Stealthy IP Prefix Hijacking:

Don't Bite Off More Than You Can Chew

CHRISTIAN MCARTHUR
Department of Computer Science
Texas A&M University
cbmcarthur@tamu.edu

MINA GUIRGUIS
Department of Computer Science
Texas State University-San Marcos
msg@txstate.edu

***Abstract*— In prefix hijacking, an Autonomous System (AS) advertises routes for prefixes that are owned by another AS, and ends up hijacking traffic that is intended to the owner. While misconfigurations and/or misunderstandings of policies are the likely reasons behind the majority of those incidents, malicious incidents have also been reported. Recent works have focused on malicious scenarios that aim to maximize the amount of hijacked traffic from all ASes, without considering scenarios where the attacker is aiming to avoid detection. In this paper, we expose a new class of prefix hijacking that is stealthy in nature. The idea is to craft path(s) – of tunable lengths – that deceive only a small subset of ASes. By finely tuning the degree to which ASes are effected, the attacker can handle the hijacked traffic while the victimized AS would not observe a major reduction in its incoming traffic that would raise an alarm. We give upper bounds on the impact of those attacks via simulations on real BGP Internet announcements obtained from Route-Views. We discuss shortcomings in current proposed defense mechanisms against attackers which can falsify traceroute replies. We also present a defense mechanism against stealthy prefix hijacking attacks.**

I. INTRODUCTION

Scope and Motivation: The potential of prefix hijacking poses an ongoing threat to almost any Autonomous System (AS). In prefix hijacking, an AS advertises routes for prefixes that are owned by another AS, and ends up hijacking traffic that is intended to the owner. Recently reported incidents of prefix hijacking have been shown to be quite effective in hijacking traffic and disrupting services. One of the latest incidents occurred on Feb 24, 2008 when Pakistan Telecom advertised a prefix that belongs to YouTube and caused traffic destined to YouTube to be routed to Pakistan [1].

Prefix hijacking could occur in a number of different ways. First, administrators may incorrectly configure their border routers – via a typo or due to a misunderstanding of the local policies – which would cause invalid routes to be announced. Second, an attacker may compromise a border router and configure it to advertise invalid routes for specific prefixes. Finally, an attacker may intentionally desire to hijack a prefix (reasons are given below). In this case, the attacker could set up a legitimate peering relationship between their router and a provider, and send out invalid updates. If the provider has not configured local policies to reject those updates, the invalid routes would be distributed to other border routers around the Internet and the prefix would be hijacked. Throughout this paper, we treat the above cases as "hijacking attacks" irrespective of how they actually occurred.

An attacker may desire to hijack a prefix to perform a Denial of Service (DoS) attack against the true owner of that prefix. Since the attacker will be receiving the hijacked traffic, he/she can black-hole the traffic, or can sniff it for various types of information and utilize it for phishing [2]. Moreover, the attacker can reroute the hijacked traffic back to the owner, so that the owner would not observe a drop-off in its incoming traffic [3]. Another possibility is for the attacker to send anonymous spam e-mails through the hijacked prefix as reported in [4].

Detecting prefix hijacking and securing our routing infrastructure are crucial goals. There has been considerable efforts for developing methods to detect prefix hijacking [5]–[8] and to secure the BGP protocol [9]–[13], so that prefix hijacking would be harder to perform. However, these studies have focused on prefix hijacking attacks that are aiming to *maximize the number of ASes impacted without considering scenarios whereby attackers are aiming to avoid detection*.

But, what if the attacker's goal is not to maximize the number of affected ASes, but rather impact a small subset of ASes, so that (1) the attacker can handle the amount of hijacked traffic and (2) the victim would not observe a sharp drop-off in its incoming traffic that would raise an alarm? Such an attack can result in longer periods of hijacking traffic that can be potentially more damaging than hijacking traffic for a shorter period (until detected and mitigated).

Stealthy IP Prefix Hijacking: The main idea behind stealthy hijacking is *to tune the length of the invalid paths so that they are only appealing to a smaller fraction of ASes*. To achieve this, the attacker inserts his/her AS into pre-existing paths and announces them to his/her peers. In this manner, the crafted paths are long enough so that their effects will not be noticed by the victim's administrators, yet short enough to attract some fraction of the traffic intended for the victim. The exact fraction of ASes that are tricked in choosing the invalid paths depends on the length of the invalid path.

Contributions: We summarize our contributions in the following points:

- We expose a new class of stealthy prefix hijacking attacks that are harder to detect. They impact a small fraction of

ASes while the victim continues to receive traffic from the majority of ASes (raising no alarms).
- We illustrate different mechanisms to craft false BGP advertisements to tune/control the impact of the attack.
- We assess the vulnerabilities from stealthy prefix hijacking attacks via real Internet topologies obtained from Route-Views [14]. The results obtained give upper bounds on the impact of stealthy prefix hijacking attacks.
- We demonstrate shortcomings in current defense mechanisms against smart attackers who are stealthy and can falsify traceroute replies.
- Finally, we outline a defense mechanism that uses a combination of existing detection methods to successfully detect a stealthy prefix hijacking attack.

Paper organization: Section II covers background material, exposes our new class of stealthy prefix hijacking, and presents upper bounds on the impact of stealthy prefix hijacking attacks. We expose "fakeroute," a utility we created to falsify traceroute responses in Section III. We evaluate defense mechanisms in Section IV. In Section V we discuss related works. We conclude the paper in Section VI.

II. Stealthy Prefix Hijacking

In this section, we briefly cover background material on BGP and prefix hijacking. Then, we expose stealthy prefix hijacking attacks and study their upper bound impact.

A. BGP and Prefix Hijacking

The Internet is a collection of Autonomous Systems (ASes) that communicate through the Border Gateway Protocol (BGP) to exchange routing information. An AS will advertise the prefixes that originate from within to other ASes [15]. As this information gets propagated between ASes, a connectivity graph is formed [16]. This graph is used to determine how packets get routed through the various ASes toward their destinations. When two border routers are BGP peers, they exchange information about all routes they know for handling packets subject to their export filters, if any.

Border routers can be configured with local policies to filter routing announcements that they receive. Proper use of local policies can significantly reduce the threat from attackers. However, the history of prefix hijacks suggests that local policies are not being used on all border routers. Therefore, absent of any local policies specifying otherwise, advertised routes are accepted as facts and can be passed along to other routers. To determine which route to use, border routers evaluate all advertised paths using a series of rules [17]. Generally, apart from local policies specifying otherwise, the preferred path between any two ASes will be the shortest path between them for the longest matching prefix.

One problem that exploits the trusting nature of BGP is the prefix hijacking problem. In prefix hijacking, an AS advertises routes for prefixes that are owned by another AS, and ends up hijacking traffic that is intended for the owner. Currently there are three main approaches to hijack a prefix. One simple approach is for the attacking AS, X, to claim that it owns some prefix P. Thus it would advertise the AS-PATH = [X] for the prefix P. Such an advertisement is known as an *invalid origin* and can cause a great impact on Internet traffic [18]. Another approach is for the attacking AS to announce *a more specific prefix* than the one announced by its victim. This method can also have a great impact and was actually used by YouTube to respond to Pakistan Telecom's hijack and get their traffic back. Another approach is for the attacking AS, X, to claim that it is connected to an AS, Y, that owns the prefix P. Thus it would advertise the AS-PATH = [X, Y]. Such an advertisement is known as an *invalid next hop* [3]. It important to note that the above approaches tend to maximize the amount of hijacked traffic from all or most of the ASes on Internet.

B. Stealthy Prefix Hijacking

Aiming to maximize the amount of hijacked traffic may be problematic from a smart attacker's stand point. First, the attack is easily exposed. The victim will observe a significant decrease in its incoming traffic which would likely raise suspicion. Second, the attack puts strain on the resources used by the attacker, since they have to deal with a high volume of hijacked traffic. This may prevent the attacker from recording the traffic locally, or even re-routing it back to the victim (both of these cases could have been achieved if less amount of traffic was hijacked). Finally, the invalid origin or invalid next hop can be easily detected with the "Prefix Hijack Alert System" (PHAS) [6], as we discuss in Section IV.

Performing a successful stealthy prefix hijack requires that the advertisements should have less overall impact on the Internet. To do so, the attacker advertises an invalid route for the victim's prefix that has a longer length than may be preferred by the majority of ASes on the Internet. The exact length should be *long enough so that its effects will not be noticed by the victims' administrators, yet be short enough to attract some fraction of the traffic intended for the victim*.

To perform a stealthy prefix hijacking, the attacker carries out the following algorithm:

1) For a chosen victim, discover the AS tree where the victim is the root of the tree. This information is publicly available (e.g., RouteViews).
2) Determine which ASes in the tree you can peer with.
3) For each possible peer, estimate the potential effects of the hijack with differing lengths of invalid paths.
4) Establish a peering relationship with the desired AS.
5) Announce an invalid BGP route and analyze the amount of hijacked traffic.
6) Tune the length of the invalid path; If the amount of hijacked traffic is too much, advertise a longer route. If it is too small, advertise a shorter route.
7) Once the amount of hijacked traffic is desirable, continue to announce the route. Otherwise, select another potential peer and go to step 4.

Notice that Step 4 depends on the exact relationships between ASes (e.g., customer, peer, provider) [3]. Since it is

hard to predict the success of this step (as it depends also on who is the attacker), this paper assumes that the attacker is successful in this step, thus, our results give upper bounds on the impact of stealthy prefix hijacking.

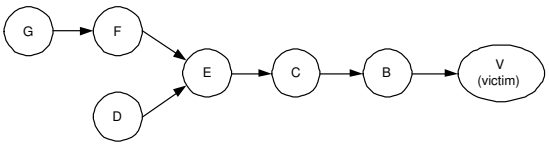

Fig. 1. An AS path tree.

To craft the exact announcements, the attacker inserts his/her own AS number into a pre-existing path and advertises this path. Figure 1 illustrates an example of a real AS tree (partial representation). The prefix being hijacked belongs to the AS node labeled "Victim," at the root of the tree. An attacker could peer with AS F and announce an AS-PATH = [Att B V] for the victim's prefix. Such an announcement suggests that the attacker is connected to AS B and can forward packets along the announced route to the victim.

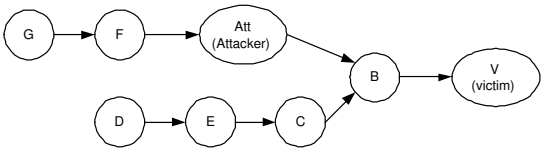

Fig. 2. An AS path tree under attack.

Fig 2 shows the possible effects of this invalid BGP announcement by the attacker. Since the attacker is peering with AS F, in our example, it is likely that AS F would see the shorter path to the victim through the attacker. As a result, AS F would forward packets destined for the victim to the attacker, trusting that the attacker is correct in his/her announcement. Traffic from AS G to the victim will also be hijacked since AS G forwards its packets through AS F.

C. Upper bounds on the Impact of Stealthy Prefix Hijacking

To assess the impact of stealthy prefix hijacking attacks, we utilize BGP routing information obtained from RouteViews [14]. Snapshots of the views of the Internet as seen by RouteViews observer routers were made from March 4, 2008. These views were combined to create a tree of AS paths for every prefix seen by those observer routers. There were more than 250,000 AS trees that we evaluated in our experiments.

Each AS in the tree was evaluated to determine if traffic from the AS to the victim could be hijacked. This was accomplished by examining the length of the current path from each AS to the victim. This length was compared to possible BGP announcements that could be sent to the AS with varying lengths from 0 to 10. In the event that the length of the existing route was the same as the length of the invalid update, preference was given to the existing legitimate route.

The effects of stealthy prefix hijacking attacks can be seen in Figure 3. The graph shows two bars for invalid path lengths from zero (an invalid origin attack) to ten. The bar on the left

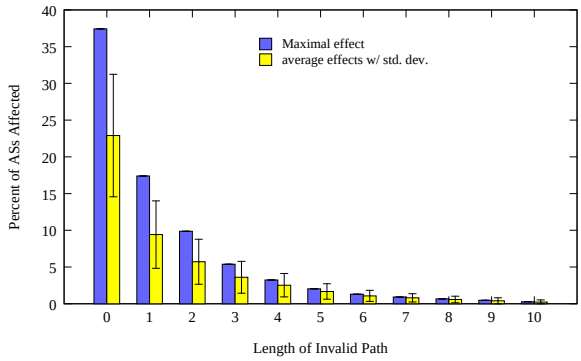

Fig. 3. Effects of single stealthy attack

represents the maximal effect on the percentage of affected ASes (the case of peering with the AS that would result in the maximum damage). The bar on the right represents the average (along with standard deviation) percentages of affected ASes (averaged over all ASes the attacker is peering with).

If an attacker performs an invalid origin attack (announces a BGP path of length zero), the maximum percentage of ASes that could be influenced is around 37%. The attack could affect on average almost 23% of ASes. Since our goal with stealthy prefix hijacking is to affect a small number of ASes, this graph helps to show the possible lengths that can achieve our goal. The results show that announcing an invalid path of length 4 or greater will affect less than 5% of ASes both maximally and on average. As the path length grows to 6 or more, the effects of the attack drop to 1% or less of ASes.

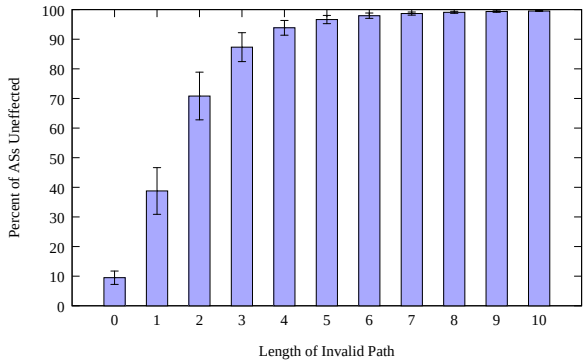

Fig. 4. ASes unable to be effected.

To assess the extent to which ASes *cannot* be affected by stealthy prefix hijacking attacks, Figure 4 shows the average percentage of ASes that cannot be affected under all possible attacks. As expected, if an attacker announces an invalid origin update (zero path length) very few ASes will not be affected. These unaffected ASes will usually be the true origin AS for the prefix and the ASes that are directly connected to the true origin AS. As the path length increases, more ASes will not be affected as they will already know of a shorter path than the invalid one.

D. Discussion

Multi-homed Attackers: In the above stealthy attack, we assumed that the attacker has only one peering point to a provider. If the attacker is multi-homed, he/she can send

multiple advertisements with different lengths to each one of his/her peering points for the same victim. Such an attack gives more control over the number of affected ASes. To analyze the effects of a dual attack (the attacker has two peering points), we ran simulations in which after determining the effects of the first attack, we evaluated the effects of a secondary attack on the AS tree. We continued to look for upper bounds and the results are shown in Table I. A dual attack can potentially affect a greater percentage of ASes than a single attack.

		Initial Attack Path Length						
		1	2	3	4	5	6	7
Dual Attack Length	2	30.30						
	3	29.61	18.23					
	4	29.19	18.03	10.13				
	5	29.05	17.90	10.03	6.17			
	6	29.98	17.86	9.97	6.12	3.83		
	7	28.93	17.84	9.95	6.10	3.80	2.49	
	8	28.93	17.83	9.94	6.09	3.78	2.47	1.78

TABLE I
PERCENT OF ASES AFFECTED BY DUAL ATTACKS

Notice that an attacker can create its own table (similar to Table I) and use it as a *look-up table* to determine the path lengths to announce. Notice also that even though the table shows little changes from varying the lengths of the secondary attack, the effects can still be significant. A change of 0.001% results in more than 65 actual affected ASes.

Chosen-AS attack: In this variant of stealthy prefix hijacking, the attacker aims to hijack traffic *from* a particular AS that is destined to the victim AS. In this situation there are two "intended" victims, the source-of-traffic victim and the regular destination victim. This attack can be achieved since the attacker has access to many publicly available BGP routing data (like the ones we use in this paper). The attacker could form an AS routing tree for the destination victim, then selects a downstream provider for the source victim to peer with and advertises BGP announcements through it. Any traffic destined to the destination victim from the chosen provider and ASes upstream from it, including the source victim, will be hijacked.

Amounts of Hijacked Traffic: While the results above give upper bounds on the percentages of affected ASes, they do not translate directly into amounts of hijacked traffic. This is so because different ASes have very different characteristics in terms of the amount of traffic they produce/carry (e.g., tier-1 AS versus tier-3 AS). As it turns out, accounting for the exact amounts of hijacked traffic (whether in a stealthy manner or not) is very hard to achieve since there is no information available on how much data each AS generates/carries. Our initial investigation – using data collections from iPlane [19] that estimate the capacities of access links for different /24 prefixes – shows that access links' capacities across ASes tend to have a heavy-tail distribution which would enable an attacker to find/tune/control a larger number of ASes with relatively less traffic to impact.

III. FAKEROUTE

Many of the methods used in detecting prefix hijacking (described in Section IV) use traceroute to look for discrepancies in the routing process. To combat these methods, we present a utility we call "fakeroute that falsifies traceroute replies, adding a significant layer of stealthiness. The tool is a multi-threaded application implemented in C and uses raw sockets to respond to traceroute requests for any IP address that belongs to the hijacked prefix from the victim's AS.

We assume that the attacker, *before* hijacking a prefix, would perform a traceroute to the intended victim, to learn about the ASes and the routers along the legitimate path to the victim. The attacker would also learn about the timing characteristics (delay on the links) of the path. The information will be used by fakeroute to respond correctly (with the corresponding IP and round-trip time) to any request. Fakeroute will spoof the IP source in its outgoing replies. We assume that the attacker knows the round-trip time between itself and its peering point.

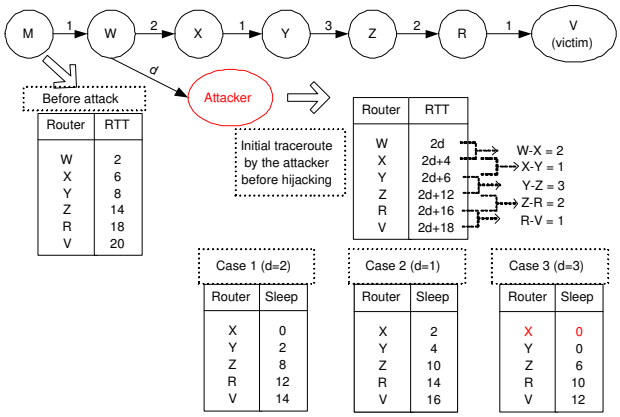

Fig. 5. Description of the fakeroute tool.

Figure 5 shows a general scenario for fakeroute. The delay on the links correspond to the one-way delay between routers. For simplicity, we assume that the links are symmetric, so the round-trip time on a link is twice the one-way delay. The figure also shows the initial traceroute to the victim from M and from the attacker before the hijacking occurs. Notice that this information enables the attacker to infer the delay between any two hops along the path to the victim. Once the hijack occurs, the attacker needs to respond to all traceroute queries in the correct time, in order to bypass detection from M. Ideally, the response from the attacker should be as close as possible to the traceroute obtained by M before the attack.

There are three possible cases. The first case is when both links (W-Att and W-X) have relatively the same delay. In this case the attacker responds without any waiting time for router X and would delay sending responses (using the usleep() function) for any other router by twice the one-way delay between that router and X. This case is illustrated in the figure when d is equal to 2. The second case is when the delay on (W-Att) is shorter than the delay on the link (W-X). In this case the attacker needs to sleep more time (the time is exactly twice the difference between the one-way delay on W-Att and W-X)

for router X and would add this initial delay to all subsequent replies. This case is illustrated in the figure when d is equal to 1. The third case is when the delay on (W-Att) is longer than the delay on the link (W-X). In this case the attacker would lag behind in its responses (its first few replies only). The best strategy here is to send immediate replies for all routers which have one-way delay to W that is smaller than the delay on (W-ATT). Then the attacker would resume delaying by twice the one-way delay for subsequent routers. This case is illustrated in the figure when d is equal to 3.

To test fakeroute in a possible hijacking scenario, we configured a static route at a monitor machine that would forward all of its traffic (intended to the victim) to the attacker machine. At the attacker machine, we used iptables to redirect incoming traffic from the monitor to fakeroute (by overriding the destination address from an IP that belongs to the victim to the attacker's local machine IP, otherwise, the kernel would drop the packets). Then, our application responds back to the monitor with the replies of its choosing. The responses from fakeroute could not be differentiated from legitimate responses. It is also possible for the attacker to choose the IP addresses of the intermediate routers to correspond to ASes of its choosing and to insert extra routers along the way.

IV. Defense Mechanisms

In this section we evaluate current detection methods in their capabilities to detect stealthy prefix hijacking. We then outline our proposed defense mechanism.

A. Current Defense Mechanisms

PHAS: The "Prefix Hijack Alert System", PHAS [6] analyzes BGP updates from RouteViews for prefixes that are registered with their service. It will raise an alarm (e.g., email the administrator) in the case of invalid origin, sub-prefix attacks, and invalid next hop attacks. Based on postings on the NANOG mailing list [20] from the authors of PHAS, it appears that PHAS cannot detect these kinds of stealthy attacks, since the attacker would insert its AS further out from the victim's AS.

IP Traceroute and Hop Count Changes: The authors in [8] propose a light weight scheme to detect prefix hijacking by having monitors distributed throughout the Internet that track the number of hops between themselves and possible victims' prefixes over time. Should the number of hops "exhibit significant differences", then the system would raise an alarm. Since the attacker is hijacking traffic, it could defeat this method by intercepting and responding to traceroute requests sent by the monitor. The result is that the monitor will get a match on the hop count, before and after the attack. With the "fakeroute" tool, an attacker can respond to traceroute requests with any set of routers it desires (along with fake timing on responses that match the old route's characteristics). Moreover, the authors did not specify a specific detection threshold, but suggested that it needs to be large enough not to cause false alarms due to the normal temporary routing changes that occur, which in turn give the attacker more room to evade detection.

Traceroute Path Disagreement: The authors of [8] also provide a hijack detection method that relies on comparing a possible route to the victim with another route to a reference point. The reference point is chosen to be as close as possible to the victim, yet outside the victim's prefix (hence traffic for it should not be hijacked). To combat traceroute path disagreements, the attacker would falsify traceroute replies with "fakeroute" to make it appear to the monitor that no routes has changed (since the attacker already learned the original route along with its timing information).

AS Traceroute: AS Traceroute [21] takes traditional traceroute listings of routers and map the routers' IP addresses into the corresponding AS numbers. To detect prefix hijacking, IP traceroute is converted into an AS path and compared to the BGP routing data for any discrepancies [7]. Again, an attacker who is able to falsify traceroute responses can ensure that they match the BGP paths that have been announced (since the attacker is choosing the IP addresses of the routers in his/her responses). Notice that with such false responses, an attacker can also defeat the method suggested in [7] whereby invalid links can be recognized by examining the physical distances between the two ASes in the new announcement.

Heuristic Based Detection: In [5], observers are set up on the Internet with access to the BGP routing data. First, the observers form, for a prefix, a list of valid links and a list of valid origin ASes for a period of time. Once this "learning" phase is over, the observers will examine any new links or origin announcements for legitimacy. Any announcement made that does not have a match in their lists becomes a suspect. Then, they apply some heuristics to decide whether the new, previously unknown, announcement is valid or not. While the stealthy hijacking attempt may raise an alarm in the short term, we believe that on the long term hijacking attempts could be successful. The authors offer suggested thresholds of 24 hours of uptime over the last 30 observed days to accept a new link in their database. So the attacker could announce a longer path to the victim than is currently being reported that includes its AS number. Since the path is long it will not be flagged. After the path has been announced for 24 hours, the attacker could change the route to be a shorter one. Since the attacker's AS number is in the observer's tables for that prefix, an alarm will not be raised.

iSPY: In iSPY [22], the victim continuously performs traceroute to transit ASes and looks for changes. iSPY analyzes the number of cuts. A cut is defined as a link in the path that becomes unreachable, but links further down the path may still be reachable. iSPY relies on a large number of distinctive cuts as a signature, something that stealthy prefix hijacking aims to prevent from happening. Thus iSPY may confuse a stealthy prefix hijacking as just a minor link failure. Moreover, with fakeroute, an attacker may still respond with traceroute replies to the victim to match the original path.

B. Effectively Detecting Stealthy Prefix Hijacking

Our proposed method to detect stealthy prefix hijacking combines the traceroute path disagreement [8] method with the

AS Traceroute tool [21]. In order to avoid having discrepancies between the hijacked path and the BGP route, the attacker must provide a traceroute path in which the IPs of the routers in the traceroute translate to AS numbers that match the BGP route. We show below that this is very hard to achieve.

A monitor would typically find that the traceroute response matches the BGP route to the victim. If the monitor performs a traceroute to a reference point (close to the victim), it will get the legitimate path to the reference point. There will be a disagreement in the paths from the monitor to the reference point and from the monitor to the victim. Thus an alert would be raised. The attacker could respond to traceroute requests such that the path to the victim would match the route to the reference point. However, in this case, the IPs for each hop of the traceroute, when translated to AS numbers, will show an AS path that differs from the announced BGP path. Again, an alert would be raised for this case. Therefore, by employing these two methods of detection, path disagreement with a reference point and mapping the IP traceroute to the BGP route, stealthy prefix hijacking has become detectable.

V. Related Work

In the previous section, we examined the current efforts for defense mechanisms in detecting prefix hijacking attacks, here we focus on other related works.

The authors in [18] studied the effects of an invalid origin attack (where the attacker announces the ownership of the victim's prefix). Their results showed that some ASes can be greatly impacted by hijacking. With an invalid origin attack, more than 10,000 networks may be affected in many cases. They also advocated, for resiliency, that ASes would best protect themselves by being multi-homed to multiple tier-1 providers. The authors in [3] examined the effects of an invalid next hop attack. The study showed that if a tier-1 AS were to attempt to hijack a prefix, it would be successful with a probability up to 75%. The authors demonstrated, not just hijacking traffic from the victim, but also the feasibility to reroute the hijacked traffic back to the victim. Our work here is different from the above studies since our focus was on stealthy hijacking attacks that aim to avoid detection.

There have been a number of efforts to secure the BGP protocol against prefix hijacking attempts [11]–[13], among others. The main goal is to provide a cryptographic mechanism for signing and verifying BGP announcements. This requires significant changes to the border routers to support this functionality as well as agreement on a repository of cryptographic keys. Other works [9], [10] are examples of non-cryptographic methods for preventing invalid announcements from propagating and prefix hijacks from occurring. These proposals also require changes to border routers and BGP.

VI. Conclusion

To the best of our knowledge, we believe this work is the first to expose an emerging class of prefix hijacking that aims to impact a smaller number of ASes in order to evade detection while enabling the attacker to keep up with the hijacked traffic. Based on our assessment with real Internet topologies, we have determined that it is possible for an attacker to tune the degree of control over the percentage of ASes that are affected. Moreover, with the proper planning and execution, the attacker can even choose the affected ASes. We have also exposed the "fakeroute" tool that falsifies traceroute replies to evade detection by a number of proposed detection methods. Finally, we have outlined a defense method that would "raise the bar" for an attacker to mount a successful stealthy attack. We believe that this work highlights the importance of putting into action effective methods to mitigate prefix hijacking attacks (especially stealthy ones) and to ensure a secure cyber-infrastructure.

References

[1] RIPE, "YouTube Hijacking: A RIPE NCC RIS case study," web: http://www.ripe.net/news/study-youtube-hijacking.html.

[2] O. Nordstrom and C. Dovrolis, "Beware of BGP Attack," *Computer Communication Review*, vol. 34, no. 2, pp. 1–8, April 2004.

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

[4] A. Ramachandran and N. Feamster, "Understanding the Network-level Behavior of Spammers," in *Proceedings of SIGCOMM*, Pisa, Italy, Sep 2006.

[5] J. Qiu, L. Gao, A. Ranjan, and A. Nucci, "Detecting Bogus BGP Route Information: Going Beyond Prefix Hijacking," in *Proceedings of SecureComm*, Sep 2007.

[6] M. Lad, D. Massey, D. Pei, Y. Wu, B. Zhang, and L. Zhang, "PHAS: A Prefix Hijack Alert System," in *Proceedings of 15th USENIX Security Symposium*, 2006.

[7] Xin Hu and Z. Morley Mao, "Accurate Real-time Identification of IP Prefix Hijacking," in *Proceedings of the IEEE Symposium on Security and Privacy*, Oakland, CA, May 2007.

[8] C. Zheng, L. Ji, D. Pei, J. Wang, and P. Francis, "A Light-Weight Distributed Scheme for Detecting IP Prefix Hijacks in Realtime," in *Proceedings of SIGCOMM*, Kyoto, Japan, Aug 2007.

[9] J. Karlin, S. Forrest, and J. Rexford, "Pretty Good BGP: Improving BGP by Cautiously Adopting Routes," in *Proceedings of IEEE ICNP*, Santa Barbara, CA, Nov 2006.

[10] T. Wan, E. Kranakis, and P. Oorschot, "Pretty Secure BGP, psBGP," in *Proceedings of NDSS*, San Diego, CA, Feb 2005.

[11] Y. Hu, A. Perrig, and M. Sirbu, "SPV: Secure Path Vector Routing for Securing BGP," in *Proceedings of SIGCOMM*, Portland, OR, Aug 2004.

[12] S. Kent, C. Lynn, and K. Seo, "Secure Border Gateway Protocol (S-BGP)," in *Proceedings of NDSS*, San Diego, CA, 1999.

[13] L. Subramanian, V. Roth, I. Stoica, S. Shenker, and R. Katz, "Listen and Whisper: Security Mechanisms for BGP," in *Proceedings of NSDI*, San Francisco, CA, Mar 2004.

[14] RouteViews, "University of Oregon Route Views Page," web: http://www.routeviews.org.

[15] J. Hawkinson and T. Bates, "RFC 1930: Guidelines for Creation, Selection and Registration of an Autonomous System (AS)," 1996.

[16] Y. Rekhter, T. Li, and S. Hares, "RFC 4271: A Border Gateway Protocol 4 (BGP-4)," 2006.

[17] CISCO, "BGP Best Path Selection Algorithm," web: http://www.cisco.com/warp/public/459/25.shtml.

[18] M. Lad, R. Oliveira, B. Zhang, and L. Zhang, "Understanding Resiliency of Internet Topology against Prefix Hijack Attacks," in *Proceedings of DSN*, Edinburgh, UK, Jun 2007.

[19] iPlane, "iplane web site: Datasets," web: http://iplane.cs.washington.edu/data.html.

[20] NANOG, "NANOG Mailing list," web: http://www.merit.edu/mail.archives/nanog/.

[21] Z. Mao, J. Rexford, J. Wang, and R. Katz, "Towards an Accurate AS-level Traceroute Tool," in *Proceedings ACM SIGCOMM*, karlsruhe, Germany, Aug 2003.

[22] Z. Zhang, Y. Zhang, Y. Hu, Z.Mao, and R. Bush, "iSPY: Detecting IP Prefix Hijacking on My Own," in *Proceedings of SIGCOMM*, Seattle, WA, Aug 2008.