing prefixes under the control of one or more network operators that presents a common, clearly defined routing policy to the Internet. In other words, AS is one network or set of networks under a single administrative control. An AS might be the set of all computer networks owned by a company, or a college. Companies or organizations might sometimes own multiple ASs. In such cases, each of them is managed independently. A good example is UUNet, which owns one AS for domestic networks (AS 701) and another for International networks (AS 702).

The AS numbers are allocated by the Internet Assigned Numbers Authority (IANA) to the Regional Internet Registries (RIR), which in turn, assigns them to the customers. The low level routing i.e. Intra domain routing is handled by Interior

Gateway Protocols such as Routing Information Protocol (RIP) and Open Shortest Path First (OSPF). At higher level, Exterior Gateway Protocol such as BGP determines AS connectivity or Inter domain routing. Also Classless Inter Domain Routing (CIDR) allows BGP routers to advertise aggregated addresses that reduce the amount of global routing information that needs to be exchanged [2].

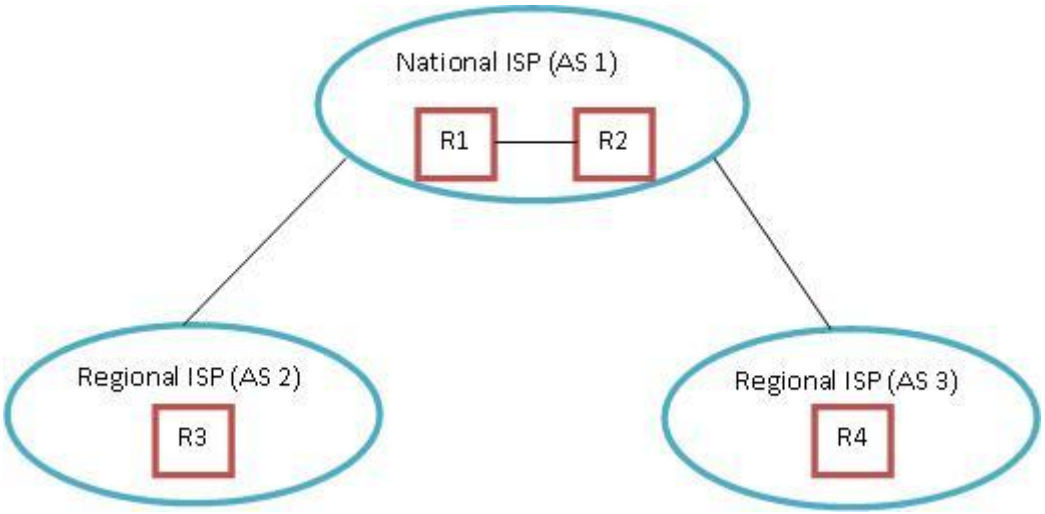

Figure 1. Inter Domain Versus Intra Domain Routing

In Figure 1, AS 1 is a national ISP with two routers R1 and R2, AS 2 and AS 3 are regional ISPs with Routers R3 and R4 respectively. The routing between R1 and R2 is governed by Intra domain routing policies where as routing between R1/R2 and R3, R1/R2 and R4 is governed by Inter domain routing such as BGP. Usually when BGP is used between Autonomous Systems, the protocol is referred to as External BGP (EBGP), if the service provider is using BGP to exchange routes within an AS it is referred to as IBGP (Interior BGP).

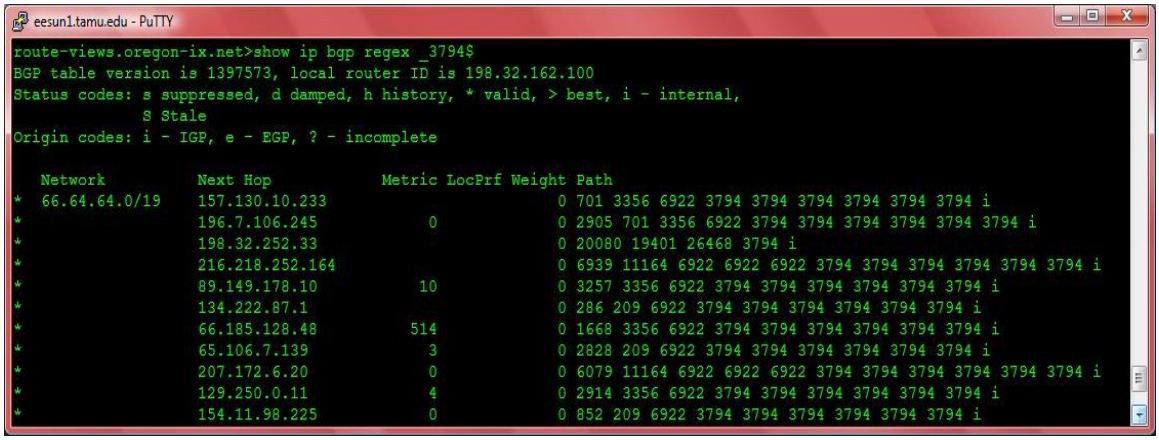

Figure 2. Typical BGP Routing Table

Figure 2 shows some entries in the BGP routing table obtained from RouteViews for IP addresses of AS 3794 which belongs to Texas A&M University. The attributes listed in the Figure 2 are elaborated below [4]:

Network: This is the destination IP address to which the packet must be delivered.

Next Hop: This attribute is the IP address that is used to reach the destination router.

Metric: This is also referred to as Multi Exit discriminator. In Figure 3, Router C is advertising the route 172.16.1.0 with a metric of 10, while Route D is advertising 172.16.1.0 with a metric of 5. The lower value of the metric is preferred, so AS 100 will select the route to router D for network 172.16.1.0/24 in AS 200 [4].

Local Preference: This attribute is used to choose a suitable exit point from the AS. In Figure 4, Router A belonging to AS 100 receives advertisement for network 172.16.1.0/24 and it sets the Local Preference to 50. Similarly Router B belonging to AS 100 receives the advertisement for 172.16.1.0/24 and it sets the Local Preference to 100. These values will be exchanged between Routers A and B, because Router B has higher

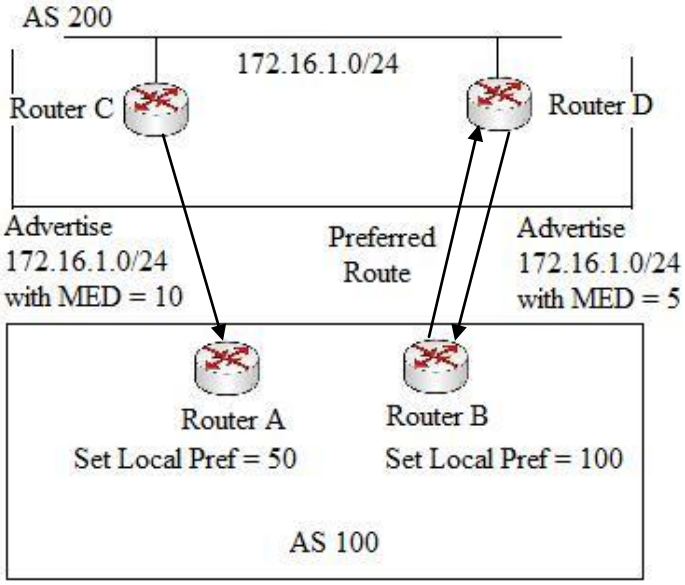

Figure 3. BGP Metric Attribute [4]

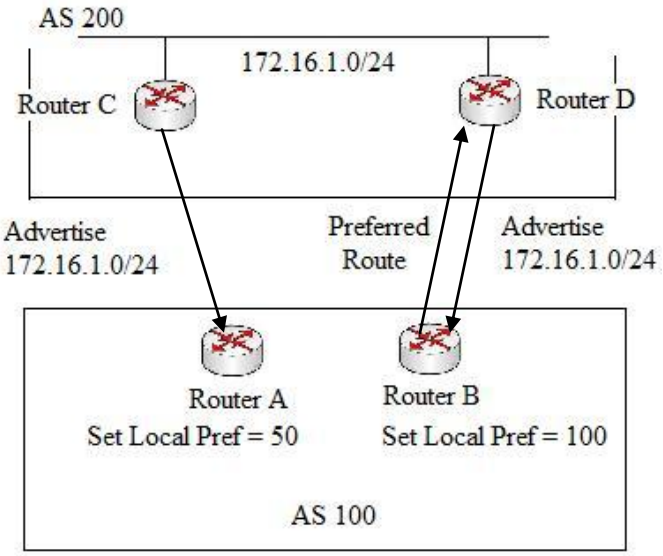

Figure 4. BGP Local Preference Attribute [4]

Local Preference value, it will be used as an exit point to reach network 172.16.1.0/24 while sending data from AS 100 to AS 200.

Weight: This is a Cisco defined proprietary attribute that is local to the router. If the router learns about more than one route to a destination, the route with the highest weight will be preferred.

Path: BGP is a path vector routing protocol. BGP advertises the sequence of AS numbers to reach a destination. In Figure 5, AS D is the owner of the prefix set P_D say 165.91.0.0/16. AS D announces this information to its neighboring routers B and C. Now B propagates the information obtained from D to its other neighboring routers A and C by appending itself to the route. Thus C receives the following information [Path B, D: Destination 165.91.0.0/16] from B and [Path D: Destination 165.91.0.0/16] from D. Now AS C can either choose the path through B to reach D or directly route information to D. The decision depends on the business relationship between Autonomous Systems which is discussed in detail later. If we assume, for now, that path length is the deciding factor, AS C will directly try to route information to D instead of routing it through B. Now C propagates this information to its neighboring routers B and A. B would stick to its original route i.e. directly route traffic to D instead of routing it through C which leads to longer path length. Now A receives the following BGP updates from B [Path B, D: Destination 165.91.0.0/16] and C [Path C, D: Destination 165.91.0.0/16]. Since both paths have the same length (assuming length here refers to number of hops), it can arbitrarily choose the path or the administrator of AS A can create rules/attributes to choose the path appropriately. However, in the real world, such

routing based on path length is rarely true. The business relationship between two connected ASs plays a major role in determining what traffic is routed on a link.

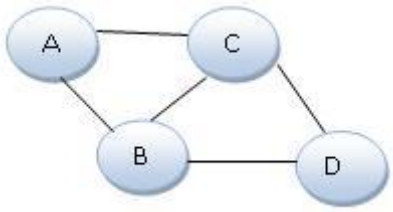

Figure 5. BGP Path Vector Routing

The Business Relationships/Contractual Agreements between the Autonomous Systems are usually classified into the following categories [5].

Customer – Provider: In this scenario, the customer AS pays the provider AS for routing its traffic to the rest of the world. Thus a provider can transit traffic to a customer, but the customer cannot transit traffic between its providers. This is explained in Figure 6. P1 and P2 are the provider ASs of AS C. Thus P1 can route traffic to C or P2 can route traffic to C, but P1 cannot route traffic to P2 via C, because C would end up paying both P1 and P2 for data exchanged between P1 and P2. This is also called the valley free property of Internet.

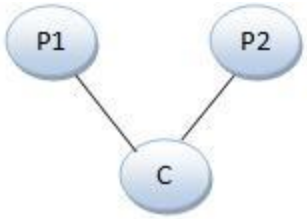

Figure 6. Illustrating Customer Provider Relationship

Peer – Peer: In this, there is usually no monetary flow involved. Two peers exchange traffic between their respective customers. A peer does not act as transit for exchanging data between two other Peers. In Figure 7, P1 is the provider of AS 1, C1 is the customer of AS 1. AS 1 - AS 2 and AS 1 – AS 3 are peers. Now if AS 2 wants to send data to C1, it can send it through AS 1 since C1 is a customer of AS 1. However AS 1 cannot act as transit for traffic between AS2 – P1 and AS2 – AS3.

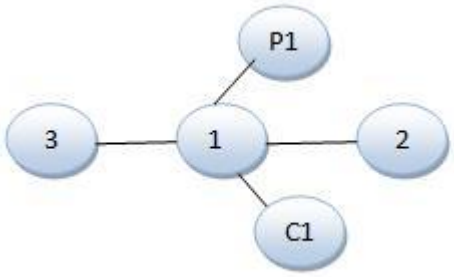

Figure 7. Illustrating Peer - Peer Relationship

Sibling: This refers to link between ASs of the same organization.

The properties described above can be summarized as follows. If an Origin AS can reach a destination AS using a Provider, Peer and a Customer AS, it chooses Customer AS to route its traffic. If the Origin AS has to decide between a Provider and a Peer, it chooses the Peer AS. In other words, the order of preference is Customer > Peer > Provider. A destination AS is on a Customer (Peer, or Provider) route from origin AS, if the first non sibling edge on the route from Origin AS is a Provider – Customer (Peer – Peer, or Customer – Provider) edge. Also, if an AS receives updates for the same prefix from multiple ASs at the same level, say multiple customers, it then decides based

on the path length i.e. it chooses the AS with the least number of hops. Although the administrators of ASs may chose different policies, [6] verifies that the above described no-valley prefer-customer policy is followed in most of the ASs since it makes more commercial sense.

There is no publicly available information about inter AS relationships. Internet registries such as American Registry for Internet Numbers (ARIN) only provide information about who administers an AS. The authors of [5] have proposed an algorithm which classifies the relationships with 99% accuracy. More than 90.5% of AS pairs in the Internet have customer-provider relationship, less than 1.5% of AS pairs have sibling relationships and less than 8% of AS pairs have peer relationships.

Traffic Engineering in Autonomous Systems: An AS with more than one provider is called a multihomed AS. Motivated by the need to improve network resilience and performance, increased number of enterprise and campus networks are connecting through multiple providers [7]. These multihomed ASs, therefore, must undertake the task of engineering the traffic flowing in and out of the network through these multiple links using different inter-AS traffic engineering approaches. ASPP (AS path prepending) is a popular method to achieve such objectives. Prepending here means an AS path that has duplicated AS numbers that appear consecutively. The BGP routing table in Figure 2 shows such paths. Consider the traffic from AS 1 to AS 4 in Figure 8. AS 1 receives two routes for prefixes in AS 4 i.e. (AS2, AS4) and (AS3, AS4). Assuming same local preference, AS 1 can route data either through AS 2 or through AS 3 since the path length is also same for both the cases. If AS 4 wishes that traffic from

AS 1 go through the link AS 2 −AS 4, it can use ASPP and announce AS path (AS 3, AS 4, AS 4) to AS 1. Now AS 1 receives two routes with AS path (AS 2, AS 4) and (AS 3, AS 4, AS 4). Therefore, the router in AS1 would choose the first route.

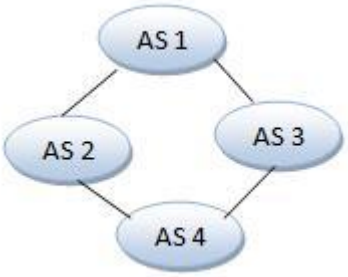

Figure 8. Path Prepending Illustration

There are several other approaches [8], [9] apart from AS path prepending to perform traffic engineering. These techniques are mainly used for Load Balancing, Cost Minimization, Performance Optimization, and Backup Routes [7]. Sometimes these approaches are also used for detecting and preventing prefix hijacks.

AS Hierarchy: ASs with large customer cones have an important role in the Internet structure. At the top of the hierarchy are ISPs known as Tier-1 ISPs. A Tier-1 AS is an AS with no providers and Peers with all Tier-1 ASs [10]. Hence they are at the top of the Internet routing hierarchy. Table 1 shows networks that are believed to be Tier 1 by Wikipedia [11] and CAIDA [12]. Because of no formal definition or body that determines Tier-1 ASs, the term is often misused for marketing purposes. The list of ASs in Table 1 is an approximate estimate and cannot be guaranteed to be 100% accurate. Tier 2 networks are “regional aggregators”, they collect traffic from Tier 3 sites and, if

they cannot satisfy them directly, they pass it on to Tier 1 sites. In other words, Tier 2 acts as transit between Tier 3 and Tier 1. Most of the ISPs are located in Tier 3. An AS connected to only provider ASs is called a stub. Stub ASs are located at the bottom of the AS hierarchy. More details about the BGP protocol/standards can be found in RFC 4271 [13].

Table 1. Tier 1 Networks

AS Number	AS Information	
	ISP Name	Country
AS 1239	Sprint	US
AS 701	UUNET Technologies	US
AS 7018	AT&T	US
AS 3356	Level 3	US
AS 209	QWEST	US
AS 174	Cogent Communication	US
AS 3549	Global Crossing	US
AS 3561	Savvis	US
AS 2914	NTT Communication	JP

Content Distribution Networks (CDN): We used the largest CDN, Akamai for the purpose of analyzing our proposed prefix hijack prevention scheme. CDNs such as

Akamai (http://www.akamai.com) attempt to improve web performance by delivering content to end users from multiple, geographically dispersed servers located at the edge of the network [14]. Most CDNs have their ASs in points of presence of major ISPs so that requests can be forwarded to topologically proximate replicas. Thus they serve as excellent points to implement our prefix hijack prevention framework.

CHAPTER III

PREFIX HIJACKING

We already know that an AS announces IP prefixes belonging to its customers. A prefix hijack occurs when an AS announces prefixes that it does not own. From here on, we will refer to such AS as Hijacker and the AS to which the prefix originally belongs to will be referred to as a victim. In Figure 9 [6], assume AS 6 wrongly announces the prefix that belongs to AS 1. AS 5 previously routed through AS 3 to reach AS 1. On receiving a customer route through AS 6, it prefers the customer route over the Peer route and hence believes the false route. Thus, in this example of prefix hijack, AS 6 announces prefix that it does not own and deceives AS 5.

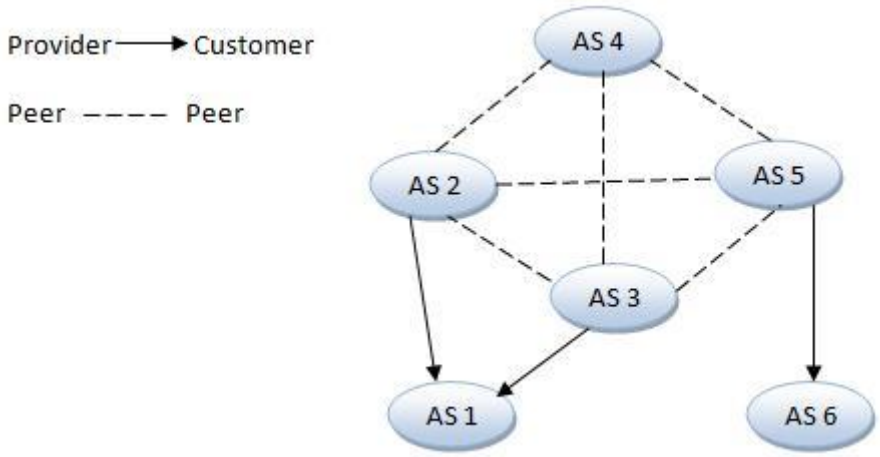

Figure 9. Illustrating Prefix Hijack

Currently there is no authentication mechanism in BGP. Any AS can claim to be the owner of any prefix in the Internet. Taking advantage of this weakness is the fundamental mechanism for constructing prefix hijack attacks [15]. The Internet Routing

Registries such as ARIN maintain databases of prefix ownership, however, the contents are not maintained up to date and hence are not reliable. The attacks where the hijacker announces the same prefix as that of a victim is referred as an exact prefix hijack. Another type of hijack called sub prefix hijack involves an AS announcing a more specific prefix. Say the hijacker announces a /24 prefix, when the true origin announces a /16 prefix. In this case, BGP will treat them as different prefixes and will maintain separate entries for them in the routing tables. Due to the longest prefix matching rule, data destined to prefixes in the /24 range will be routed to the Hijacker AS. One might argue that since both Victim AS and Hijacker AS are claiming to be the owners of the same prefix, we can find multiple entries with the same prefix but different origins (once with Victim as the origin, another one with hijacker as the origin) in the routing table. Can such multiple origin ASs (MOAS) be used as a signature to detect prefix hijacking? The answer is unfortunately NO. The authors of [16], [17] show MOAS prefix can be legitimately announced by multiple origin ASs. Also, hijacker AS can cleverly avoid MOAS anomaly by announcing an AS with an invalid next hop. Assume AS X is the victim which advertises to sender AS S. The path at AS S would be [AS S, …, AS X]. Now in an invalid next hop attack, hijacker AS Y sends advertisement such that AS S will see the path as [AS S, …, AS Y, AS X]. Thus, the origin will remain the same i.e. AS X, making it difficult to detect. However, because of the increased path length, such attacks usually have low impact.

Based on how the hijacker treats the hijacked traffic, prefix hijacks can be classified into following categories:

(i) Blackholing: The attacker simply drops the attracted packets, i.e. packets destined to victim address prefix.

(ii) Interception: The attacker forwards the hijacked traffic to the Victim after eavesdropping on the information in the packets.

Usually Interception is hard to detect and can last a long period of time before being detected. The consequences of Blackholing can be online phishing, spam emails [18] and DDos attack. Interception on the other hand is much more dangerous and at the same time hard to achieve.

Hijack Analysis: Here we discuss all the scenarios occurring in a prefix hijack. The rules obtained here are the basis for the simulation experiments we did to test our hijack prevention framework.

Scenario 1: Existing route is a Customer route, invalid route from the hijacker is a Customer Route. Since both Hijacker and Victim are on the Customer Path, path length is the deciding factor. If the path to hijacker is shorter, hijack succeeds. However, if the path to the Victim is shorter, hijack fails. If both have the same path length, the routing policy attributes such as weight and Local Preference decide the route. In our simulation analysis, the decision is made randomly.

Scenario 2: Existing route is a Customer route, invalid route from the hijacker is a Peer route. Since ASs give higher preference to advertisements from Customer routes compared to Peer routes, the hijack fails.

Scenario 3: Existing route is a Customer route, invalid route from the hijacker is a Provider route. Since ASs give higher preference to advertisements from Customer routes compared to Provider routes, the hijack fails.

Scenario 4: Existing route is a Peer route, invalid route from the hijacker is a Customer route. Since ASs give higher preference to Customer routes compared to Peer routes, the hijack succeeds.

Scenario 5: Existing route is a Peer route, invalid route from the hijacker is also a Peer route. Since both the Hijacker and the Victim are on the same Peer path, path length is the deciding factor. If the path to the hijacker is shorter, hijack succeeds. However, if the path to the Victim is shorter, the hijack fails. If both have the same path length, the routing policy attributes such as weight and Local Preference decide the route. In our simulation analysis, the decision is again made randomly.

Scenario 6: Existing route is a Peer route, invalid route from the hijacker is a Provider route. Since ASs give higher preference to Peer routes compared to Provider routes, the hijack fails.

Scenario 7: Existing route is a Provider route, invalid route from the hijacker is a Customer route. Since ASs give higher preference to Customer routes compared to Provider routes, the hijack succeeds.

Scenario 8: Existing route is a Provider route, invalid route from the hijacker is a Peer route. Since ASs give higher preference to Peer routes compared to Provider routes, the hijack succeeds.

Scenario 9: Existing route is a Provider route, invalid route from the hijacker is also a Provider route. Since both Hijacker and Victim are on the same Provider path, path length is the deciding factor. If the path to the hijacker is shorter, the hijack succeeds. However, if the path to the Victim is shorter, the hijack fails. If both have the same path length, the routing policy attributes such as weight and Local Preference decide the route. In our simulation analysis, the decision is again made randomly.

Table 2 summarizes all the scenarios for prefix hijack analysis. In this table we assume the path length to the Victim, i.e., existing path length is always 'n'.

Interception Analysis: In order to intercept traffic, the hijacking AS should reroute the captured traffic to the Victim. It can do so by forwarding the traffic along its existing valid route to the Victim. None of the ASs in this route should choose the invalid advertisement sent by the hijacker. If they do, traffic loops back to the Hijacker before reaching the victim. The authors of [10] have done a detailed analysis of Interception techniques based on the no-valley, prefer-customer policies discussed earlier. Since our framework concentrates on hijacking, we don't elaborate Interception techniques here. A summary of interception analysis from [10] states that an AS trying to intercept traffic to Victim prefix p can advertise the invalid route to all its neighbors unless its existing route for p to Victim AS is through a provider, in which case, the invalid route should not be advertised to other providers of the AS.

The authors of [6] perform analysis on Resilience and Impact of Autonomous Systems in prefix hijacking. Resilience is defined as the defensive power of a node in hijack, where as Impact measures the attacking power of an AS. It was observed that

Resilience and Impact are directly proportional i.e. a node with high resilience during hijack can also cause major impact during an attack. Figure 10 indicates the Resilience/Impact of nodes in different Tiers.

Table 2. Prefix Hijack Analysis

	Invalid Route	Customer	Peer	Provider
Existing Route	Length			
Customer	<n	Hijack Fails	Hijack Fails	Hijack Fails
	=n	Random	Hijack Fails	Hijack Fails
	>n	Hijack Succeeds	Hijack Fails	Hijack Fails
Peer	<n	Hijack Succeeds	Hijack Fails	Hijack Fails
	=n	Hijack Succeeds	Random	Hijack Fails
	>n	Hijack Succeeds	Hijack Succeeds	Hijack Fails
Provider	<n	Hijack Succeeds	Hijack Succeeds	Hijack Fails
	=n	Hijack Succeeds	Hijack Succeeds	Random
	>n	Hijack Succeeds	Hijack Succeeds	Hijack Succeeds

Historically, it was believed that Tier 1 nodes are the most resilient nodes, but now it came to light that it is the well connected Tier 2 nodes with highest

impact/resilience. Akamai with its heavy presence in Tier 2, or major ISP's such as AT&T with large AS presence in Tier 2 networks can cause high impact in the case of a hijack. This is one of the major reasons to choose Akamai and AT&T for analyzing our framework.

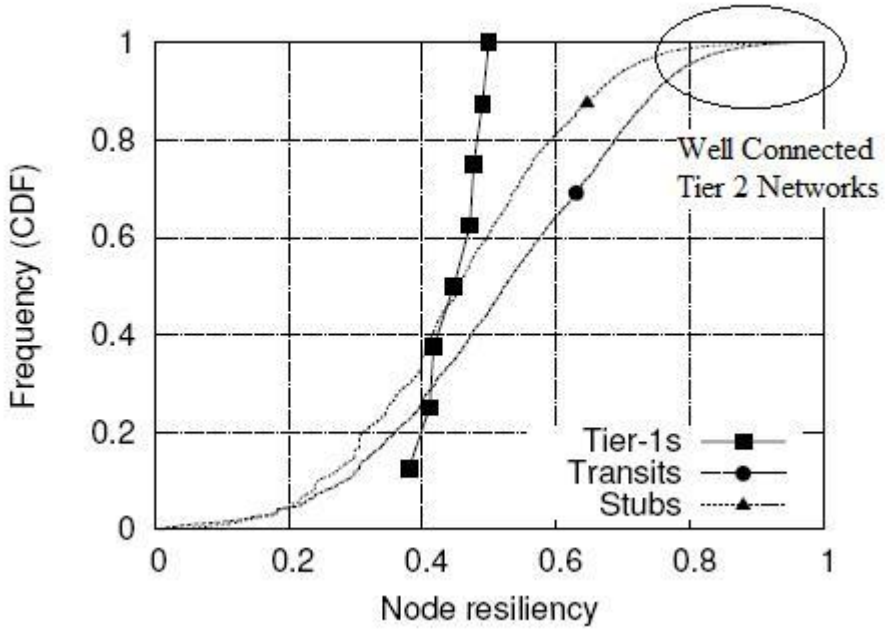

Figure 10. Resilience/Impact of Nodes in Different Tiers [6]

Our prefix hijack threat model assumes the following:

(i) An attacker can hijack TCP connection between peers.

(ii) An attacker can modify updates, delay or delete them.

(iii) An attacker can get access/control to a BGP router and generate false advertisements of prefixes that it does not own or generate non authentic updates.

CHAPTER IV

PREFIX HIJACK INCIDENTS CASE STUDY

There have been several incidents of prefix hijacking due to router misconfiguration. RIPE [19] reports an incident involving prefix hijacking of YouTube on 24 February 2008. AS 36561 (Authentic AS) announces prefix 208.65.152.0/22 belonging to YouTube. Pakistan Telecom (AS 17557) which was trying to block YouTube in its own country claimed as the owner of 208.65.153.0/24. Since PT announced a more specific prefix, due to the way BGP is organized, majority of the ASs chose PT instead of original YouTube AS to route data. Most of the customers of YouTube started receiving "Server not found" error. YouTube realized this and started announcing 2 prefixes: 208.65.152.0/22, 208.65.153.0/24 (longer prefix). This lead to decrease in impact and YouTube started receiving some traffic. After sometime, YouTube started announcing more specific prefixes 208.65.153.128/25, 208.65.153.0/25 as well which further decreased the hijack impact. By this time, PT realized its mistake and used AS prepending to further reduce the hijack impact, and after sometime it withdrew its false announcement. The key points to take from this incident are: Announcing the hijacked route only mitigates the problem but does not solve it. There is some delay before YouTube could realize prefix hijacking on its domain. Our proposed model on the other hand reacts quickly and announces the same prefix from multiple ASs distributed around the world reducing the hijack impact to minimal. It also raises an important question, in case of hijack, can't we announce a more specific prefix and reclaim traffic? The answer is NO. Most ISPs block prefixes longer than /24 because it

leads to explosive growth in the routing tables. Figure 11 indicates the scenarios before and after the hijack of YouTube prefix.

On 25 April 1997, a misconfigured router maintained by a small service provider in Virginia injected incorrect routing information into the global Internet and claimed to have optimal connectivity to all Internet destinations. Because such statements were not validated in any way, they were widely accepted. As a result, most Internet traffic was routed to this small ISP. The traffic overwhelmed the misconfigured router and other intermediate routers, and effectively crippled the Internet for almost two hours [20].

Malicious prefix deaggregation can allow adversaries to take over a prefix by advertising a more specific prefix block. An example occurred in 1997, when misconfigured routers in the Florida Internet Exchange (AS7007) deaggregated every prefix in their routing table and started advertising the first /24 block of each of these prefixes as their own. A /24 block is the smallest prefix generally allowed to be advertised by BGP, and because of its specificity, routers trying to reach those addresses would choose the small /24 blocks first. This caused backbone networks throughout North America and Europe to crash, as AS 7007 was overwhelmed by a crush of traffic and the routes it advertised started flapping [20]. This was not a malicious attack, but an error made by the network operator.

The authors of [6] report more such incidents. On Jan 22, 2006, AS-27506 belonging to RCN New York Communications announced a number of prefixes that did not belong to it. Almost 40 prefixes belonging to 22 unique ASs were hijacked. In another incident, AS 9121 falsely announced routes to over 100,000 prefixes on

December 24, 2004. AS 174 (Cogent) hijacked a prefix (64.233.161.0/24) belonging to AS 15169 (Google) on May 07, 2005. All prefix hijack events are frequently reported in the NANOG [21] mailing list.

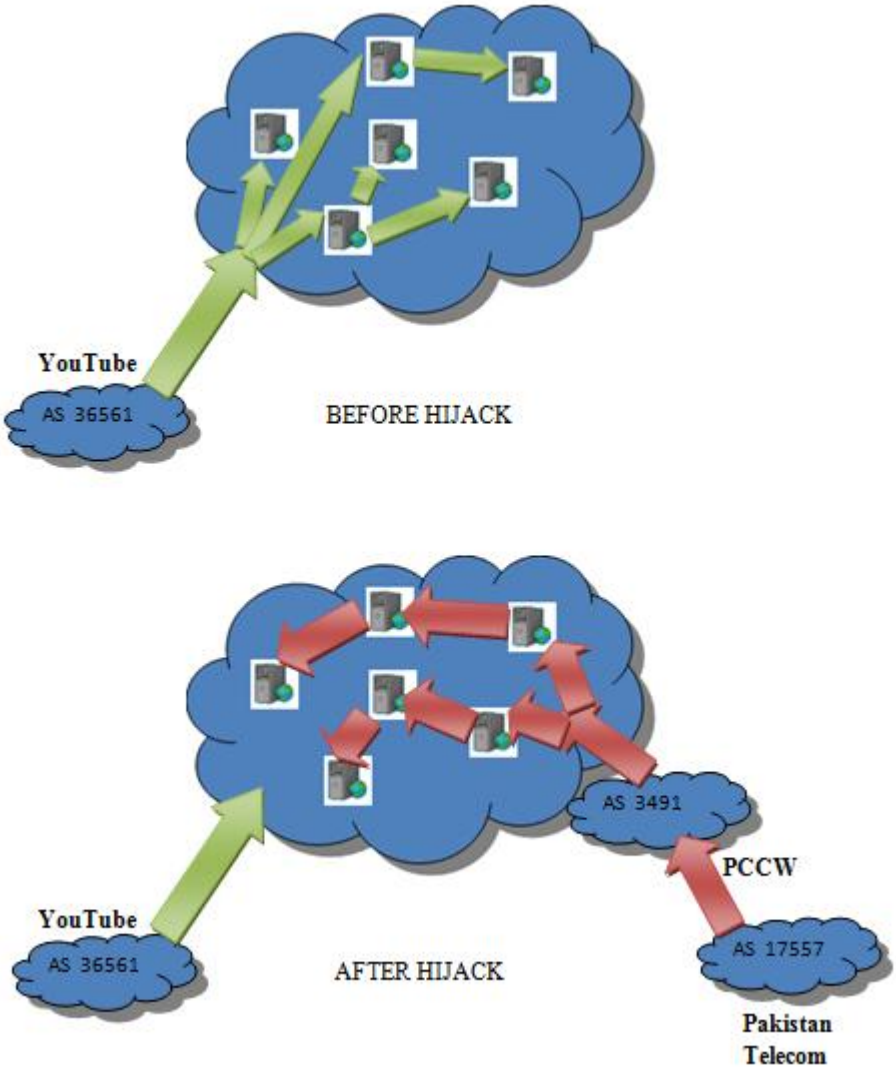

Figure 11. YouTube Hijack Incident [20]

CHAPTER V

RELATED WORK ON PREFIX HIJACK DETECTION AND PREVENTION

The authors of [22] propose a Prefix Hijack Alert System (PHAS). All the prefix owners who are interested in using their service register with the PHAS server and provide contact email addresses. PHAS uses BGP monitor data from RIPE [23] and Route Views [24] to maintain current origin set for each registered prefix. A change in this origin set triggers an origin event, which in turn translates to a notification message to the prefix owner. To control origin events for prefixes with frequent origin changes, a time-window based mechanism is used. This adaptive window based scheme is central to ensure that the system scales from the perspective of origin set monitoring and limits the number of false positives. In addition, there is a local notification filter that administrators can fine tune according to the properties of their AS to further reduce false positives. In fact, RIPE operates an online service MyASN [25] which notifies the network operators when their prefix is announced with an incorrect AS path. It is based on similar principles as that of PHAS.

The authors of [26] propose a protocol enhancement which enables BGP to detect bogus route announcements from false origins. They propose to create a list of multiple ASs who are entitled to originate a particular IP address prefix, and then attach this list to the route announcements by all the originating ASs of this prefix. The BGP community attribute [27] provides a simple way of attaching the MOAS list to a route announcement. The community is a BGP attribute of variable length. It can be used to convey additional information to the global routing system for a group of prefixes that

share some common properties. BGP routers that receive the route announcements from multiple origins can verify that the MOAS is intentional and valid. If another AS makes a faulty route announcement of a prefix p, BGP routers which have received the right route to this prefix p can easily detect the fault since the faulty route's origin AS will not be in p's MOAS list. In Figure 12, prefix p is multihomed and is originated by AS 1 and AS 2. A MOAS list is attached to the routing enhancements indicating that both AS 1 and AS 2 can serve as origin ASs for this prefix. Hijacker AS 4 also originates a route to prefix p, but AS 4 does not appear in the MOAS list announced by AS 1 and AS 2 indicating ongoing prefix hijack. This scheme will only help to detect hijack but does not give information about who the hijacker is. In the above example, it can be either AS 4 or AS1, AS2. Also schemes that require BGP protocol changes at massive levels have failed when it comes to practical implementation. MOAS list also adds to the ever increasing overall size of the routing table and route announcements.

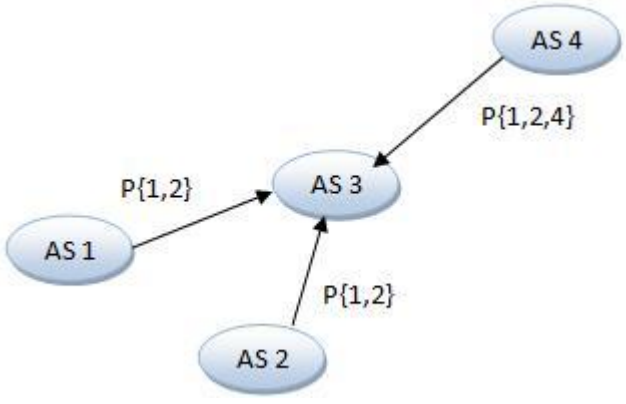

Figure 12. MOAS List Enhancement to BGP Protocol

The authors of [15] present a detection scheme based on the following observations. When a prefix is not hijacked:

(i) The hop count of the path from a source to prefix is generally stable.

(ii) The path from a source to this prefix is almost always a super path of the path from the same source to a reference point along the previous path, as long as the reference point is topologically close to the prefix.

By carefully selecting multiple vantage points and monitoring from these points for any departure from the above listed observations, one can detect a hijack.

Cryptographic based solutions S-BGP [28], SoBGP [29] requiring public key infrastructures for the entire IP address space and AS number space to certify if an AS has the authority to advertise a prefix have been proposed. Such solutions although necessary are hard to be implemented and require universal adoption.

The authors of [30] propose a prefix hijack mitigation system. They preselect several ASs and call them lifesaver ASs. In case of a hijack, a detection system notifies these lifesaver ASs with information about the hijacker AS, Victim AS and the victim prefix. All the lifesaver ASs attempt to purge bogus routes originating from hijacker and promote victim routes by reducing path length in AS_SET.

The authors of [31] propose a detection system based on prefix owner's view of reachability. The system is based on the fact that during prefix hijack, large percentages of ASs in the Internet are polluted, and hence, probes (such as ping) initiated from victim's network are expected to witness unreachability to large number of ASs. In other words, unreachability to large number of ASs is used as a signature of a prefix hijack.

From the current literature, we came to a conclusion that there have been several methods proposed to detect prefix hijack, but very few proposed to prevent a hijack. The ones proposed to prevent hijack are very complex and require global adoption of several BGP protocol changes. Such solutions don't appeal to ISPs because of complexity and overhead. Our proposed framework on the other hand requires no changes at all and can be offered as a paid service making it commercially an attractive and implementable solution.

CHAPTER VI

PROPOSED FRAMEWORK AND RESULTS

MODEL

The goal of our proposed framework is to provide safety against prefix hijack as a service from CDNs such as Akamai or major ISPs such as AT&T with significant presence of ASs in Tier 2 and are well connected. The summary of our goals include:

- Provide Hijack Prevention as a commercial service.
- No infrastructure upgrade required.
- Mitigate an ongoing hijack.
- Not effecting the routing of prefixes in other ASs.

Figure 13 shows an idealistic view of high level Internet topology. It ignores multihoming and cross connections between customer cones.

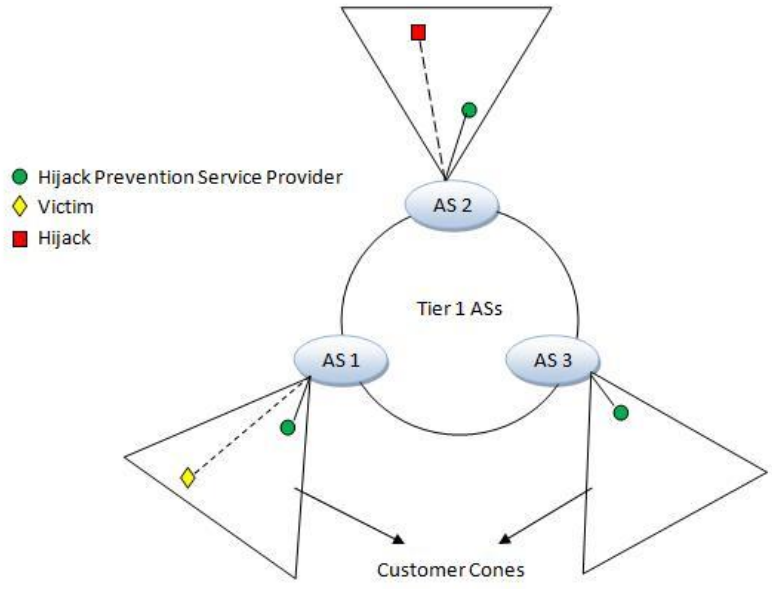

Figure 13. High Level View of Internet Topology

In Figure 13, AS 1, 2 and 3 are Tier 1 ASs and the cones represent their respective customer base. We call the ASs used to implement hijack prevention service as HPSP (Hijack Prevention Service Provider) ASs. These are marked as circles. Diamond represents the Victim AS and Square represents the hijacker AS. In the above scenario, hijacker AS announces a prefix owned by victim AS. HPSP gets its input from any of the several Hijack detection schemes discussed in the previous chapter. Once the HPSP knows the prefix being hijacked, it advertises the same prefix, in other words HPSP claims to be the owner of the hijacked prefix. Since ASs give preference to advertisements from customers, all Tier 1 ASs i.e. AS 1, 2, and 3 believe in HPSP because HPSPs are usually concentrated in Tier 2 and are direct customers of multiple Tier 1 ASs. Now, if any sender in the customer cone of AS 1 or AS 3 tries to send the data to hijacked prefix, it ends up either in the victim or in the HPSP which in turn routes it back to the victim. Thus the impact of hijacker is throttled. In this idealistic scenario, as long as the hijacker and the sender are in different customer cones and HPSP has presence near all Tier 1 networks, the data will always be sent to Victim or HPSP. We assumed that there are 9 Tier 1 ASs in chapter I. There are around 60,000 ASs. Assuming a uniform distribution, each cone has 6666 ASs.

A sender and hijacker can be chosen in ${}^{(9*6666)}C_2$ ways i.e. 1799610021.

Scenarios where the hijacker and sender belong to the same customer cone = $9*({}^{6666}C_2)$ = 199930005

Thus the probability of Hijack success = (199930005)/(1799610021) = 11.1%. However in reality, the scenario is different. There will be cross cone links, distances of

hijacker and sender from Tier 1 and the routing policies of ASs all play an important role.

In all our experiments, we assume there is a strong hijack detection scheme that feeds hijack related data to HPSP in real time. Going forward, we try to answer two important questions:

(i) In case of a hijack, what is the traffic distribution to the Hijacker (ASs that modify their route by believing the hijacker) and to the Victim (ASs that continue to route to the Victim) without HPSP service? Also, what is the traffic distribution to the Hijacker, the Victim and the HPSP AS with active HPSP service?

(ii) How many times can we route the data from the HPSP to the victim?

SIMULATION SETUP

For answering the first question, we used the BGP Routing Information Base (RIB) data from Route Views [24]. Route Views currently collects BGP routing table data from 30 Peers around the world. We can peek into AS level connectivity from these 30 peers mostly distributed in Tier 1 and Tier 2. The raw data obtained has several duplicates, incomplete paths that need to be cleaned. The StraightenRV [32] script available from CAIDA does this. StraightenRV massages a Route Views table for further processing. In addition, it produces a number of files containing statistics, and a 'full' version of the RV table. The full RV table is a standard, easy-to-parse version of the RV table. The .as file contains the following counts for each AS: origin, transit, peer and degree. The origin count of an AS is the number of times the AS appears in the origin

(last) position of AS paths. The peer count of an AS is the number of times the AS appears in the peer (first) position of AS paths longer than one. The transit count of an AS is the number of times an AS appears in a transit (any but first or last) position of AS paths. Once we obtain this data, we need to derive relationships between the ASs. For this we use the Gao's [5] heuristic algorithm. It is based on the valley free property of the Internet. Valley-free property states that:

(i) A provider-to-customer edge can be followed by only provider-to-customer.

(ii) A peer-to-peer edge can be followed by only provider-to-customer edges.

The heuristic algorithm goes through the AS path of each routing table entry. It finds the highest degree AS and lets the AS be the top provider of the AS path. Here degree refers to the number of neighboring ASs. Knowing the top provider, we can infer that consecutive AS pairs before the top provider are Customer to Provider, and consecutive AS pairs after the top provider are Provider to Customer edges. If two pairs of nodes have been classified as both Customer – Provider and Provider – Customer, we convert their relationship to siblings. We ignore siblings for most of our analysis since they constitute approximately 1% of the total links. Now, we need to identify peering relationships. If an AS pair appears consecutively in the AS path and neither of the AS pair is the top provider of the AS path, then the AS pair has a transit relationship and cannot be peers. An AS path has at most one consecutive AS pair that has a peering relationship. That is, a top provider can have a peering relationship with at most one of its neighbors in the AS path. Also, AS pairs that peer have comparable degree. All the facts stated above are used to identify peer relationships.

RESULTS

We first built the basic AS topology and identified the business relationships between each neighboring AS pairs. Since the hijack success rate increases as we move to the top of the AS hierarchy, we analyzed the distribution of ASs at various levels. Figure 14 illustrates the distribution of ASs with respect to average distance from a Tier 1 AS. We measured the path length of an AS to all well known Tier 1 ASs, averaged it and rounded it off to the nearest integer. Although this may not be the exact classification to distribute ASs into Tiers we chose this approach since there are no formal rules for classification of ASs into Tiers. From here on, Tier X AS refers to an AS whose average path length is X-1 from all Tier 1 ASs. The AS distribution data indicated that most of the ASs are located at Tier 3 (41.08%), 4 (36.22%) and 5 (13.81%). Thus ASs from these tiers account for 91.11% of the total ASs.

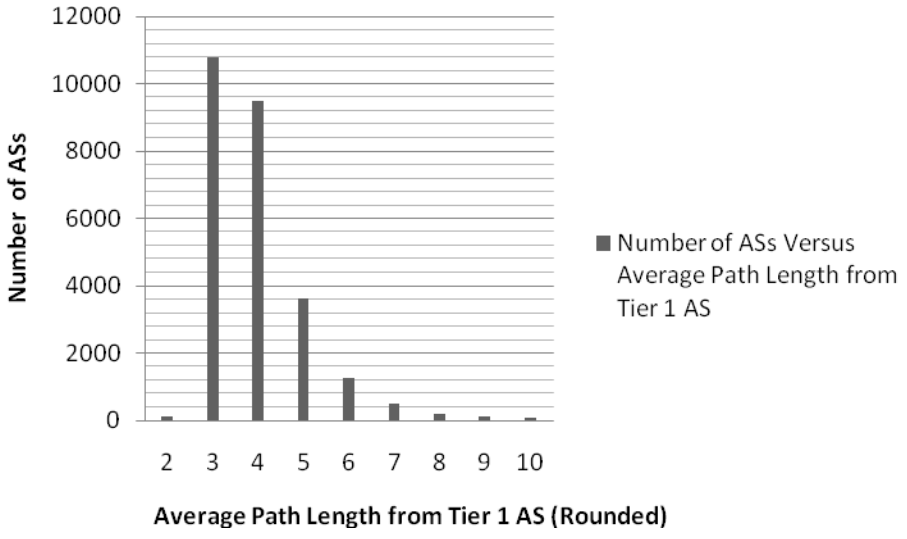

Figure 14. AS Distribution at Different Tiers

To measure hijack success rates, we used Route View monitors as information sources i.e. senders. The hijackers and victims were randomly chosen. Initially, we chose a hijacker at Tier 2. One of the Route View monitors acted as the Sender. Victim was chosen at different tiers. We repeated the experiment now with different senders in the Route View set. The entire process is repeated again for different hijackers in the Tier 2 and the data are averaged. If there were X experiments in total and if it was observed that Y (Y<= X) times the data was routed to the hijacker, then the hijack success rate is defined as:

Hijack Success rate at Tier 2 = (Y/X)*100.

Now this hijack success rate is measured at different Tiers. Figure 15 gives the pseudo code of the algorithm described.

Figure 16 shows the Hijack Success Rate at different Tiers without any Hijack Prevention Service Provider (HPSP). We observe that hijack is successful 90.36% of the time if launched from a hijacker in Tier 2, 84.65% if launched from a hijacker in Tier 3, 82.45% of the time if launched from a hijacker in Tier 4, 79.85% of the time if launched from a hijacker in Tier 5. Thus we conclude that the impact of hijack is severe without any hijack prevention/mitigation mechanism.

Now, we repeat the experiment with a HPSP. The pseudo code in Figure 15 still applies, but now in Step 20, we carry out hijack analysis with H (Hijacker), V (Victim), RV (Route View monitor/Sender) and HPSPs (Hijack Prevention Service Providers). Also in Step 22, hijack succeeded if Victim gets the data, but now hijack succeeds if

either victim or HPSP gets the data. The routing of data from HPSP to victim is analyzed later.

```
1.  X=2, Total_Experiments = 0, Hijack_Success = 0, Count = 10.
2.  Choose a Unique Random Hijacker H in Tier X.
3.  Count--
4.  If Count = 0
5.      Count = 10
6.      X = X + 1
7.      If X > 5
8.          Print Total_Experiments, Hijack_Success
9.          End Program
10.     EndIf
11.     Go to Step 2
12. EndIf
13. If all RV monitors have been selected as Senders
14.     Go to Step 2
15. Else
16.     Select a Route View monitor RV.
17. EndIf
18. For Y = 2 to 5
19.     Choose a Victim V in Tier Y.
20.     Carry out hijack analysis with H, RV and V.
21.     Total_Experiments++
22.     If Hijack Succeeds
23.         Hijack_Success++
24.     EndIf
25. EndFor
```

Figure 15. Pseudo Code to Measure Hijack Success Rate

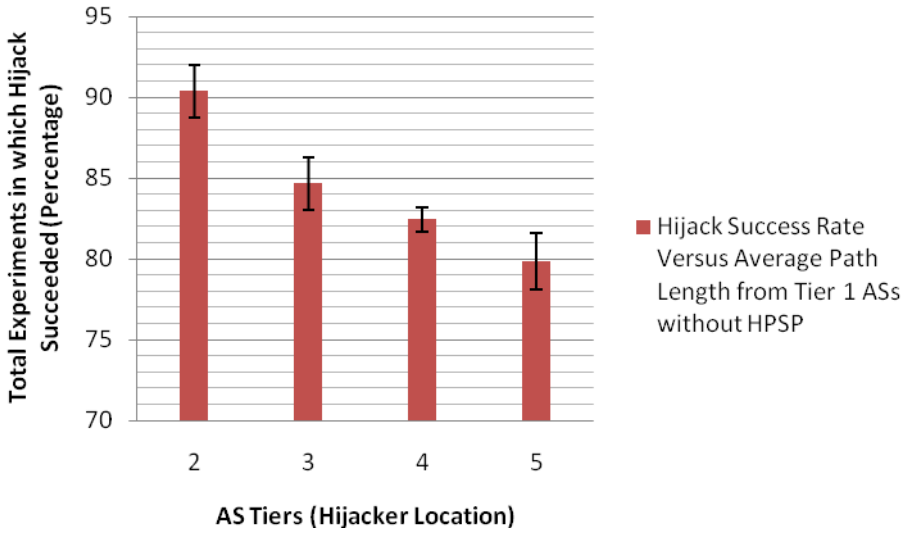

Figure 16. Hijack Success Rate without HPSP

Ideally HPSP ASs should be located in Tier 2, with strong connectivity to Tier 1 and Peer Tier 2 ASs. CDN's such as Akamai and large ISPs such as AT&T fall into this category. Akamai has around 22 ASs distributed mainly in Tier 2, Tier 3 and Tier 4. Figure 17 indicates the results of Hijack success rates with Akamai as HPSP. We observe the hijack is successful 30.53% of the time if carried from a Tier 2 hijacker, 10.98% of the time if launched from a Tier 3 hijacker, 8.39% of the time if launched from a Tier 4 hijacker, and 2.66% of the time if launched from a Tier 5 hijacker. In conclusion, we can say that the hijack success rates reduced significantly by using HPSPs. Figure 18 indicates the results of Hijack success rates with AT&T as the HPSP. The results are similar to that observed with Akamai. We observe the hijack is successful 33.96% of the time if carried from a Tier 2 hijacker, 6.69% of the time if launched from

a Tier 3 hijacker, 10.68% of the time if launched from a Tier 4 hijacker, and 9.5% of the time if launched from a Tier 5 hijacker.

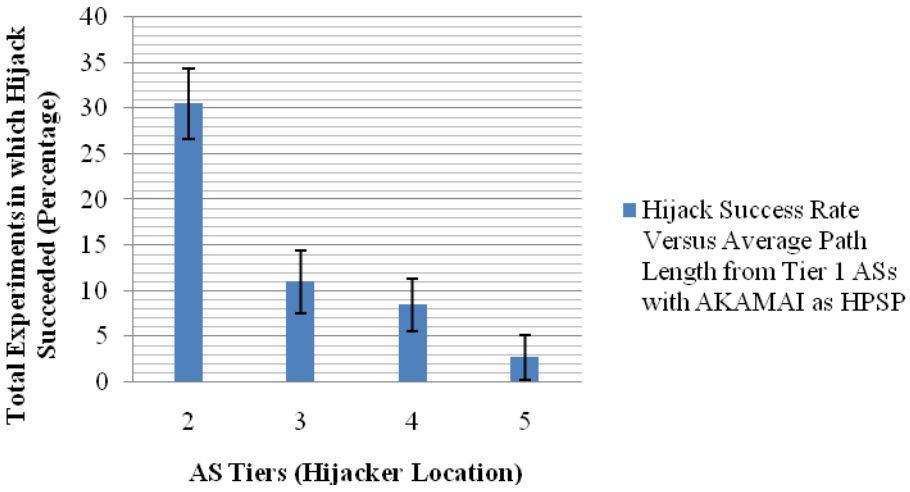

Figure 17. Hijack Success Rate with AKAMAI as HPSP

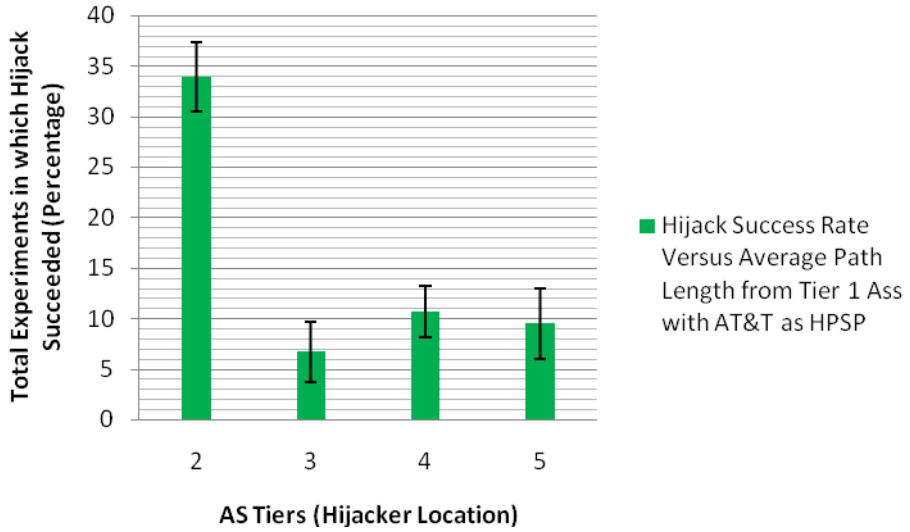

Figure 18. Hijack Success Rate with AT&T as HPSP

In Figure 19, we compare all the hijack experiments. We can conclude clearly that using HPSP reduces the hijack success rate drastically. Also Figure 19 shows results with both Akamai and AT&T acting as HPSPs. For such a scenario, we observe the hijack is successful 24.3% of the time if carried from a Tier 2 hijacker, 5.62% of the time if launched from a Tier 3 hijacker, 2.96% of the time if launched from a Tier 4 hijacker, and 2.2% of the time if launched from a Tier 5 hijacker. Assuming a hijacker can originate with equal probability from any AS, we take the distribution of ASs in Tier 2, 3, 4 and 5 into account and combine with the hijack success results and calculate Weighted Hijack Success Rate.

$$\text{WHSR} = \sum_{Tier=2}^{5} (\text{Hijack Success Rate in Tier} * \text{Average AS distribution in Tier})$$

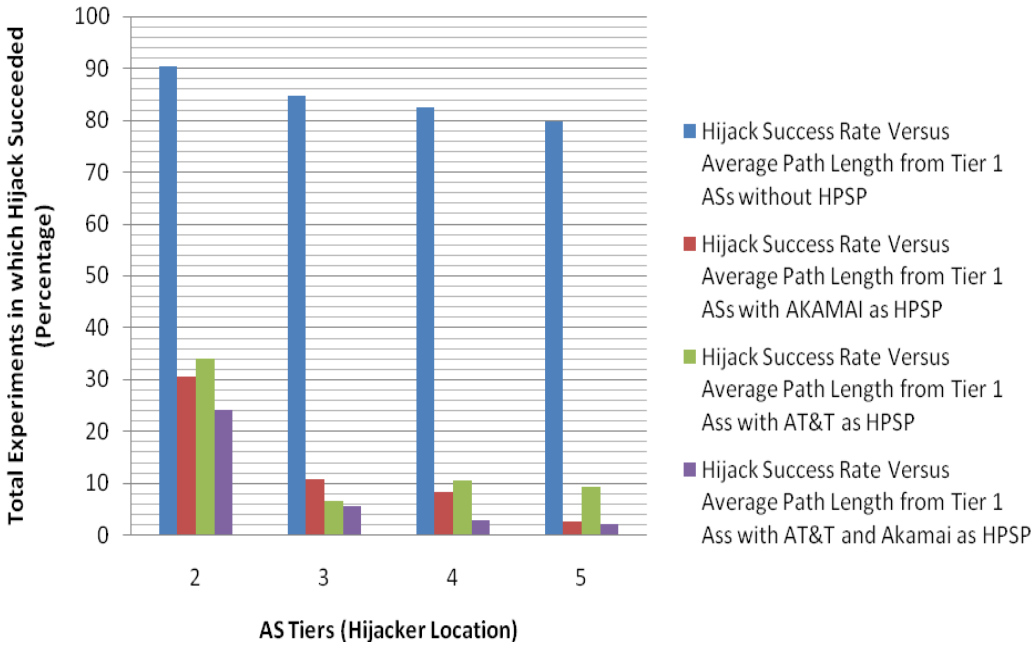

Figure 19. Comparing All the Hijack Experiments

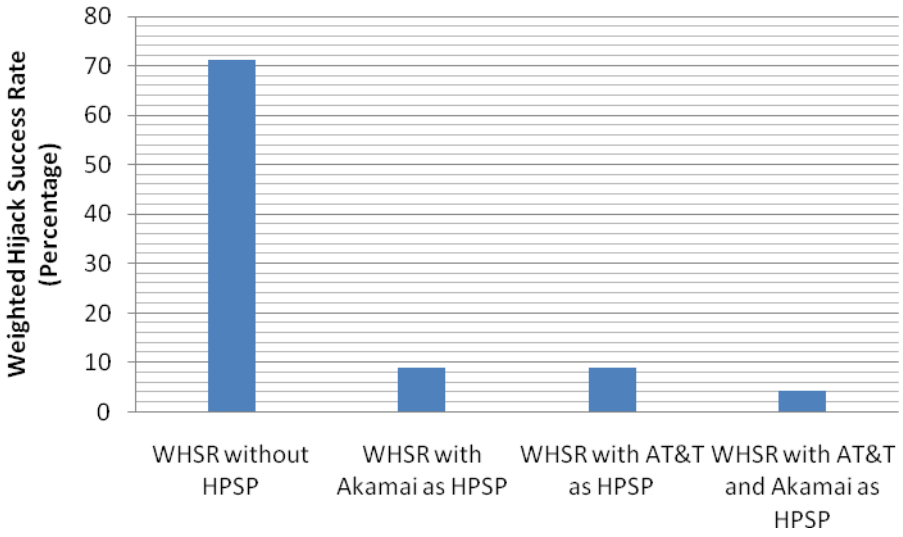

Figure 20. Weighted Hijack Success Rates for Different HPSPs

Figure 20 gives the WHSR figures for all the experiments. WHSR drops from 71% to 8.8% by using an HPSP, and from 8.8% to 4% using multiple HPSPs.

We studied if there exists any correlation between the geographic locations of HPSP ASs to the geographic locations of data originators/source to guide the placement of Hijack Prevention Service Provider ASs. We choose Akamai as HPSP for this experiment. The results are illustrated in Table 3. RV refers to Route View node or data originator, A refers to AKAMAI node i.e. HPSP. ASi refers to Asia, EU refers to Europe and US refers to United States. A value of 36 for (RV(US), A(US)) indicates that in 36 hijack events the data from a Route View AS node in US ended up at AKAMAI AS in US. From the data we found that geographic location does not have a strong impact on the traffic that it attracts. For example most of the data originated at US, ASi ended up at

EU AKAMAI node. We attribute this result to Akamai AS node 20940 located in Germany which is well connected with Peer ASs located in US and ASi.

Table 3. Correlating Geographic Locations of HPSP ASs with That of Data Originators/Source

Data Originator, Destination AS	Hijack Events
RV(US), A(US)	36
RV(EU), A(EU)	43
RV(ASi), A(ASi)	0
RV(US), A(EU)	40
RV(US), A(ASi)	0
RV(EU), A(US)	11
RV(EU), A(ASi)	0
RV(ASi), A(US)	5
RV(ASi), A(EU)	32

To prove that well connected ASs at Tier 2 do a good job as hijack defender, we did two experiments with ATT ASs. In the first experiment we chose only 6 ATT ASs that have strong peer network at Tier 2 and that are directly connected with multiple Tier 1 ASs. The hijack success rates were observed to be 36.77%, 10.34%, 7.4% at Tier 2, Tier 3 and Tier 4 respectively. We repeated the experiment with all the 122 ATT ASs

providing the service, which included the 6 ATT ASs in the first experiment. The success rates were observed to be 33.96%, 6.69% and 10.68% respectively at Tier 2, Tier 3 and Tier 4 respectively. The results are compared in figure 21. Thus, we don't see much difference in hijack success rates even though we increased the ASs providing the service from 6 to 122. From this we can deduce that, ASs that are customers of multiple Tier 1 ASs and have a strong peer network at Tier 2 are the ideal choice for providing the proposed hijack defense service.

We also did an analysis on which Akamai ASs i.e. HPSPs captured the hijacked data and results are illustrated in Figure 22 (X axis indicates hijack simulation events, Y axis indicates ASs. A long horizontal bar indicates that AS was able to capture data from several hijack simulation events). Although all the 22 Akamai ASs are announcing the prefix, only 4 ASs (12222 in USA, 20940 in Germany, 34164 in London, 21342 in Asia) are successful in capturing the data most of the time. This can cause a problem because, if a single HPSP AS is overwhelmed with traffic, it might lead to congestion near this AS. To prevent this, we experimented by increasing the path lengths for ASs 12222 and 20940 which capture most of the traffic using Route Aggregation which can be implemented without any changes to the BGP protocol. This, however, leads to increase in hijack success rate which is illustrated with an example in Figures 23 and 24. Figure 23 indicates a network topology where A refers to HPSP node, RV is the route view node which originates traffic, H is the hijacker node trying to capture traffic. T is a transit node. A is on a customer path for RV node, and so do H and T. H is also on a customer path for T. During Route aggregation, node A tries to append itself to the path

before sending the BGP update to RV node. If A appends itself 3 times, A appears to be 3 hops away for RV node as shown in Figure 24. Thus RV would send the traffic to Hijacker instead of AKAMAI.

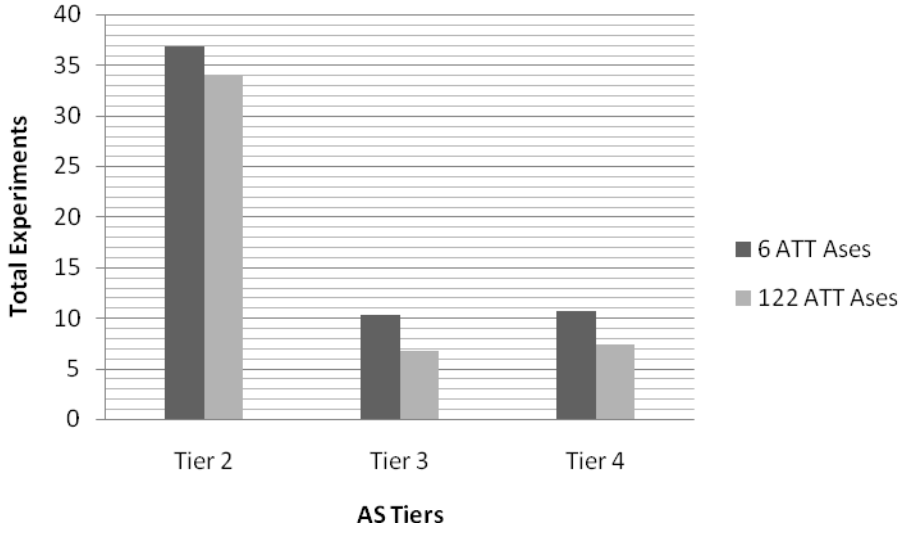

Figure 21. Comparing HSR between 6 (Well connected) ATT ASs and 122 ATT ASs

We experimented with several combinations and finally found that increasing the path length for AS 12222 by 1 and that of AS 20940 by 2 yields best traffic distribution with minimal increase in hijack success rate. The traffic distribution results after Route Aggregation are shown in Figure 26. The effect of Route aggregation on Hijack Success Rate is indicated in Figure 25. As expected the Hijack Success Rate (HSR) rates go up.

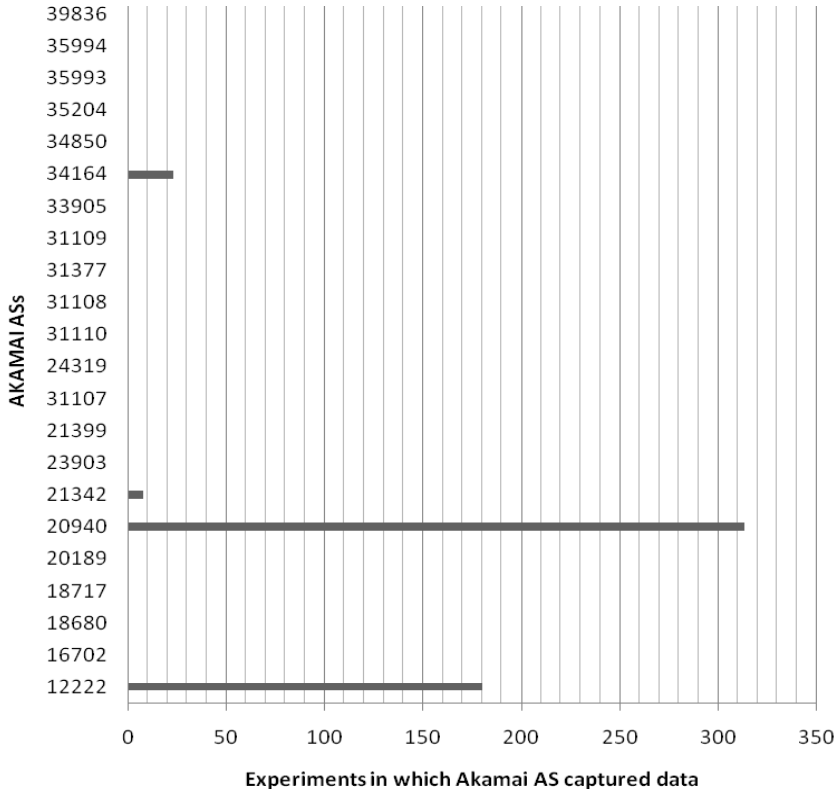

Figure 22. Traffic Distribution in HPSP ASs

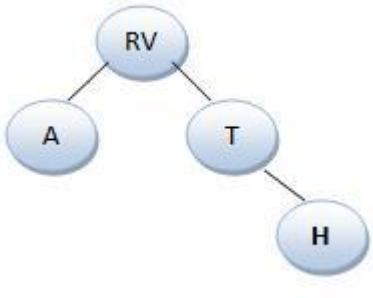

Figure 23. Normal Topology

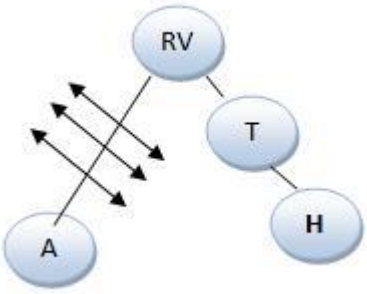

Figure 24. Topology with Route Aggregation

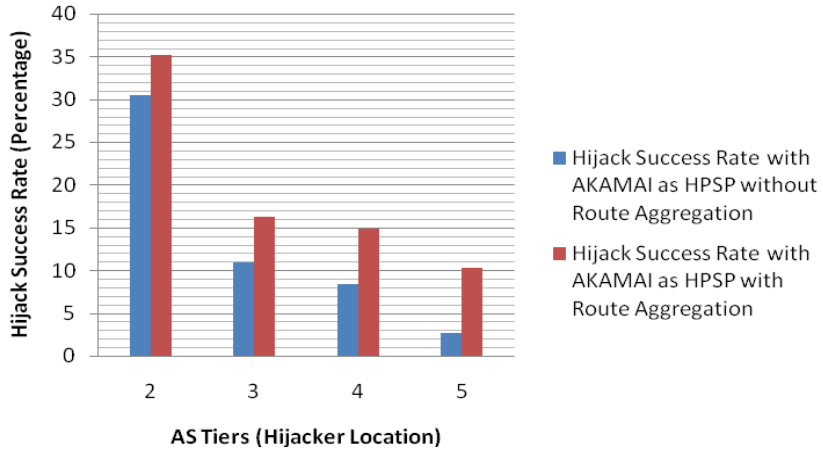

Figure 25. Effect on HSR due to Route Aggregation for Better Traffic Distribution

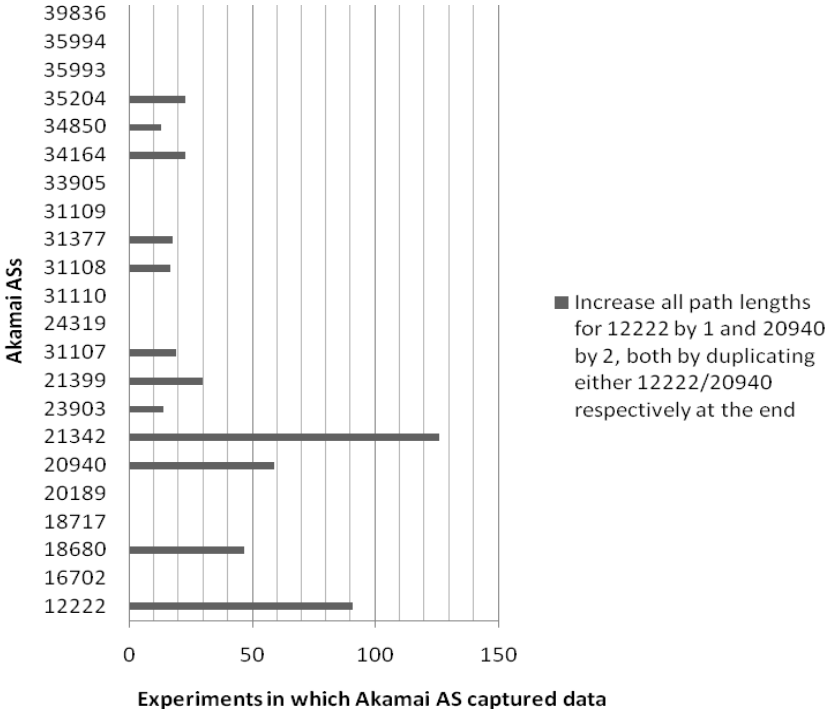

Figure 26. Traffic Distribution after Route Aggregation

ROUTING DATA FROM HPSP TO VICTIM

Once the HPSP captures the data, it needs to route it back to the Victim or the authentic owner of the prefix. In other words, HPSP is now trying to intercept the Victim's data. This is shown in Figure 27. The circles which represent HPSP ASs should route the captured data back to the Victim. However, sometimes, it might happen that all the neighboring ASs of HPSP are corrupted by HPSP/Hijacker ASs and thus HPSP can't route data to Victim. From Victims point of view, such traffic, which is captured by HPSP but can't be routed to the Victim, is equivalent to Hijacked traffic thus increasing HSR.

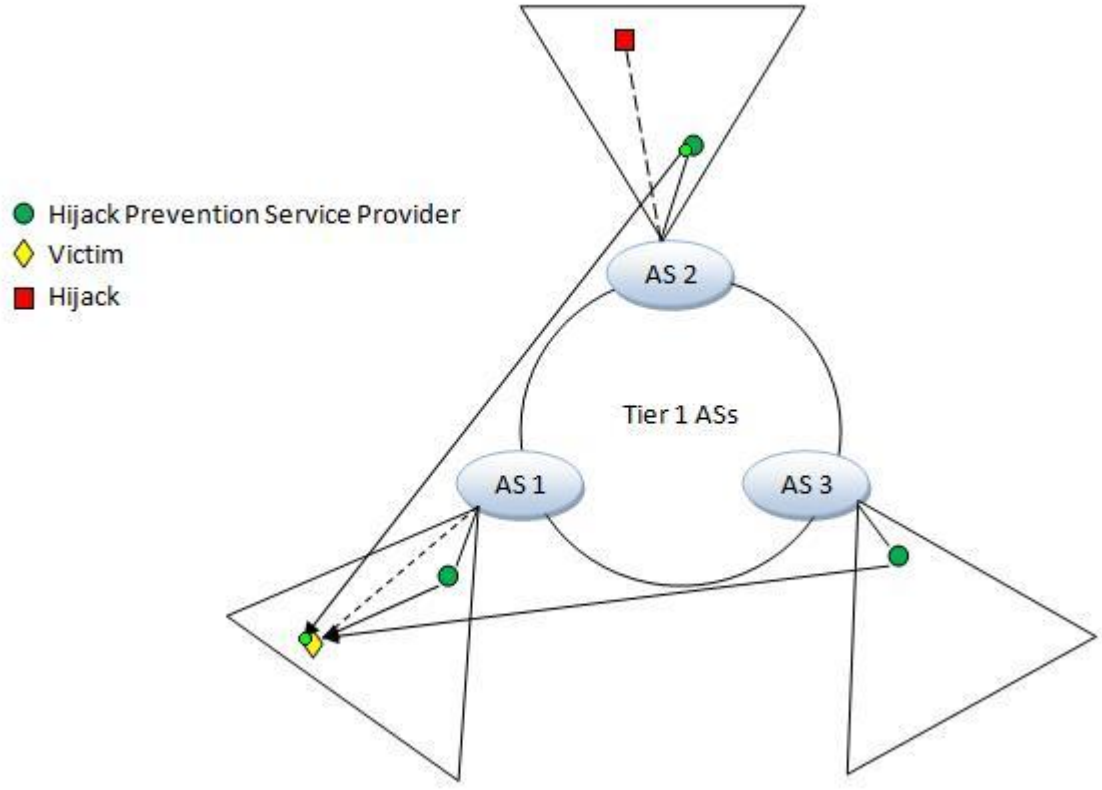

Figure 27. HPSP Trying to Route the Data Back to Victim

For all the simulations till now, we used the data from Route Views. Since we always chose Route View monitors as sources of information, we had path information from sender to victim, sender to HPSP, sender to Hijacker to carry out hijack analysis. But now, we also need data about all possible paths from HPSPs to Victim and if any ASs in this path prefer Hijacker/HPSP instead of the Victim. For this, we constructed a strong AS level connectivity graph by combining the data from Route Views [24], RIPE [23]. Although the quality and accuracy of such AS maps is not 100%, it's a generally accepted notion that such BGP maps constructed from publicly available data are good enough for most BGP simulation experiments [33].

Table 4 summarizes the results of our experiments with routing back the data from HPSP (in this experiment Akamai ASs) to the victim. For this, we randomly chose a Sender AS, a Hijacker and a Victim AS. Then we did an analysis on who will capture the data sent by the Sender AS. If HPSP wins, we retain the experiment results, else, we ignore the results of the experiment. We repeat this now by choosing a different Hijacker and a Victim AS. This process is repeated until we have 15 valid experiments in which HPSP wins. The entire process is again repeated with a different sender AS. Average Path Length in Column 3 indicates the average path length from sender to victim without any hijacker or HPSP AS. Column 4 lists the cases in which HPSP receives the data and can send it back to the Victim. Column 5 lists scenarios, where HPSP cannot send the data directly to Victim, thus uses the prefixes advertised by neighbors to tunnel the data to the victim. This requires victim ASs to have some partnership with neighbors so that the tunneled data can be routed back to the victim. Column 6 indicates scenarios where

data can neither be sent directly nor tunneled. This might occur because, the neighboring ASs of the Victim believe in advertisements from the Hijacker or the HPSP rather than the Victim. However, such instances are very rare. Column 7 indicates the new average path length because of routing data through the HPSP.

Table 4. Analysis of Routing Back the Data from HPSP (Akamai) to Victim

Sender AS	Total Experiments (We ignore details of experiments in which HPSP fails to capture the data)	Average Path Length	Experiments in which HPSP can send the data directly	Exp. in which HPSP can send data to its neighbor	Exp. in which HPSP cannot send the data	Average path Length in Experiments where it can send the data (Sender -> HPSP + HPSP -> Victim/Neighbor)
AS18423	15	3.8	11	4	0	5.85
AS29520	15	3.4	14	0	1	5.57
AS3893	15	5.26	13	2	0	7.07
AS16484	15	3.8	10	4	1	7.75
AS6885	15	4.33	8	7	0	5.2
AS22321	15	4.13	6	9	0	5.28
AS1890	15	3.66	9	6	0	8.22
AS11299	15	5.2	10	4	1	7.5
AS18924	15	3.66	12	3	0	4.08
AS31223	15	5.6	10	5	0	6.2
AS22312	15	4	12	3	0	4.92
AS2299	15	3.8	11	4	0	7
AS5433	15	4.46	9	6	0	8.5
AS42332	15	4.8	13	2	0	5.84

```
1. Total_Experiments = 0; HPSP_direct = 0; HPSP_indirect = 0; HPSP_nodata = 0,
   Orig_avg_path_len =0; New_avg_path_len = 0;
2. Choose a sender AS.
3. While Total_Experiments <=15
4.      Choose a random Hijacker AS and Victim AS.
5.      Analyze advertisements received at Sender AS from Hijacker, Victim and HPSP
        nodes.
6.      If Hijacker/Victim wins the traffic from Sender
7.              continue;
8.      EndIf
9.      Total_Experiments++;
10.     Orig_avg_path_len = Orig_avg_path_len + (No. of hops from Sender to Victim)
11.     Analyze advertisements at HPSP nodes to determine, if there exists a path to the
        Victim node.
12.     If path exists from any HPSP node to Victim
13.             HPSP_direct++;
14.             Choose shortest path from HPSP to Victim. (NOTE: We assume overlay
                network HPSP Peers. Thus the HPSP node that receives or captures the
                data may be different from HPSP node that sends data to Victim. This
                reduces the path length/delay).
15.             New_avg_path_len = New_avg_path_len + (Hops from Sender to HPSP)
                +  (Hops from  HPSP to Victim)
16.     Else
17.             Get the neighbor ASs of Victim
18.             Tunnel the data to neighbor of Victim using prefixes announced by the
                neighboring AS.
19.             If any neighboring AS can send data to Victim
20.                     HPSP_indirect++;
21.                     New_avg_path_len = New_avg_path_len + (Hops from Sender to
                        HPSP) + (Hops from HPSP to neighbor) + 1
22.             Else
23.                     HPSP_nodata++;
24.             EndIf
25.     EndIf
26. EndWhile
27. New_avg_path_len = New_avg_path_len / (HPSP_indirect);
28. Orig_avg_path_len = Orig_avg_path_len/(Total_Experiments);
```

Figure 28. Pseudo Code to Route Data from HPSP to Victim

Table 5. Analysis of Routing Back the Data from HPSP (AT&T) to Victim

Sender AS	Total Experiments (We ignore details of experiments in which HPSP fails to capture the data)	Average Path Length	Experiments in which HPSP can send the data directly	Exp. in which HPSP can send data to its neighbor	Exp. in which HPSP cannot send the data	Average path Length in Experiments where it can send the data (Sender -> HPSP + HPSP -> Victim/Neighbor)
AS18423	15	3.62	12	3	0	4.84
AS29520	15	5.4	14	1	0	6.22
AS3893	15	4.66	9	6	0	6.8
AS16484	15	3.62	13	2	1	5
AS6885	15	5.4	9	6	0	5.8
AS22321	15	4.00	8	6	1	4.6
AS1890	15	6.26	9	6	0	6.8
AS11299	15	5.6	14	1	0	6.2
AS18924	15	5.89	13	2	0	6.84
AS31223	15	6.5	14	1	0	7.6
AS22312	15	4.7	11	4	0	6.54
AS2299	15	5.34	15	0	0	6.75
AS5433	15	5.4	10	5	0	5.8
AS42332	15	3.8	13	2	0	5.2

Figure 28 shows the pseudo code used to perform such an analysis. We observe that without tunneling we can route the data 70% of the time from the HPSP to the Victim. However, using tunneling, we can route the data 98.09% of the time. The cost of such redirection is an average increase in path length of 2.07 AS hops. Table 5 summarizes the results of our experiments with routing the data from HPSP with AT&T as the HPSP, to the Victim. We observe that without tunneling we can route the data 78% of the time from the HPSP to the Victim. However, using tunneling, we can route

the data 99.52% of the time. The cost of such redirection is an average increase in path length of 1.057 AS hops. Figure 29 summarizes the results of experiments on routing data from HPSP to the Victim.

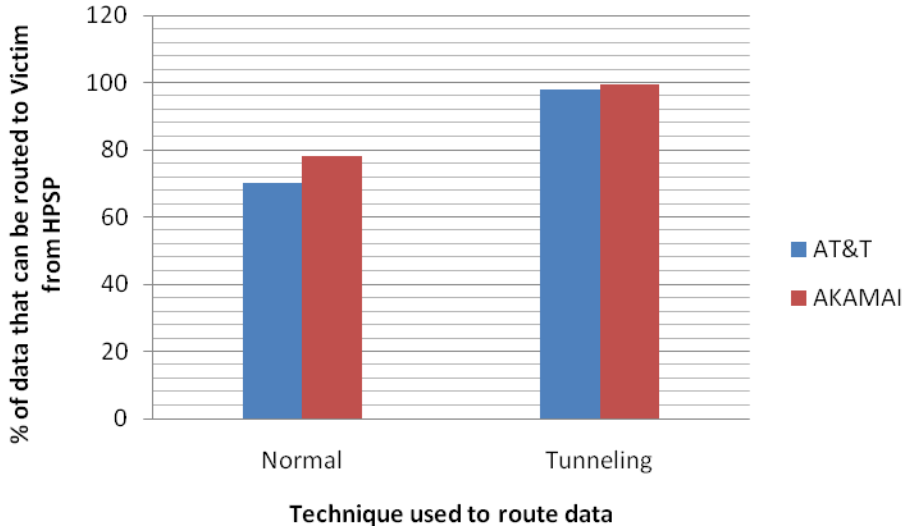

Figure 29. Summary of Routing Data from HPSP to Victim

We conclude that the illustrated HPSP framework is highly successful in preventing prefix hijacks and can be fine tuned and better improved by carefully choosing locations of HPSP ASs.

CHAPTER VII

CONCLUSION AND FUTURE WORK

In this research, we proposed and evaluated a framework for Hijack Prevention. While the analysis involved routing tables from Route Views and RIPE, the simulation cannot be 100% accurate unless we have the routing tables from Akamai and AT&T whom we used as HPSP. We plan to further investigate on schemes to route data optimally from HPSP to the Victim so that the average increase in AS hops is minimal. We would also like to extend our analysis by choosing other Internet Service Providers as HPSPs. Also, we would like to configure our scheme in Autonomous Systems to test real time performance.

BGP was designed in 1980's. It has definitely failed to address security issues in the current era. A transition to stronger and more secure Inter domain routing protocol is required. Such transition will require global cooperation among all Autonomous Systems. We believe our scheme can serve to mitigate prefix hijack attacks on BGP and act as insurance during hijack in this period of transition.

REFERENCES

[1] NANOG The Relationship between Network Security and Spam, http://www.nanog.org/mtg-0310/spam.html, 2008.

[2] L. Garcia, and I. Widjaja, "Communication Networks - Fundamental Concepts and Key Architectures," In *Packet Switching Networks*, 2nd ed., India, McGraw-Hill, pp. 490-492, 2004.

[3] Autonomous System (AS) in Internet - Wikipedia web page, http://en.wikipedia.org/wiki/Autonomous_system_(Internet), 2008.

[4] Internetworking Technology Handbook – Border Gateway Protocol (CISCO), http://www.cisco.com/en/US/docs/internetworking/technology/handbook/bgp.html, 2008.

[5] L. Gao, "On Inferring Autonomous System Relationships in the Internet," *IEEE/ACM Transactions on Networking*, vol. 9, no. 6, pp. 733-745, 2001.

[6] M. Lad, R. Oliveira, B. Zhang, and L. Zhang, "Understanding Resiliency of Internet Topology against Prefix Hijack Attacks," *Proceedings of the 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, 2007.

[7] J. H. Wang, D. M. Chiu, J. C. S. Lui, and R. K. C. Chang, "Inter-AS Inbound Traffic Engineering via ASPP," *IEEE Transactions on Network and Service Management*, vol. 4, no. 1, pp. 62-70, 2007.

[8] L. Swinnen, S. Tandel, S. Uhlig, B. Quoitin and O. Bonaventure, "An Evaluation of BGP-based Traffic Engineering Techniques," http://www.info.ucl.ac.be/people/OBO/papers/cost263-chapter.pdf, 2008.

[9] B. Quoitin, C. Pelsser, O. Bonaventure, and S. Uhlig, "A Performance Evaluation of BGP-based Traffic Engineering," *International Journal on Network Management*, vol. 15, no. 3, pp. 177–191, 2005.

[10] H. Ballani, P. Francis, and X. Zhang, "A Study of Prefix Hijacking and Interception in the Internet," *Proceedings of ACM SIGCOMM*, 2007.

[11] Tier1 Network -Wikipedia web page, http://en.wikipedia.org/wiki/Tier_1_carrier, 2008.

[12] CAIDA AS topology and rank, http://www.caida.org/research/topology/rank_as, 2008.

[13] Border Gateway Protocol – RFC, http://www.ietf.org/rfc/rfc4271.txt, 2008.

[14] A. J. Su, and D. R. Choffnes, "Drafting Behind Akamai (Travelocity-Based Detouring)," *Proceedings of ACM SIGCOMM*, 2006.

[15] C. Zheng, L. Ji, D. Pei, J. Wang, and P. Francis, "A Light Weight Distributed Scheme for Detecting IP Prefix Hijacks in Real Time," *Proceedings of ACM SIGCOMM*, 2007.

[16] X. Zhao, D. Pei, L. Wang, D. Massey, A. Mankin, S. Wu and L. Zhang, "An Analysis of BGP Multiple Origin AS (MOAS) Conflicts," *Proceedings of ACM IMW*, Oct. 2001.

[17] R. Mahajan, D. Wetherall, and T. Anderson, "Understanding BGP Misconfiguration," *Proceedings of ACM SIGCOMM*, Aug. 2002.

[18] A. Ramachandran, and N. Feamster, "Understanding the Network-Level Behavior of Spammers," *Proceedings of ACM SIGCOMM*, Sep 2006.

[19] YouTube Prefix Hijacking Incident – RIPE, http://www.ripe.net/ripe/meetings/ripe-56/presentations/Refice-YouTube_Prefix_Hijacking.pdf, 2008.

[20] K. Butler, T. Farley, P. McDaniel and J. Rexford, "A Survey of BGP Security," Technical Report TD-5UGJ33, AT&T Labs-Research, Florham Park, NJ, April 2005.

[21] The NANOG mailing list, http://www.merit.edu/mail.archives/nanog/, 2008.

[22] M. Lad, D. Massey, D. Pei, Y. Wu, B. Zhang, and L. Zhang, "PHAS: A Prefix Hijack Alert System," *Proceedings of USENIX Security Symposium*, 2006.

[23] Routing Information Service (RIS) - Raw BGP Data from RIPE NCC Projects, http://www.ripe.net/projects/ris/rawdata.html, 2008.

[24] Raw BGP Data from RouteViews, http://archive.routeviews.org/bgpdata/, 2008.

[25] MyASN Prefix Hijack Notification Detection and Service from RIPE, http://www.ris.ripe.net/myasn.html, 2008.

[26] X. Zhao, D. Pei, L. Wang, D. Massey, A. Mankin, S. Wu, and L. Zhang, "Detection of Invalid Routing Announcement in the Internet," *Proceedings of the IEEE DSN*, June 2002.

[27] R. Chandra, P. Traina, and T. Li, "BGP Communities Attribute – RFC," http://www.ietf.org/rfc/rfc1997.txt, 2008.

[28] K. Stephen, L. Charles, and S. Karen, "Secure Border Gateway Protocol (S-BGP)," *IEEE Journal on Selected Areas in Communications - Special Issue on Network Security*, vol. 18, no. 4, pp. 582-592, 2000.

[29] R. White, "Securing BGP through Secure Origin BGP," *Internet Protocol Journal*, vol. 6, no. 9, pp. 15-22, 2003.

[30] Z. Zhang, Y. Zhang, Y. C. Hu, and Z. M. Mao, "Practical Defenses against BGP Prefix Hijacking," *Proceedings of the ACM CoNEXT conference*, 2007.

[31] Z. Zhang, Y. Zhang, Y. C. Hu, Z. M. Mao, and R. Bush, "iSPY: Detecting IP Prefix Hijacking on My Own," *Proceedings of ACM SIGCOMM'08*, August 17–22, 2008.

[32] Straighten Route Views – Tool to Analyze/Cleanse BGP Routing Table from CAIDA, http://www.caida.org/projects/routing/atoms/download/rv2atoms-0.4/straightenRV.1.ps, 2008.

[33] R. Oliveira, D. Pei, W. Willinger, B. Zhang, and L. Zhang, "In Search of the Elusive Ground Truth: The Internet's AS Level Connectivity Structure," *Proceedings of ACM SIGMETRICS*, 2008.

VITA

Krishna Chaitanya Tadi received his B.E. degree in electronics and communication engineering from Jawaharlal Nehru Technological University, India in 2005 and his M.S. in computer engineering from Texas A&M University, College Station, TX, in 2009. His research interests include Network Security, Computer Communication Networks. Prior to joining Texas A&M, he worked as a Software Engineer in Infosys Technologies, India from 2005-2006.

Mr. Krishna Tadi may be reached at 214 Zachry Engineering Center, College Station, Texas 77843-3128. His email is tadi.krishna@gmail.com.