Cross-Modal Vulnerabilities: An Illusive form of Hijacking

Eric Osterweil
UCLA
eoster@cs.ucla.edu

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

Christos Papadopoulos
Colorado State Univeristy
christos@cs.colostate.edu

ABSTRACT

Content, connection, and other types of hijacking are a common occurrence in today's Internet. One can broadly classify various types of hijacks as being *locally scoped* to an administrative domain, or pushed *externally*; where one administrative domain (intentionally or unintentionally) hijacks users in other domains. Current work in identifying and reacting to various types of Internet hijacking has focused on the network control plane and has not included *cross-modal* hijacks that involve both the control plane *and* the data plane of the Internet. In this work we introduce the idea that cross-modal threats exist in the Internet and form a highly illusive, but serious threat. Further, we detail an actual instance of Internet-scale cross-modal hijacking whose behavior depends on both network control data and data plane such as the order in which users request connections. Based on anecdotal evidence gleaned from several websites, it appears that this hijack existed for many months (and possibly years) before its recent detection.

1. INTRODUCTION

In the context of Internet systems and protocols, the term "hijack" often refers to the situation in which an entity in the network attempts to misdirect clients away from a specific data source and towards another. For example, a hijacker may want all web traffic bound for CNN's website to be directed to a webserver that she controls in order to plant false news headlines. In fact, there are many reasons why this behavior occurs and multiple ways in which users can be hijacked. We broadly classifying the *scopes* of these hijacks into two conservative categories: 1) sometimes administrative authorities exert *local* control over their users and 2) sometimes an authority will initiate a hijack in which control information is pushed out, and whose effects spread beyond the scope of that authority's domain, whereby the victims are *external* to the administrative domain. In each case, there may be different reasons, and different justifications for this behavior (including accidental misconfigurations).

There is a great deal of anecdotal evidence that demonstrates the existence of type 1 (or local) hijacking. Examples include hotels that rewrite DNS responses for their guests, WiFi hotspots that intercept IP traffic from customer laptops until they pay, universities and corporations that block access to websites that have been indicted of hosting "illicit" data, and the list goes on. The judgment of whether this behavior is appropriate is a policy issue, because it relates to how authorities manage their own domains. Thus, we neither impugn it, nor condone it here. However, there is room for a technical judgment whenever this rewriting spreads beyond the local administrative boundary that is orchestrating it. In this case, an authority may begin hijacking *other people's clients*, and this becomes a type 2 hijack.

In a type 2 hijack, an authority is systemically advertising the effects of their rewriting to reach a global audience. These cases are often easy to detect because an authority will issue control information that advertises their hijacked data. For example, in BGP prefix hijacking an Autonomous System (AS) falsely claims that it is the origin of an IP prefix. One of the most widely publicized prefix hijacks came in 2008 when Pakistan began hijacking YouTube's IP prefixes [9]. In this case, Pakistan allowed their type 1 local rewriting behavior (from their authoritative domain) to leak out to the rest of the world (thus making this a type 2 hijack).

In this work we identify a 3^{rd} type of hijack that is more illusive, but just as real and whose effects can be just as detrimental. We call this type 3 hijack a *Cross-Modal* threat because its effects are not seen without interactions between both the control plane and the data plane. We contend that this type of hijack is a previously under appreciated threat to both clients who may become hijacked and authorities whose clients may be unable to reach them and not understand why. It is particularly pernicious because while it is entirely deterministic (i.e. it constantly exists in the network) its effects are intermittent. This complexity makes its detection and correction more difficult too, because it doesn't necessarily affect all communications and it can come from an entity somewhere in the middle of the network. Many infrastructure monitoring systems [11, 7, 6] are designed to detect hijacking threats. However, without a notion that the data plane can be responsible for directing vulnerable traffic over a type 1 link, these approaches cannot identify cross-modal hijacking.

We go on to illustrate this type of threat by examining an actual instance of it. Early in 2010, the existence of a cross-modal attack was discovered to exist in the Asia-Pacific region of the Internet. In this particular case, the cross modal interaction between BGP's anycast service for the DNS roots, and DNS' lookup strategy lead directly to a cross-modal vulnerability for *only* certain DNS referrals. We explain the details of this instance later in the text, but based on anecdotal evidence gleaned from several websites [2, 1, 4], it appears that this hijack existed for many months (and possibly years) before its recent detection.

The remainder of this paper is organized as follows: in Section 2 we describe details of DNS and BGP that are useful in understanding our measurements. Next, in Section 3 we describe how the cross-modal hijack can occur between DNS, BGP, and user behavior. Then we discuss the effects of an actual cross-modal attack in Section 4 before concluding in Section 5 where we also outline our future work.

2. BACKGROUND

In this Section we first give a brief summary of how the Domain Name System [8] (DNS) is structured and how it manages its zone delegations. Next we give an overview of the Border Gateway Protocol [12] (BGP) and describe the way in which some critical DNS zones use it for redundancy and load balancing.

DNS.

The DNS is the de facto name mapping service for the Internet. All data in the DNS is stored in a data structure called a *Resource Record* (RR), and each RR has an associated name, class, time-to-live (TTL), and type. For example, an IPv4 address for `www.ucla.edu` is stored in an RR with name `www.ucla.edu`, class IN (Internet), and type A (IPv4 address). When a DNS resolver (a client) queries for a name the reply will come with data and a TTL value. The purpose of the TTL is to inform DNS caching resolvers how long they may cache data before they must flush it.

The DNS is a distributed database organized in a tree structure. At the top of the tree, the root zone delegates authority to *top level domains* like `.com`, `.net`, `.org`, and `.edu`. The zone `.com` then delegates authority to create `google.com`, `.edu` delegates authority to create `ucla.edu`, and so forth. In the resulting DNS tree structure, each node corresponds to a *zone.* Though the DNS is organized hierarchically, the "zone cuts" that divide names into zones are dynamic and not pre-specified. That is, the distinction that `www.cs.ucla.edu` is in the zone `cs.ucla.edu` and *not* in `ucla.edu` is not knowable a priori. Rather, resolvers that are walking the DNS must initially presume that *all* domain names belong to the root zone. Thus, initially, a resolver will send its first query (for example) `www.facebook.com` to the root zone. The root zone will then respond with a *referral* to the `.com` domain. In this way, the root says, "I don't know who `facebook.com` is, but `.com` may know. Try sending your query there..." In this way, the resolver learns the DNS hierarchy and will send queries for all `.com` subdomains there instead of the root[1].

From this we can see that the DNS root zone is of critical importance to DNS, because it is the starting point for all resolvers. To bolster its availability and reliability it has always been served by 13 separate name servers (called instances `A` through `M`), and great care has been taken to distribute them globally for performance and defense against attacks. In the last 10 years, part of the resilience of this distribution has been augmented by routing techniques (described below).

BGP.

BGP is the Internet's de facto routing protocol. It is a *path vector* routing protocol in which each routing entity is called an Autonomous System (AS). It is similar to distance vector protocols, in that each AS floods announcements for the blocks of IP addresses (called prefixes) that it is the *origin* for. These flooded announcements are relayed hop-by-hop and at each stage, the relaying AS appends itself to the *AS-path.* Thus, whenever an AS sees an announcement for a prefix, it can determine i) if there are any loops, and ii) if the path is shorter (or in some way preferable) to any existing paths it has seen for a prefix.

One of the tools people use in BGP is called *anycast.* The name comes from the same nomenclature as unicast (meaning point to point), broadcast (one point to all points), and multicast (one point to many points). In this vein, anycast means one point to one of many points. The basic idea is that separate instances of an AS that are distributed throughout the network announce the same prefix. All other ASes will implicitly choose the instance that is topologically closest. Thus, anycast acts as a form of topological load balancing and allows redundant instances of an AS to exist simultaneously.

3. A CROSS-MODAL THREAT TO DNS

The coupling between DNS and BGP anycast has greatly aided DNS by enhancing its performance and adding resiliency against some types of attacks (such as DDoS) without over complicating its design. However, the benefits of this coupling have also come with the liability of a cross-modal vulnerability.

[1] We note that the DNS protocol could technically tolerate having `www.facebook.com` in the root zone.

3.1 A Hijack from a Root Server

In March 2010, researchers noticed that from certain locations in the Internet, at certain times, there seemed be an intermittent hijack of well known websites. At that time, reports started surfacing that *some* resolvers in the Asia-Pacific region were resolving `facebook.com` to an invalid IP address.

In this particular case, when resolvers that were located in certain parts of the Internet issued queries to one of a few specific DNS root servers, their queries were occasionally directed to an anycast instance that was located in China. When these resolvers happened to issue queries for certain DNS names they would receive a set of spoofed response messages. However, what made this problem illusive was the following: i) spoofed responses were not issued to all queries, ii) they were not issued from all root servers, and iii) they were not issued to all resolvers in all locations. Thus, initial reports were met with confusion and some skepticism as the problem did not appear to be reproducible. Indeed, anecdotal evidence on various websites [2, 1, 4] suggests that this sort of cross-modal hijacking may have existed for some time before being properly diagnosed in March 2010.

The nature of this hijack is particularly illustrative of a cross-modal vulnerability because it not only requires a specifically well-timed set of events, but also requires events that occur in both the data plane and in the control plane. Yet, the events do result in an effective hijack of their target DNS names, and at a significant enough frequency that after being diagnosed, the operators of one of the root zone's anycast instances in China (the `i` root) took the preventative step of offlining their instance to prevent further spoofing.

It is important to note that if this were a systemic operation in just the control plane (i.e. its effects were visible without needing actions from the data plane) then operators would quickly detect it and take some form of corrective action. The gravity of cross-modal hijacking comes from the fact that it is illusive and tends to only be apparent under certain conditions.

3.2 How it Worked

The first element of this hijack was that there are ASes in the Internet acting as type 1 hijackers, or large Man-in-the-Middle (MitM) agents, and whenever they observe DNS queries for certain names (`facebook.com`, `twitter.com`, `xxx.com`, `youtube.com`, and `thepiratebay.org`) coming from within their authoritative domain they will rewrite them. Specifically, the ASes in this attack are located along the border of the Chinese authoritative domain. While this behavior is discussed on various mailing lists and on certain blogs (and is sometimes referred to as "the Great Firewall of China"), what is less well known is that many of these ASes will rewrite responses to queries that come from *outside* the Chinese authoritative domain as well as from inside.

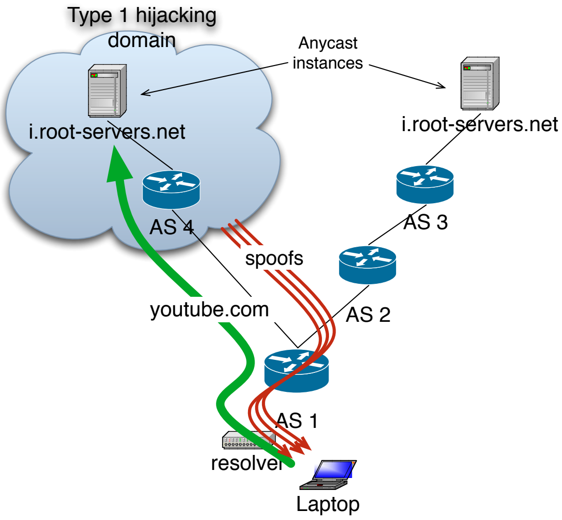

Figure 1: An example of how a short AS-path and a specific user query can elicit a hijack.

The second element of this hijack is that these type 1 ASes provide BGP transit services and route traffic from outside of China to destinations inside of China, which (in of itself) is not necessarily a concern. That is, unless the traffic includes DNS queries for the targets of the AS' type 1 spoofing.

This begs the obvious question of why clients outside of the Chinese authoritative domain would issue DNS queries in to it, and if they did, why would type 1 spoofing be unexpected (in other words, why is this a type 3 hijack instead of type 1)? This behavior comes from semantics inside the data plane, and is the final element in this hijack. Under certain circumstances, resolvers will issue queries for names like `www.youtube.com` to a root server, and sometimes that root server is in China.

In order to illustrate this, we begin by assuming a caching resolver whose cache is currently empty[2]. Recall from Section 2 that when the resolver needs to lookup any name, it begins by sending the *full* query (say, `www.youtube.com`) to one of 13 DNS root servers (`{a-m}.root-servers.net`). From this starting point, we can now show how there exists a vector for a cross-modal hijack, involving the control planes of both DNS and BGP, and the data plane of user requests. If i) the resolver does not have the `NS` records for either `youtube.com` or `.com`, ii) the next query the resolver sends is for `youtube.com`, iii) the query is sent to a root server that has an anycast instance in china (`{f,i,j}.root-servers.net`) *and* iv) BGP happens to route that resolver's query through a type 1 AS, then the re-

[2] This could be because it just started up with a cold cache, or it just timed out relevant RRs.

sult is that that resolver (who is perhaps caching for a large number of clients in a large ISP) will believe that `youtube.com`'s IP address is the value included in the spoofed response.

Consider Figure 1: here a user who is hosted by an ISP (AS 1), issues a request to their local resolver. In this case the resolver's cache is empty and it chooses to send the query to the `i` root server. From this ISP the shortest path to the `i` root server is $AS1 \rightarrow AS4$ (rather than $AS1 \rightarrow AS2 \rightarrow AS3$). Therefore, the DNS request is sent over the perimeter of a type 1 hijacking domain, and the result is a set of spoofed responses.

In the above example, every component is acting correctly, according to its design: a user just happened to issue a specific query at a point when his/her DNS caching resolver did not know who the authority was. Then, DNS did what it was supposed to do and sent its initial priming query to the only authority it knew at the time (the root). Similarly, BGP simply did what it was supposed to and sent the IP packet to the nearest anycast instance of the AS who hosts the given root server. Yet, the net result was that for the duration of that response's lifetime (the DNS `TTL` of the RRs), that resolver and all clients that use it will fall victim to a hijack. Later, if an operator tries to reproduce the problem, cache entries, routing dynamics, or simply choosing a different root server will make it difficult to re-observe the phenomenon.

4. THE EFFECTS OF THE HIJACK

We begin our investigation by examining routing behavior for six months between December 2009 and May 2010. These dates were selected because they include at least part of the period in which this hijack was observed [2, 1, 4] and some time after. We first examine the control information from BGP, collected from RouteViews [10]. Then we launched a set of active measurements of various candidate ASes, and their announced prefixes. From these measurements we analyze the rate and number of responses, and also the various aspects of the response packets themselves. In the remainder of this section we summarize some of the specific behavior of this specific cross-modal hijack. The goal of this presentation is simply to illustrate this *specific* cross-modal hijack's behavior and does not necessarily dictate the behavior of any *other* instances of this type of attack that may exist.

Control Information.

Since 2009 there have been three anycast instances of the DNS root in China: `{f,i,j}.root-servers.net`. This means that any AS who has a shorter AS-path to the instances of these servers in China may have a cross-modal vulnerability if there are *also* any type 1 spoofing ASes in its path. Thus, our analysis begins by studying the AS-path information from multiple RIBs, which were collected during this period from monitors in the RouteViews [10] deployment. From these RIBs we identified AS-paths that lead from any monitor to any of the DNS root anycast instances in China, and from these paths we selected those ASes whose registration information [5] identified them as being Chinese. This list of ASes formed our candidate set of possible hijacking ASes. Our candidate set was comprised of 14 ASes and six of these report to be NICs, backbones, or other infrastructure ASes (for example, ASN 4837 "CHINA169-BACKBONE CNCGROUP China169 Backbone").

Rates and Distribution of Responses.

Our goal in the active measurement component of this analysis was to test which ASes send spoofed responses to DNS queries that cross their perimeters. To do this we wrote a simple tool called `jacksniff` that uses the high-speed parallel framework in `libvdns` [3] to issue DNS queries of any name we chose across the range of addresses in an IP prefix. The tool then uses `libpcap` to observe the IP packets returned (if any). We ran `jacksniff` against each prefix announced by each AS in our candidate set.

We observed that not all ASes registered in China will spoof answers, but of those infrastructure ASes that were still announcing prefixes at the time of our measurements[3], all issued spoofed responses. It is unclear exactly how many other ASes are rewriting in this way, but those that are not on *someone's* AS-path to a root server do not contribute to this threat[4]. However, the simple presence of *any* rewriting ASes in any path leading to a DNS root server creates a viable cross-modal vulnerability.

Our examination of these ASes showed that their spoofing is limited to only certain DNS names. While we were not able to identify an exhaustive list, using `jacksniff`, we were able to demonstrate this behavior for at least four common names: `facebook.com`, `youtube.com`, `twitter.com`, and `xxx.com`. In fact, the spoofing occurred for any DNS name when one of these names is a substring (such as `xxx.com.foo`).

From Figure 2 we can see that almost all of the DNS responses seen were spoofed responses. Occasionally, however, among the responses from a spoofing AS there were valid answers that refused to reply (rcode `REFUSED`) or issued a proper referral. This suggests that at these IP addresses there may have been an actual DNS resolver or name server.

A more detailed examination shows that, in fact, some prefixes do not spoof answers for *all* of their IPs. In

[3] Some of the ASes in our set do not appear to announce prefixes and are, therefore, difficult to query directly.

[4] As an anecdotal note, we have verified the existence of over 100 ASes that perform response spoofing.

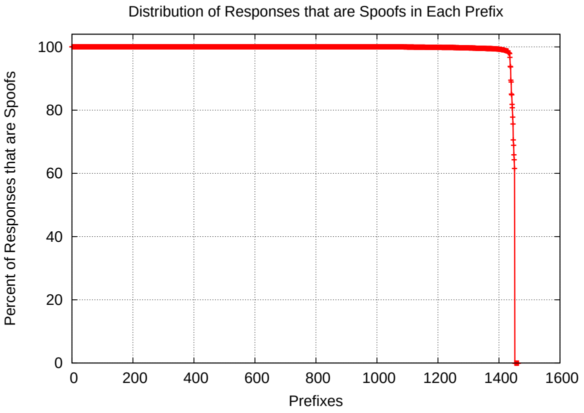

Figure 2: Distribution of percent of responses that were spoofed answers per prefix.

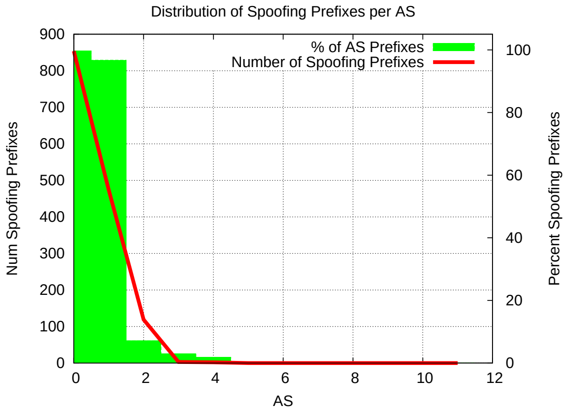

Figure 4: Fraction of prefixes that are announced by an AS that also spoof.

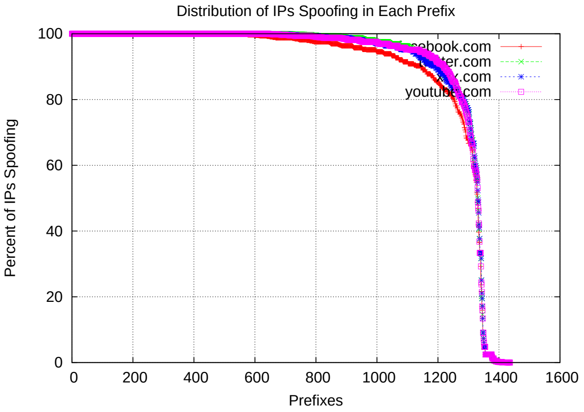

Figure 3: The distribution of how many IPs in each prefix spoof answers.

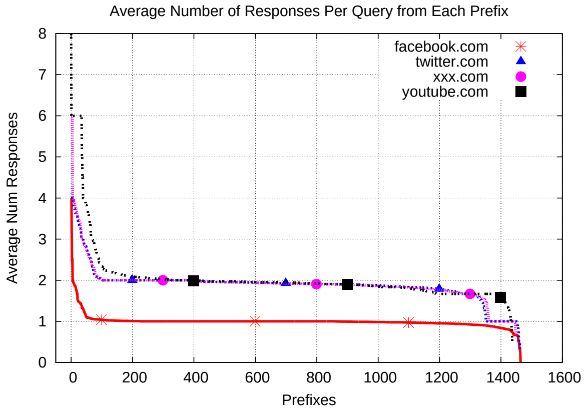

Figure 5: The number of spoofed responses can vary depending on which prefix is answering.

Figure 3 we can see the distribution of what fraction of responses seen from each prefix are actually spoofed answers (as opposed to timeouts or refusals, etc.). From this Figure we can see that of those prefixes that do spoof, roughly 90% of them spoof responses for at least 80% of their IPs. In addition, Figure 4 illustrates that of those ASes in our candidate set that do practice type 1 hijacking, there is a correlation between the number of prefixes they announce, and the percent of those prefixes that spoof responses. That is, not all prefixes announced by each AS spoof responses, but those announcing more prefixes seem to spoof from a greater percent of them.

When an AS spoofs responses, it tends to respond with more than one message. Figure 5 shows the average number of response returned from each prefix. We can see that one query can (in some cases) produce up to eight responses. Our speculation is that as the original query message traverses deeper into the spoofing authority, it elicits more spoofed responses. Therefore, we theorize that resolvers that fell victim to this attack likely did eventually receive valid referrals from the root server, but they were likely received *after* a spoofed response was already accepted.

Examination of Responses.

Our results show that 9 specific `A` record responses are present in more than an order of magnitude more responses than others (i.e. 9 IPs are returned much more often than all others). This is described in Figure 6 (which is in log-log scale, and shows a heavy tailed distribution), and Table 1 enumerates these `A` record values. The reasons for this are not immediately clear (as these IP addresses belong to widely different authorities), but one theory is that within the spoofing domain, these IP addresses are routed to local web servers,

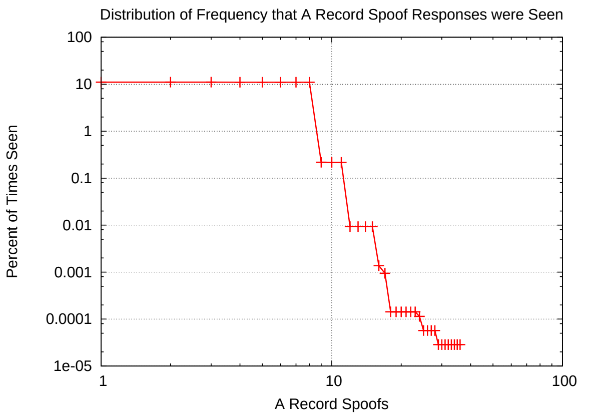

Figure 6: Distribution of how often specific A records were returned in spoofed responses.

Percent	A Record
11.0421%	46.82.174.68
11.0393%	203.98.7.65
11.0393%	159.106.121.75
11.0388%	8.7.198.45
11.0352%	243.185.187.39
11.0351%	78.16.49.15
11.0303%	37.61.54.158
11.0228%	59.24.3.173
11.021%	93.46.8.89

Table 1: The most common A record responses.

rather than the authentic destinations. The actual allocations of these IP address range from registrants in RIPE NCC, to APNIC, to ARIN.

Another odd pattern emerges in Figure 7, where the responses to some of the domain names had much different rates of checksum errors on the spoofed responses. `youtube.com`, for example, tended to get a lot more checksum errors than the others, and `facebook.com` responses got very few. The fact that `twitter.com` and `xxx.com` have similar behaviors here (and in Figure 5) suggest the possibility that there is some coordination on the part of the spoofing party.

5. CONCLUSION

In this work we have detailed the existence and nature of a dangerous type of hijacking vulnerability called cross-modal. This type of threat is not easy to detect and can elude current systems designed to detect hijacks in the Internet today, and we believe we are the first to formally identify it.

To justify the gravity of this threat, we have presented a detailed analysis of an actual instance of it in the Internet. In the case presented, certain ISPs are actively acting as Man-in-the-Middle (MitM) adversaries and successfully hijacking some clients. Our analysis suggest that this is possibly the result of a type 1 hijack that has leaked into the cross-modal threat that exists today.

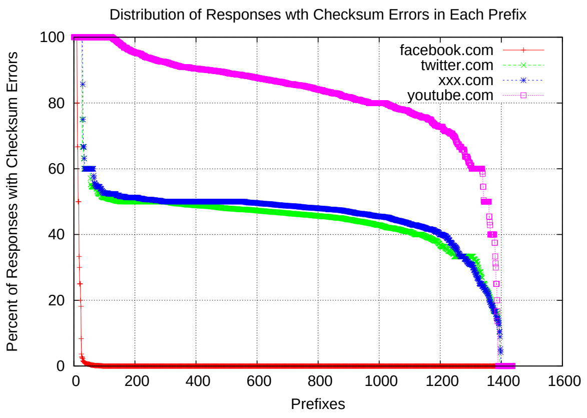

Figure 7: The rate of checksum errors on spoofed IP packets varies with the DNS name.

We feel that this work emphasizes the need for renewed investigation into identifying further examples of cross-modal threats. The evidence presented herein is not intended to fully detail the nature of *all* cross-modal threats that exist. Quite to the contrary, this work details just one instance and the difficulty current approaches have in detecting it and correcting for it.

In the future, we intend to pursue techniques that aid operators and systems in discovering these types of threats, determining their scopes, and identifying corrective behavior to overcome them.

6. REFERENCES

[1] More great firewall weirdness. `http://bokane.org/2010/02/26/more-great-firewall-weirdness/`.

[2] thepiratebay.org is reported hijacked on almost all dns servers. `http://groups.google.com/group/namebench/msg/0a458121bff88909?pli=1/`.

[3] Vantages. `http://secspider.cs.ucla.edu/vantages/`.

[4] alguien mas tiene bloqueado youtube? - pgina 5 - foros de chw. `http://www.chw.net/foro/off-topic-f16/310931-alguien-mas-tiene-bloqueado-youtube-p5.html`.

[5] Team Cymru. Ip to asn mapping. `http://www.team-cymru.org/Services/ip-to-asn.html`.

[6] Josh Karlin, Stephanie Forrest, and Jennifer Rexford. Pretty good bgp: Improving bgp by cautiously adopting routes. In *ICNP '06: 2006*

IEEE International Conference on Network Protocols. IEEE Computer Society, 2006.

[7] Mohit Lad, Dan Massey, Dan Pei, Yiguo Wu, Beichuan Zhang, and Lixia Zhang. Phas: A prefix hijack alert system. In *Fifteenth USENIX Security Symposium (USENIX Security 2006)*, August 2006.

[8] P. Mockapetris and K. J. Dunlap. Development of the domain name system. In *SIGCOMM '88*, pages 123–133, 1988.

[9] RIPE NCC. YouTube Hijacking: A RIPE NCC RIS case study. `http://www.ripe.net/news/study-youtube-hijacking.html`.

[10] University of Oregon. Route Views Project. http://www.routeviews.org.

[11] E. Osterweil, D. Massey, and L. Zhang. Deploying and Monitoring DNS Security (DNSSEC). In *2009 Annual Computer Security Applications Conference*, pages 429–438. IEEE, 2009.

[12] Y. Rekhter and T. Li. A border gateway protocol 4 (BGP-4). RFC 1771, IETF, March 1995.