Countermeasures for RAPTOR Attacks

Anne Edmundson
Princeton University
annee@cs.princeton.edu

Yixin Sun
Princeton University
yixins@cs.princeton.edu

ABSTRACT

Tor is a widely-used anonymity system, but has also been shown to be vulnerable to different types of traffic analysis attacks. Recently, experiments have shown that Tor is susceptible to asymmetric traffic analysis attacks, as well as other attacks at the AS level. These include malicious manipulations of traffic such as BGP prefix hijack and interception attacks. This paper presents a new evaluation of how resilient the Tor network is to prefix hijack attacks. We also present a set of novel countermeasures that will help prevent these types of attacks from being successful: a new path selection algorithm, a live monitoring framework that uses a variety of heuristics for attack detection, and suggestions for Tor relay operators.

1. PROBLEM MOTIVATION

Tor is a widely used system for anonymous communications. However, Tor is known to be vulnerable to traffic correlation attacks in which an adversary can correlate the traffic at both ends of the communication to deanonymize the users. Recently, researchers have shown that AS-level adversaries can exploit the dynamics of BGP routing and launch new attacks on Tor [31], including both passive attacks exploiting routing asymmetry and natural churn, as well as active attacks with BGP prefix hijacking/interception. Thus, these new attacks urge the need to build countermeasures to defend Tor against such malicious AS-level adversaries. In this work, we will focus on developing countermeasures against active BGP routing attacks.

There are two potential ways to counter the attacks: (1) Mitigating traffic interception, and (2) Mitigating correlation attacks. However, mitigating correlation attacks usually involve employing extra encryption schemes and/or packet obfuscation, which either requires extensive engineering efforts, or could result in high latency of Tor. Thus, given the constraints of mitigating correlation attacks, we will build countermeasures that focus on mitigating traffic interception, more specifically, against active BGP routing attacks.

Our approach includes two parts, as following.

The first part is a set of proactive approaches to countering BGP attacks. These approaches make it more difficult for an attacker to hijack Tor traffic. There's a surprisingly large number of Tor relays that are announced in prefixes that are larger than /24, which means they are vulnerable to sub-prefix hijack attacks. Therefore, one countermeasure is for operators to announce Tor relays in a /24. Similarly, operators can use a static route to guard relays, so that the traffic cannot be hijacked. In the case of equal-length prefix hijack attacks, we also propose a new path selection algorithm for the path from the client to the guard relay.

The second part is a reactive approach: a monitoring framework that can detect attacks in real-time. We have built a BGP monitoring framework for the Tor network that uses multiple different techniques to detect suspicious prefix announcements. This is used in conjunction with a traceroute monitoring framework for validation of path changes and suspicious paths.

Previous work has shown that the Tor network is susceptible to AS-level adversaries, and specifically prefix hijacks. In this work, we measure how resilient each AS – with at least one Tor relay – is to a prefix hijack from anywhere else on the Internet. Our contributions are:

- Measure resiliency of the ASes that contain relays and compare the resiliency of ASes that contain more relays to those that contain few relays.
- Measure impact of a BGP prefix hijack on the Tor network.
- Measure probability of any given Tor relay being decieved by a BGP prefix hijack.
- Build a real-time monitoring system for the Tor network, which uses both the control plane and data plane.
- Develop a new path selection technique for Tor clients, which we evaluate and show is more resistent to hijack attacks.
- Discuss experiences with relay operators in regards to: announcing relays in a /24 and using static routing.

The rest of the paper is as follows. Section 2 is a brief background discussion on Tor and RAPTOR attacks. Section 3

describes the metrics and methodology used to measure Tor relay resilience to prefix hijack attacks. Proactive countermeasures and reactive countermeasures are presented and discussed in Sections 4 and 5, respectively. Section 6 discusses related work and we present future work in Section 7. Lastly, we conclude in Section 8.

2. BACKGROUND

Here we discuss some background on the Tor network as well as RAPTOR attacks [31].

2.1 Tor

To communicate with a destination, Tor clients establish layered circuits through three subsequent Tor relays. The three relays are referred to as: entry (or guard) for the first one, middle for the second one, and exit relay for the last one. To load balance its traffic, Tor clients select relays with a probability that is proportional to their network capacity. Encryption is used to ensure that each relay learns the identity of only the previous hop and the next hop in the communications, and no single relay can link the client to the destination server.

It is well known that if an attacker can observe the traffic from the destination server to the exit relay as well as from the entry relay to the client (or traffic from the client to the entry relay and from the exit relay to the destination server), then it can leverage correlation between packet timing and sizes to infer the network identities of clients and servers (end-to-end timing analysis). This timing analysis works even if the communication is encrypted [31].

2.2 RAPTOR Attacks

RAPTOR attacks are a suite of attacks that can be launched by Autonomous Systems (ASes) to compromise user anonymity. There are three different types of attacks in this classification.

Asymmetric Traffic Analysis. AS-level adversaries can exploit the asymmetric nature of Internet routing to increase the chance of observing at least one direction of user traffic at both ends of the communication.

Natural Churn. AS-level adversaries can exploit natural churn in Internet routing to lie on the BGP paths for more users over time.

BGP Hijacks. Strategic AS-level adversaries can manipulate Internet routing via BGP hijacks (to discover the users using specific Tor guard nodes) and interceptions (to perform traffic analysis).

3. MEASURING TOR'S CURRENT STATE OF RESILIENCY TO HIJACK ATTACKS

Because the Tor network is susceptible to network-level adversaries, we aim to quantify how much of the Tor network would be affected by a BGP prefix hijack. First, we start by analyzing recent, publicized hijacks, and determine if they have affected the Tor network. Then, we look at how to evaluate the Tor network in terms to susceptibility to hijack attacks. The metrics used help enlighten the community about the state of the Tor network, in terms of how resilient the relays are to hijack attacks. Additionally, this helps quantify how vulnerable the Tor network is to network-level adversaries in a novel way. Specifically, we measure:

- Resiliency of the ASes that contain relays and compare the resiliency of ASes that contain more relays to those that contain few relays.
- Impact of a BGP prefix hijack on the Tor network.
- Probability of any given Tor relay being decieved by a BGP prefix hijack.

3.1 Recent Hijacks in the Wild

There have been a number of prefix hijack attacks in the past year. We analyzed the BGP routing announcements and withdrawals from the known time-frames to find the prefixes that were hijacked and compared them to the list of Tor relay IP addresses at the time. This shows us that prefix hijacks in the past year have affected Tor relays; it's also important to note that the analysis of publicized hijacks is not exhaustive. There may be hijack attacks that were not publicized or not recorded, and could have affected the Tor network as well. This analysis presents a lower-bound on the number of affected relays from two separate hijack attacks.

From routing announcement logs, we have seen two attacks that have involved Tor relays:

- On November 6th, 2015, AS9498 (BHARTI Airtel Ltd.) hijacked about 16,000 prefixes [3].
- On June 12th, 2015, AS4788 Telekom Malaysia started to announce about 179,000 of prefixes to Level3 (AS3549, the Global crossing AS) [4].

These are just incidents from the past year, but previous work has shown other, additional attacks that have affected the Tor network prior to 2015 [31].

3.2 Resilience Methodology

Previous work has tackled questions of AS resilience using simulations of the entire Internet [21]. We build off of this work by applying these metrics to the Tor network. First, we measure the resiliency of Tor-related ASes to equal-length prefix hijack attacks. To do so, we use an Internet topology [2] to get all of the AS relationships, and construct and AS-level graph. We identify the Tor-related ASes and simulate prefix hijacks on the graph. Specifically, our methodology consists of:

1. Construct an AS-level graph from an Internet topology.
2. Identify ASes that have at least one Tor relay.
3. Calculate the number of equally preferred paths from AS A to AS B, where AS B is a Tor-related AS.
4. Calculate the number of equally preferred paths from AS A to AS C, where AS C is the attacking (hijacking) AS.

5. Calculate resiliency using the equation in [21].

We follow this methodology for every AS A $\neq$ a Tor-related AS, every AS C $\neq$ a Tor-related AS, and for every AS B = Tor-related AS.

[21] explains the probability of a node v being deceived by a given false origin a announcing a route that belongs to true origin t:

$$\beta(a,t,v) = \frac{p(v,a)}{p(v,a)+p(v,t)}$$

where $p(v,a)$ is the number of equally preferred paths from node v to false origin a and $p(v,t)$ is the number of equally preferred paths from node v to true origin t. Using this probability, the same researchers introduced the resiliency metric – the resilience of a node t is the fraction of nodes that believe the true origin t given an arbitrary hijack against t:

$$R(t) = \sum_{a \in N} \sum_{v \in N} \frac{\beta(t,a,v)}{(N-1)(N-2)}$$

where N is the total number of ASes.

3.3 Resiliency Results and Values

We obtained the list of Tor guard/exit relays from Tor consensus in December 2015 and retrieved their belonging ASes. Then, we downloaded the AS topology published by CAIDA in December 2015. We evaluated the resiliency of each of the ASes which contain Tor guard/exit relays based on the AS topology using the method in [21].

The AS topology contains 52680 ASes, in which 612 ASes contain a total of 2548 Tor guard/exit relays. We simulated *all* possible hijacking scenarios against each of the 612 ASes, totaling $52679 \times 612 = 32,239,548$ prefix hijacks. As stated in [21], the resiliency of an AS t is calculated by:

$$R(t) = \sum_{a \in N} \sum_{v \in N} \frac{\bar{\beta}(t,a,v)}{(N-1)(N-2)} \quad (1)$$

in which N is the set of all ASes, and $\bar{\beta}(t,a,v)$ is defined as:

$$\bar{\beta}(t,a,v) = \frac{p(v,t)}{p(v,t)+p(v,a)} \quad (2)$$

in which $p(v,n)$ is the number of equally preferred paths from node v to node n given the routing policy and path lengths. Thus, we first calculate the resiliency of each Tor AS from each of the 52680 ASes as the *source* AS as following, and then sum up the resulting resiliency R to obtain the total resiliency for each of the Tor ASes.

function CalcResilience(graph G, node t)
  CalcPathsFromNode(G,t)
  $zeros(R)$
  for each reachable node v from node t **do**
    if node v contains Tor guard/exit relays **then**
      $n \leftarrow$ num. of less preferred nodes than node v
      $R[v] \leftarrow n + \bar{\beta}(v,a,t)$ $\forall$ equally preferred node a
    end if
  end for
  return R
end function

Note that, the CalcPathsFromNode(G,t) step above requires AS-level path predictions. Previous works have shown that AS level paths are determined mainly based on two preferences [30]:

- Local Preference
- Shortest Path

Further more, the AS paths should also have the *valley free* property. Thus, we use breadth first search to traverse the graph from a given source node based on this property and the preferences. We first explore provider-customer paths, which are the most preferred; next, we explore one peer-to-peer path followed by a sequence of provider-customer paths, which are the next preferred; finally, we explore customer-provider paths followed by an optional peer-to-peer path and then followed by a sequence of provider-customer paths. Note that, nodes are explored in the most preferred to least preferred order, and those which are explored in the same step are equally preferred. This ordering will help the above resiliency calculation.

After calculating the resilience for each AS that contains at least one Tor relay, we can see that most ASes lie in the middle of the spectrum for resilience (as shown in Figure 1).

We then looked at which ASes had the most relays, and what their corresponding resilience is; ideally, ASes with more relays are also more resilient to hijack attacks. Figure 2 shows the resilience value for an AS and the AS's corresponding number of relays.

We can see one outlier - AS16276 - which contains 339 relays; unfortunately, this AS also has a relatively low resilience against hijack attacks. This motivates our proactive and reactive countermeasures against hijack attacks.

4. PROACTIVE APPROACHES

We take three different proactive approaches to counter RAPTOR attacks: 1) convincing relay operators to announce the relay in a /24 prefix, 2) analyzing the feasibility of having a static route to guard relays, and 3) introducing a new path selection algorithm that minimizes the likelihood that a client sees a hijacked route in the case that her guard is hijacked.

4.1 Using /24 Prefixes

Sun *et al.* [31] recently found that >90% of BGP prefixes hosting relays are shorter than /24, making them vulnerable to a more-specific BGP prefix attack. Thus, one quick way to make Tor relays more resilient to such active routing

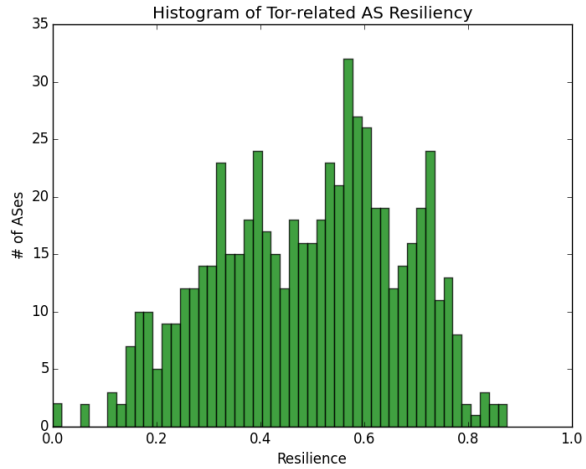

Figure 1: A histogram of resilience values for all ASes that contain a Tor relay.

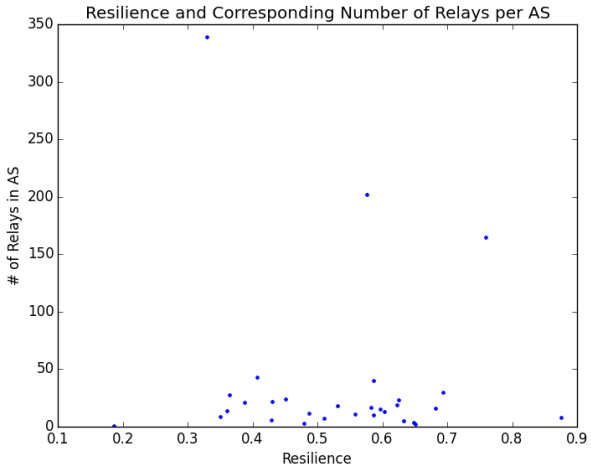

Figure 2: Resilience values for each AS and each AS's corresponding number of relays.

attacks is to announce /24 prefix covering Tor relays. In order to make real world impact of this approach, we plan to start a campaign by contacting network operators whose prefixes contain Tor relays, and asking them to announce a more specific /24 prefix covering the relay.

4.2 Static Routing and Path Protection Mechanisms

There has been progress in protecting certain BGP paths by using static routes, or other protection mechanisms. We plan to explore how difficult it is to create static routes to guard relays in order to help prevent prefix hijacks that aim to hijack a guard relay.

In the past, there has been work on protecting routes to top-level DNS servers. This was done by a fairly aggressive set of filters to be a little more more conservative in accepting alternate routes to the DNS servers [5].

4.3 Guard Relay Selection

Guard relay is at an important position that it has direct connection with the Tor client. Thus, securing the guard relay would be our first step. It has been shown that AS-level adversaries can launch a more-specific prefix attack to intercept the Tor traffic from the guard relay to the malicious AS [31], and this can be potentially prevented by advertising /24 prefixes. However, even if the guard relay belongs to a /24 prefix, it is still subject to an equally-specific prefix attack. Unlike more-specific attacks which spread through the whole internet, equally-specific attacks can only affect connections within a small range - i.e., ASes that are within a certain number of hops away, depending on the influence of the announcing AS. Thus, picking a guard relay that is relatively close to the client AS could make it more resilient to such equally-specific attack.

On the other hand, we also don't want to pick a guard relay that is too close - in an extreme case, picking a guard within the same AS as the client will reveal client location to the adversary.

Therefore, we propose a new path selection algorithm that incorporates this aspect, picking a guard relay that satisfies the optimal balance between resilience to routing attacks and privacy protection.

Recall from Section 2 that we sum up the resiliency from *each* source AS to obtain the total resiliency for the Tor-related ASes. However, from the perspective of a Tor client, the client would only care about the resiliency of a Tor-related AS from where the client is located as the source AS instead of the total resiliency. Thus, we propose guard relay selection algorithm as following:

1. Upon initiating the Tor connection and before selecting a guard relay, the Tor client will download the latest AS topology (<700KB if compressed) along with the Tor consensus data.
2. The Tor client will run the resiliency calculation from the source AS where he is located to all ASes which contain Tor guard relays. By running the resiliency calculation locally at the client, it ensures that the location of the client will not be revealed to any outside servers.
3. Tor relay selection has been bandwidth-aware and prefers high bandwidth relays. We will still take into account the bandwidth, and offer the users a tunable parameter α in the selection. Each relay i will be assigned a weight as:
$$W(i) = \alpha \times R(i) + (1 - \alpha) \times B(i)$$
in which $R(i)$ is the resiliency calculated from the previous step and $B(i)$ is the relay bandwidth obtained from Tor consensus. When α is set to 0, the relay selection becomes the same as bandwidth-only selection.
4. If we simply select the set of guard relays based on the probability of $relay\_weight/total\_weight$, an adversary can potentially run a relay that is close enough to the Tor client, such that it has high resiliency from the client AS as the source, and thus obtains high probability of being chosen. Therefore, we will first select a cluster of $m + \alpha \cdot g \cdot (N - m)$ relays, in which m is the minimum number of relays needed (i.e., 3), N is the total number of relays and g is a configurable parameter indicating the maximum percentage of additional relays we want to pick into the cluster. Then, we will pick the set of guard relays at random from the cluster. Note that when g is set to 0, then no randomization will be performed, while if g is set to 1, all guard relays will be picked randomly regardless of bandwidth or resiliency. We will evaluate how different values of g may impact the performance.
5. Finally, after selecting the set of entry guard relays, the remaining part of the circuit construction process stays the same as it is in Tor.

4.4 Relay Selection Evaluation

We evaluate the AS resilience based relay selection from performance and security perspectives.

Performance Evaluation

First, we evaluate the runtime of the AS resilience calculation given a source AS. We pick 1000 ASes randomly as the source AS, and record how much time it takes each of them to complete the calculation. Figure 3 show the CDF of the runtime of AS resilience calculation. Most of the source ASes finish within 0.6 second.

Security Evaluation

Next, we evaluate the security aspect of the relay selection. We first evaluate the relay selection variance in entropy without doing the clustering in step 4. We use the Gini coefficient as the entropy evaluation metric, as it has been used in previous work to measure anonymity selection in Tor [6]. We evaluate five values of $\alpha : \{0, 0.25, 0.5, 0.75, 1\}$. Note that when $\alpha = 0$, it is equivalent to bandwidth-based selection. Figure 4 shows the result. The green line to the right is when $\alpha = 0$, so it's solely based on bandwidth resulting in a Gini coefficient of 0.607 for all source ASes. It is interesting that the Gini coefficients for the other four α values which involve resilience-based selection are very similar, all with much lower Gini coefficients (higher entropy and lower

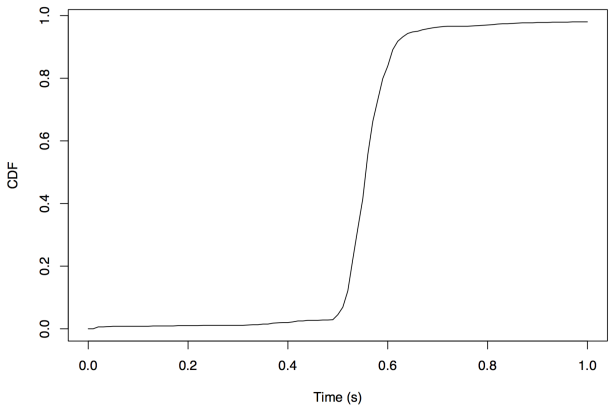

Figure 3: Runtime of AS resilience calculation from a given source AS

skew in relay selection probability) than bandwidth-based selection.

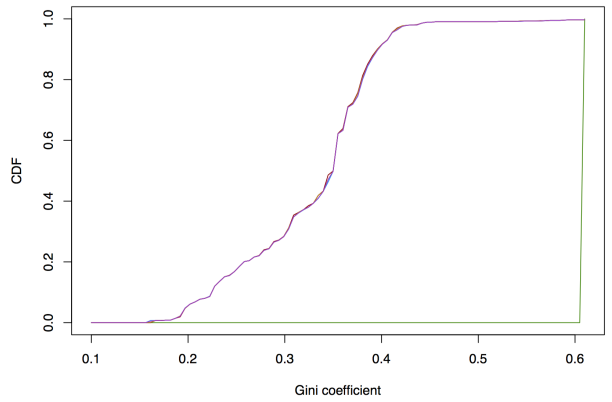

Figure 4: Gini coefficients with different α values

5. REACTIVE APPROACHES

In addition to proactive approaches to helping prevent RAPTOR attacks, we have also taken a reactive approach. In the case that an attack is happening, we can detect it using a live monitoring framework.

5.1 A Live Monitoring Framework for Tor

A possible countermeasure against routing attacks on the Tor network is attack detection and user notification. Routing authorities then know which relays are being hijacked and/or intercepted, and can make routing decisious accordingly. We observed that routing attacks are almost always short-lived, which allows routing authorities to suspend use of hijacked/intercepted relays until enough time has passed to use them again.

The live monitoring framework aims at detecting suspicious routing attacks that affect Tor relays, and then react correspondingly to alert Tor clients of the scenario. The monitoring framework consists of two parts: BGP monitoring and Traceroute monitoring.

Relay Info from Tor Consensus Tor consensus releases up-to-date information for current running relays every hour. Our system automatically grabs this consensus data once it's updated. We focus on guard relays and exit relays, which reside at the two ends of the communication path and can easily be the target of an adversary. Further more, since we focus on AS-level adversaries, so it is unnecessary to monitor each individual relay by its IP address. But instead, we monitor the /24 prefixes which contain Tor guard and exit relays. Note that, there is no need to monitor a more specific prefix than /24, since generally /24 is the longest prefix accepted in BGP announcement.
Thus, we construct a live monitoring database which is being updated every hour with latest Tor relay data. The table contains the following fields:

/24 Prefix	Total Bandwidth	Number of Guards	Number of Exits	Timestamp

Each /24 prefix that contains any guard/exit relays will have one entry in the table, and the list of /24 prefixes will be used for our BGP and Traceroute monitoring frameworks, which we describe in the following.

BGP Monitoring Framework monitors the control plane of internet routing. We collect live BGP announcements data from BGP Stream, in combination with the latest Tor relay data. We monitor all the Tor-related /24 prefixes we obtain, as described in the table above. We check if any activity related with the prefixes exhibit any anomaly. These anomalies can be detected by developing certain heuristics, such as the amount of time that a BGP path is used or the frequency that a path us announced; if certain anomalies fall under a threshold for a given heuristic, they should be flagged as potential attacks. This analysis will require saving offline relay bgp info for some period of time. This framework helps:

- Identify suspicious prefix announcements.
- Differentiate between potential attacks and "normal" behavior, such as multiple origin AS conflicts, backup paths, etc [40].

We use Team Cymru to obtain AS ownership of prefixes. Some prefixes are owned by an organization with multiple AS numbers, so we take this aspect into consideration and store the AS origins of these prefixes. If we observe any change in AS paths, we will first check if the prefix has multiple AS origins, and if so, as long as the new on-path AS also owns the prefix, then it would not be seen as an attack.

The implementation of the BGP Monitoring framework is based on BGP Stream [1]. We analyze the live stream of BGP updates and withdrawals, focusing just on the prefixes that contain a Tor relay. We monitor the prefixes that are reported through Team Cymru, as well as the /24 that contains each relay; we do this because we would like to be able to detect sub-prefix hijack attacks, so we must monitor longer prefixes in addition to the reported prefix.

In addition to comparing the AS that owns a prefix (according to Team Cymru) with the AS that announces the prefix, our analysis involves three different detection techniques:

1. Origin AS check. We collect the origin AS in the live BGP update and comparing it to the owner AS reported by Team Cymru. If these don't match up, then we flag the update (and prefix) as a potential hijack, otherwise we ignore it.

2. Frequency heuristic. Routing attacks can be characterized by an AS announcing a path once (or extremely rarely) to a prefix that it does not own. The frequency heuristic detects attacks that exhibit this behavior. It measures the frequency of each AS that originates a given prefix; if the frequency is lower than a specified threshold, then it could be a potential hijack attack.

3. Time heuristic. Most known attacks last a relatively short amount of time. The time heuristic measures the amount of time each path to a prefix is announced for; if the amount of time is extremely small (below a specified threshold), then there is the possibility of it being a routing attack.

Traceroute Monitoring Framework monitors the data plane of internet routing, i.e., how packets travel through the Internet in reality. The traceroute monitoring framework is used as a verification mechanism if the BGP monitoring framework flags certain behavior. There may be many false positives in detecting hijack/interception attacks due to the nature of BGP. With many false positives, the traceroute monitoring framework will be used often for verification - this raises a question of optimization, which we will also address.

Our Traceroute monitoring framework retrieves updated Tor relay data hourly from the database described above. Since running large number of continuous traceroutes to relay IP addresses may create unnecessary extra traffic to the Tor network, so we selectively monitor a subset of prefixes at certain frequency rates when there is no anomaly from BGP monitoring data, and when there is suspicious activity report by BGP monitoring, the traceroute monitoring can be "triggered" to target the suspicious prefix announcement to verify the anomaly.

- Selectively monitor relays of interest.
  A /24 prefix can be evaluated based on several factors: (1) total combined bandwidth of Tor relays it covers, denoted as b_i for prefix i; (2) total number of guard relays it covers, denoted as g_i; (3) total number of exit relays it covers, denoted as e_i; and (4) resilience of the prefix to BGP hijack/interception attacks, denoted as r_i. These factors can make the prefix an attractive target to adversaries. Using these factors, we want to formulate the overall security of the system as following, and the goal is to find the monitoring frequency f_i for each relay i that maximizes the overall security of the network.

$$\max \sum_{i=1}^{N} \frac{\log(f_i + 1)}{b_i + g_i + e_i + r_i} \tag{3}$$

$$\text{s.t.} \sum_{i=1}^{N} f_i \leq F \tag{4}$$

$$0 < f_i \leq M, \forall i \tag{5}$$

  N is the total number of prefixes we want to monitor, F is the constraint on total number of traceroutes we can send from each Planetlab node per day, and M is the constraint on total number of traceroutes needed for each prefix. Given the solution, we use a collection of Planetlab nodes located in different ASes to send traceroutes to the prefix at its frequency rate.

- Monitoring target prefixes triggered by BGP.
  If we detect any anomaly from the BGP monitoring framework, we will immediately send traceroutes to the suspicious prefixes to verify whether there is truly a path change happening on the data plane to the Tor relays.

- Detecting anomaly from Traceroutes
  Even when there is no suspicious activity reported by BGP monitoring, it is also possible we detect anomaly from our selective traceroute monitoring. We keep track of the past traceroute monitoring results in a database table, as following:

Source Prefix	Dest Prefix	AS Path	Time Created	Time Last Updated	Current

  With this table, we will be able to compare the current AS path with past AS paths to detect any path changes, which may indicate a hijack event happening.

5.2 Framework Evaluation

The live monitoring framework will be evaluated on a number of characteristics, including false positive rate, false negative rate, as well as performance and overhead.

5.3 Deployment Experience

The BGP monitoring framework has been running for a week. It has recorded over 3 million announcements (not specific to Tor), 330 announcements that include a Tor relay, and no announcements that include a Tor relay and have an origin AS that disagrees with Team Cymru's data.

After implementing the three heuristics described above (frequency, time, session), we ran them on the data collected during the week of live monitoring. For our analysis, we assume that there were no hijack attacks on the Tor network during the week we collected data. Our frequency heuristic was tuned to a threshold of .01%, meaning that any prefix announced by an AS with a frequency lower than .01% of all announcements would be flagged as suspicious. This resulted in approximately 40 suspicious AS - prefix pairs. Our time heuristics was set to the same threshold, and resulted in approximately 60 suspicious pairs. This indicates that our heuristics need to be tuned for more precise and accurate results.

6. RELATED WORK

BGP Attacks and Security. BGP attacks are well-studied, particularly prefix hijack and interception attacks [8, 24, 34]. Arnbak, et al. showed that prefix interceptions could be used by nation-states as a way to conduct surveillance on their citizens [7]. It's also known that routing anomalies can lead to network snapshots that look similar to attack scenarios. These are due to a range of routing policies, misconfigurations, and multiple origin AS conflicts [10, 23, 40].

The research community has contributed a number of protocols to help secure interdomain routing [9, 11, 14, 16, 35, 33]. Unfortunately, it has also been shown that partial deployment of secure interdomain routing protocols does not provide much security [22].

There is also a large body of research with the goal of defending against and detecting prefix hijacks and interceptions. These include defensive and detection tools [20, 15, 28, 37, 41, 29, 36], as well as mechanisms such as PGBGP, which allow network administrators more time to determine if an attack is happening before using new routes [19]. There has also been research not only on detecting attacks, but on determining the location of the attacker [27]. Qui, et al. detected any bogus routes, not just hijacks or interceptions [26]. In addition to detection algorithms, there has been research in visualization of real-time detection algorithms [32]. Our work does not aim to contribute a new hijack detection tool, but rather compliments existing tools by applying a monitoring framework to the Tor network.

BGP Attack Resiliency. Prior research on prefix hijack attack resilience has been simulated on the Internet for equal-length prefix hijacks [21]. They find that customers of Tier-1 ASes are the most resilient and also create the most impact (if they were to hijack a prefix). There has been some related work in relating hijack attacks to the Internet hierarchy [39, 38]. This differs from our work; we focus on the resilience of ASes that contain Tor relays, as well as measure the resilience of guard relays and exit relays as groups.

Network Adversaries on Tor. Network-level adversaries are known in anonymity networks. Feamster and Dingledine [13] first investigated AS-level path in anonymity networks, which showed that some AS could appear on nearly 30% of entry-exit pairs. Murdoch and Zielinski [25] later demonstrated the threat posed by network-level adversaries who can deanonymize users by performing traffic analysis. Furthermore, Edman and Syverson [12] demonstrated that even given the explosive growth of Tor during the past years, still about 18% of Tor circuits result in a single AS being able to observe both ends. In 2013, Johnson *et al.* [17] evaluated the security of Tor users over a period of time, and the result indicated that a network-level adversary with just low bandwidth cost can deanonymize any users within three months with over 50% probability and within six months with over 80% probability.

AS-level Tor Path Selection. The existence of network-level adversaires urges the need to incorporate AS-awareness path selection in Tor. In 2012, Akhoondi *et al.* [6] proposed LASTor, a Tor client which takes into account AS-level path and relay locations in path selection, although LASTor neglected relay capacity and its AS resilience to active attacks. Recently, Nithyanand *et al.* [30] constructed a new Tor client, Astoria, which adopted a new path selection algorithm which considered more aspects - relay capacity, asymmetric routing, colluding ASes, etc.. However, Astoria only considers a passive AS-level attacker, while does not evaluate the AS resilience to an active routing attack.

Towards this goal, it is important to understand AS-level internet topology and network path predictions. Lad *et al.* [21] investigated the relation between internet topology and prefix hijacking, and provided a metric for evaluating AS resilience to active prefix hijack attacks. Although, the study was conducted in 2007 when there were far less ASes than now. Recently, Juen *et al.* [18] performed a measurement study using Tracecroutes on network-level paths that Tor traffic actually get routed through.

7. FUTURE WORK

This is an important research area, and there is still much more to do. There are three main areas where we wish to further this work: quantification of resiliency, monitoring framework, and qualitative suggestions for relay operators.

Resiliency.

There are still a number of characteristics of the Tor network that would be useful and helpful to quantify. These include:

- Quantify how resilient Tor ASes are to interception attacks.
- Quantify if Tor relays have become more resilient since the initial network was built.
- Quantify how fast relay resilience changes.

These metrics will better allow us to see how vulnerable the Tor network is to prefix hijack and interception attacks. In addition to measuring the current state of resilience, as was shown in Section 3, we will measure how the resilience of Tor relays has changed over the years. We answer the following questions:

- Have Tor relays become more resilient since the initial network was built?
- How fast does relay resilience change?

We plan to answer the first question by calculating the given resilience metrics for each past year - similar to a longitudinal study of Tor relay resiliency. We plan to answer the second question by calculating the given resilience metrics each week for the next couple of months. The results from answering the first question will also help us answer the second question.

Next we measure the resiliency of Tor-related ASes to prefix interception attacks. We modify the methodology from above for this measurement in the following way:

1. Construct an AS-level graph from an Internet topology.

2. Identify ASes that have at least one Tor relay.

3. Calculate the number of equally preferred paths from AS A to AS B, where AS A $\neq$ AS B, AS A and AS B are not Tor-related ASes, and there must be a Tor relay on the path from AS A to AS B.

4. Calculate the number of equally preferred paths from AS A to AS B, where AS A $\neq$ AS B, AS A and AS B are not Tor-related ASes, there must an AS C (intercepting AS) on the path from AS A to AS B, and no Tor-related AS on the path from AS A to AS B.

5. Calculate resiliency using the equation described above.

Similarly, we measure how this resilience to interception attacks has changed over time.

Monitoring Framework.

As of now, our live monitoring framework consists of two parts: data plane and control plane. One of the most important next steps is to connect both parts and run the system as one large framework. The BGP monitoring framework could run consistently, and when any suspicious announcements are flagged (either by the heuristics or the AS comparison), then this could trigger the traceroute monitoring framework.

Other future work includes developing methods to detect interception attacks - this has been shown to be difficult and accurate detection is still an open problem [8]. Furthermore, making the already implemented heuristics and techniques more accurate and precise is still to be done.

Lastly, the monitoring framework needs methods for evaluation. Metrics such as false positive rate, true positive rate, time and performance, will all be beneficial to showing the importance of the framework, as well as for improving the framework.

Qualitative Suggestions.

Additional work includes setting up our own set of relays at Princeton University and working with the necessary operators to announce the relays in their own /24 network and think of the possibility of static routing to the guard relay.

8. CONCLUSION

In this work, we have presented countermeasures to a set of RAPTOR attacks - attacks that involve AS-level adversaries. In particular, our proposed countermeasures target routing manipulations such as BGP prefix hijack and interception attacks.

First, we evaluated the Tor network for it's current state of resilience to hijack attacks. We saw some ASes have a much higher resilience than others, and we compared this to the number of relays that each AS contains.

Then we presented proactive and reactive countermeasures. The proactive countermeasures included having operators announce relays in a /24, using a static route to guard relays, and introducing a new path selection algorithm for a client to a guard relay. The reactive countermeasures included a live monitoring framework with both a data plane and control plane component, as well as a set of heuristics that help identify suspicious routing changes.

9. REFERENCES

[1] Bgp stream. `http://bgpstream.caida.org/`.

[2] The ipv4 routed /24 topology dataset. `http://www.caida.org/data/active/ipv4_routed_24_topology_dataset.xml`.

[3] Large scale bgp hijack out of india. `http://www.bgpmon.net/large-scale-bgp-hijack-out-of-india/`.

[4] Massive route leak causes internet slowdown. `http://www.bgpmon.net/massive-route-leak-cause-internet-slowdown/`.

[5] Protecting bgp routes to top-level dns servers. `http://web.cs.ucla.edu/~lixia/papers/03TPDS.pdf`.

[6] Akhoondi, M., Yu, C., and Madhyastha, H. V. Lastor: A low-latency as-aware tor client. In *Security and Privacy (SP), 2012 IEEE Symposium on* (2012), IEEE, pp. 476–490.

[7] Arnbak, A., and Goldberg, S. Loopholes for circumventing the constitution: Warrantless bulk surveillance on americans by collecting network traffic abroad, 2014.

[8] Ballani, H., Francis, P., and Zhang, X. A study of prefix hijacking and interception in the internet. In *ACM SIGCOMM Computer Communication Review* (2007), vol. 37, ACM, pp. 265–276.

[9] Boldyreva, A., and Lychev, R. Provable security of s-bgp and other path vector protocols: model, analysis and extensions. In *Proceedings of the 2012 ACM conference on Computer and communications security* (2012), ACM, pp. 541–552.

[10] Caesar, M., and Rexford, J. Bgp routing policies in isp networks. *Network, IEEE 19*, 6 (2005), 5–11.

[11] Chan, H., Dash, D., Perrig, A., and Zhang, H. *Modeling adoptability of secure BGP protocol*, vol. 36. ACM, 2006.

[12] Edman, M., and Syverson, P. As-awareness in tor path selection. In *Proceedings of the 16th ACM conference on Computer and communications security* (2009), ACM, pp. 380–389.

[13] Feamster, N., and Dingledine, R. Location diversity in anonymity networks. In *Proceedings of the 2004 ACM workshop on Privacy in the electronic society* (2004), ACM, pp. 66–76.

[14] Gill, P., Schapira, M., and Goldberg, S. Let the market drive deployment: A strategy for transitioning to bgp security. In *ACM SIGCOMM Computer Communication Review* (2011), vol. 41, ACM, pp. 14–25.

[15] Hu, X., and Mao, Z. M. Accurate real-time identification of ip prefix hijacking. In *Security and Privacy, 2007. SP'07. IEEE Symposium on* (2007), IEEE, pp. 3–17.

[16] Hu, Y.-C., Perrig, A., and Sirbu, M. Spv: Secure path vector routing for securing bgp. In *ACM SIGCOMM Computer Communication Review* (2004), vol. 34, ACM, pp. 179–192.

[17] Johnson, A., Wacek, C., Jansen, R., Sherr, M., and Syverson, P. Users get routed: Traffic correlation on tor by realistic adversaries. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security* (2013), ACM,

pp. 337–348.

[18] Juen, J., Das, A., Johnson, A., Borisov, N., and Caesar, M. Defending tor from network adversaries: A case study of network path prediction. *arXiv preprint arXiv:1410.1823* (2014).

[19] Karlin, J., Forrest, S., and Rexford, J. Pretty good bgp: Improving bgp by cautiously adopting routes. In *Network Protocols, 2006. ICNP'06. Proceedings of the 2006 14th IEEE International Conference on* (2006), IEEE, pp. 290–299.

[20] Lad, M., Massey, D., Pei, D., Wu, Y., Zhang, B., and Zhang, L. Phas: A prefix hijack alert system. In *Usenix Security* (2006).

[21] Lad, M., Oliveira, R., Zhang, B., and Zhang, L. Understanding resiliency of internet topology against prefix hijack attacks. In *Dependable Systems and Networks, 2007. DSN'07. 37th Annual IEEE/IFIP International Conference on* (2007), IEEE, pp. 368–377.

[22] Lychev, R., Goldberg, S., and Schapira, M. Bgp security in partial deployment: is the juice worth the squeeze? *ACM SIGCOMM Computer Communication Review 43*, 4 (2013), 171–182.

[23] Mahajan, R., Wetherall, D., and Anderson, T. Understanding bgp misconfiguration. In *ACM SIGCOMM Computer Communication Review* (2002), vol. 32, ACM, pp. 3–16.

[24] McArthur, C., and Guirguis, M. Stealthy ip prefix hijacking: don't bite off more than you can chew. In *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE* (2009), IEEE, pp. 1–6.

[25] Murdoch, S. J., and Zieliński, P. Sampled traffic analysis by internet-exchange-level adversaries. In *Privacy Enhancing Technologies* (2007), Springer, pp. 167–183.

[26] Qiu, J., Gao, L., Ranjan, S., and Nucci, A. Detecting bogus bgp route information: Going beyond prefix hijacking. In *Security and Privacy in Communications Networks and the Workshops, 2007. SecureComm 2007. Third International Conference on* (2007), IEEE, pp. 381–390.

[27] Qiu, T., Ji, L., Pei, D., Wang, J., Xu, J. J., and Ballani, H. Locating prefix hijackers using lock. In *USENIX Security Symposium* (2009), pp. 135–150.

[28] Shi, X., Xiang, Y., Wang, Z., Yin, X., and Wu, J. Detecting prefix hijackings in the internet with argus. In *Proceedings of the 2012 ACM conference on Internet measurement conference* (2012), ACM, pp. 15–28.

[29] Sriram, K., Borchert, O., Kim, O., Gleichmann, P., and Montgomery, D. A comparative analysis of bgp anomaly detection and robustness algorithms. In *Conference For Homeland Security, 2009. CATCH'09. Cybersecurity Applications & Technology* (2009), IEEE, pp. 25–38.

[30] Starov, O., Nithyanand, R., Zair, A., Gill, P., and Schapira, M. Measuring and mitigating as-level adversaries against tor. *arXiv preprint arXiv:1505.05173* (2015).

[31] Sun, Y., Edmundson, A., Vanbever, L., Li, O., Rexford, J., Chiang, M., and Mittal, P. Raptor: routing attacks on privacy in tor. *arXiv preprint arXiv:1503.03940* (2015).

[32] Teoh, S. T., Ranjan, S., Nucci, A., and Chuah, C.-N. Bgp eye: a new visualization tool for real-time detection and analysis of bgp anomalies. In *Proceedings of the 3rd international workshop on Visualization for computer security* (2006), ACM, pp. 81–90.

[33] van Oorschot, P. C., Wan, T., and Kranakis, E. On interdomain routing security and pretty secure bgp (psbgp). *ACM Transactions on Information and System Security (TISSEC) 10*, 3 (2007), 11.

[34] Zhang, Y., and Pourzandi, M. Studying impacts of prefix interception attack by exploring bgp as-path prepending. In *Distributed Computing Systems (ICDCS), 2012 IEEE 32nd International Conference on* (2012), IEEE, pp. 667–677.

[35] Zhang, Y., Zhang, Z., Mao, Z. M., and Hu, Y. C. Hc-bgp: A light-weight and flexible scheme for securing prefix ownership. In *Dependable Systems & Networks, 2009. DSN'09. IEEE/IFIP International Conference on* (2009), IEEE, pp. 23–32.

[36] Zhang, Z., Zhang, Y., Hu, Y. C., and Mao, Z. M. Practical defenses against bgp prefix hijacking. In *Proceedings of the 2007 ACM CoNEXT conference* (2007), ACM, p. 3.

[37] Zhang, Z., Zhang, Y., Hu, Y. C., Mao, Z. M., and Bush, R. Ispy: detecting ip prefix hijacking on my own. In *ACM SIGCOMM Computer Communication Review* (2008), vol. 38, ACM, pp. 327–338.

[38] Zhao, J., and Wen, Y. Analysis on the effect of prefix hijacking attack and internet hierarchy. In *Computer and Information Technology (CIT), 2012 IEEE 12th International Conference on* (2012), IEEE, pp. 375–382.

[39] Zhao, J., Wen, Y., Li, X., Peng, W., and Zhao, F. The relation on prefix hijacking and the internet hierarchy. In *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012 Sixth International Conference on* (2012), IEEE, pp. 415–420.

[40] Zhao, X., Pei, D., Wang, L., Massey, D., Mankin, A., Wu, S. F., and Zhang, L. An analysis of bgp multiple origin as (moas) conflicts. In *Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement* (2001), ACM, pp. 31–35.

[41] Zheng, C., Ji, L., Pei, D., Wang, J., and Francis, P. A light-weight distributed scheme for detecting ip prefix hijacks in real-time. In *ACM SIGCOMM Computer Communication Review* (2007), vol. 37, ACM, pp. 277–288.