Improving the Security and Robustness of Internet Routing: What Can We Do Today?

Georgos Siganos, Michalis Faloutsos

***Abstract*— Attacks at the control and routing plane may be the next generation of threats for the Internet. Manipulation of the routing layer could originate from profiteering, malice, or simply human error. The community has recognized this danger and several promising approaches have been proposed. Most of these approaches attempt to capture and block routing anomalies. In practice, the difficulty of deploying such approaches limits their usefulness. Our goal is to develop a scheme that can have immediate impact today. In this light, we propose a reactive approach that can help reduce the extent and impact of routing errors. More specifically, we develop an approach and a tool to identify routing errors in BGP routing by using the policies that Autonomous Systems register in the Internet Routing Registries. We use the policy of an AS as found in these registries to detect deviations between the intended policy and the actual policy seen in BGP. As a proof of concept, we use the RIPE registry to monitor the European Internet routing for ten days. We found that for 97% of the prefixes we can validate their origin AS using the RIPE registry. Additionally, we find that for 60% of the networks, their policy is fresh and can be used to check their routing for abnormal routing behavior.**

I. INTRODUCTION

In this work, we propose a reactive approach and present a tool to identify BGP routing errors in the Internet in order to reduce their extent and impact. The Internet has revolutionized the way people work and communicate to the extent that, in some countries, it is considered to be just another utility like electricity and water. Therefore, it is important to ensure that the Internet continues to function reliably, even in the face of attacks, exploits, and errors. A fundamental component of the Internet functionality is Internet routing and therefore, it is critical to ensure its correctness and reliability. In this paper, we investigate what is the best we can do today to improve the security of Internet routing, and propose mechanisms to reduce the extent and impact of such errors. We use the term *routing security* [1] to denote the loose concepts of correctness in BGP routing according to the intended policy as defined by the network operators.

BGP [2] has evolved in an incremental way [3] [4] [5] [6] in order to address the security requirements that threatened its robust operation, and has overcome a number of problems since its original deployment. One of the problems in BGP is the unauthorized advertisement of IP prefixes. For example, in 1997, AS7007 [7] deaggregated and advertised a large portion of the Internet, thus creating a black-hole for Internet traffic. Another abnormal routing behavior can happen with illegal traffic engineering [8]. These problems can happen either because of compromised routers, or by human error. It has been documented that BGP is especially vulnerable to human errors [9]. Configuring the routers is a difficult and tedious procedure. The tools used are usually low-level with no static checking of the correctness of the configuration and no immediate feedback control on possible errors. It is difficult to predict what will happen with a configuration change [10]. As a result, it is often done using a trial and error approach.

The incremental improvements have allowed BGP to evolve and become a very complex network. But with the significance of the network ever increasing, there is a need for more security [11] [12] [13]. A number of approaches have been proposed and IETF has established a working group, RPsec [1], to address the threats and possible solutions to secure Internet routing. The most well-known and advanced proposal is S-BGP [14], [15], which is proposed by BBN and has been in development for many years. They use Public Key Infrastructure (PKI) to authenticate every aspect of a routing message. SoBGP [16] is a new proposal by engineers that work for CISCO, a company with huge influence on the Internet. Its original goal was to allow only the authorized networks to advertise their address space. Currently, they are extending it to cover various other scenarios and threats. Other more lightweight proposals include IRV [17], SPV [18], whisper [19], and moas [20].

Securing Internet routing is a daunting task. We need a flexible and scalable protocol and most importantly, a deployment strategy, since the Internet consists today of hundreds of thousands of routers and tens of thousands

---

G. Siganos and M. Faloutsos are with the Dept. of Computer Science & Engineering,University of California, Riverside, Email:{siganos,michalis}@cs.ucr.edu

of independent networks. The current proposals have four main problems. First, in most of the cases we need significant changes in the routing protocol, i.e., BGP. Any implementation will go through an infant period with new bugs and new problems to solve. Second, most require a significant amount of processing power, and the current routers may not able to keep up. For example S-BGP increases the resources needed by 800% [15]. Third, none of the current approaches has been fully approved by the community (IETF). Additionally, there exist serious considerations [21] in determining whether the path of any path vector protocol can be verified, since a network can advertise one thing to its peers and another internally. Last, but not least these proposals focus solely on how to prevent the routing errors while completely ignore the human usability. Complex solutions can steer away operators from some very useful and probably needed approaches.

In this paper, we are interested in investigating the potential for improving the security of Internet routing today. Instead of taking a proactive approach that prevents routing errors from happening, we propose to use a reactive approach. This cannot prevent abnormal routing, but it could alleviate easy attacks, before they become widely used, for example AS number and IP hijackings [22]. Our approach is based on the knowledge of the intended Internet routing. If we know what Internet routing should be, we can detect abnormal routing behavior. Two components are needed to achieve this: 1) accurate information on the policy and configuration of an AS, 2) a way to detect deviations from the expected routing. The policy of an AS can be described using the RPSL language, and there exist public repositories that networks can use to publish their policy. Additionally, we need a way to monitor Internet routing. There exist a number of monitors like Routeviews [23] and the RIS [24] project in Ripe, that exist for the sole purpose of recording Internet routing for operational and research purposes. In our previous work [25], we showed how we can extract useful information from the registries. Here, we will use part of the information for the purpose of validating Internet routing.

Our contributions can be summarized as follows:

- We propose a new approach to improve the security and robustness of BGP by monitoring its operation.
- We demonstrate the efficacy of our approach by applying it to RIPE to validate the European Internet Routing.
- We analyze for 10 days the European Internet routing and examine over 4 million updates. This allow us to check the sanity of $23,210$ distinct European IP prefixes. We find that for 97% of these prefixes we can validate their origin AS in the RIPE registry.
- After cleaning and processing the data, we could check the policy of 65% of the European ASes. We find that for 60% of the European ASes, a suprising high percentage, we could find no routing deviations during the 10 days we examined. Additionally, we find that 40% of the policy deviations are transient, and last for less than 100 seconds.

Can we expect the IRR information to be sufficiently accurate for our approach to work? We claim that this is a typical chicken and egg problem. We argue that tangible benefits could provide a compelling reason for operators to keep IRR accurate and up to date. This is true for the RIPE registry, where for 60% of the ASes, we can find no deviation between the intended policy and the actual policy as seen in BGP. Additionally, even the current proposals require some form of registry. For example the SoBGP approach requires knowledge on the topology of the network. In addition, the tool can be used incrementally even with locally correct information: between neighboring ASes. Second, we argue that collaboration and information sharing seem to be critical for a holistic approach to Internet security, since the security of an interconnected system is equal to the security of its weakest component. It is worth noting that we have presented our work, vision, and tools on IRR to network operators, and their feedback have been quite encouraging [26] [27].

The rest of this paper is structured as follows. In section II we present some definitions and background work. In section III we describe our framework. In section IV, we present how RIPE can use our approach to improve the security of the European Internet routing. In section V we discuss the necessary steps to make our approach even more effective and discuss about the practical potential of our tool. In section VI we present our conclusions.

II. BACKGROUND AND PREVIOUS WORK

In this section, we briefly describe an overview of Internet routing, the threats for its secure operation, and some proposed solutions. Then, we briefly present the Internet Routing Registries and the language used to describe the routing policy.

A. Internet and BGP-4

Internet is structured into a number of routing domains that have independent administrations, called **Autonomous Systems (AS)**. Each autonomous system is identified by a number, **asn**, which is assigned to it by an Internet registry. An Autonomous System uses an intra-domain routing protocol, like OSPF or IS-IS, inside its domain, and an inter-domain protocol to exchange routing information with other Autonomous Systems. The defacto standard for inter-domain routing is **BGP-4** [2]. The primary difference between the intra-domain and the inter-domain protocol is that the first one is optimized for performance, solely based on operational requirements, while the second is used to enforce the **policy** of the Autonomous System, which corresponds to the **business relations** with its neighboring ASes.

An Autonomous System given its policy, will advertise to its neighbors a list of **IP Prefixes**, or **routes** that are reachable through it. Each route is tagged with a number of **attributes**. The most important attribute is the **AS_PATH**. The AS_PATH is the list of ASes that packets towards that route will traverse.

An AS uses **filters** to describe what it will import from and export to a neighboring AS. The filter can include a list of routes, a list of regular expressions on the AS_PATH, a list of communities, or any possible combination of these three. Filters can have both positive and negative members. For example we can explicitly reject routes that are either private [28], or reserved [29].

B. Threats and proposed solutions

BGP has evolved in an incremental way in-order to address the security requirements that threatened its robust operation. The major event of AS7007 [7], which de-aggregated and advertised a large portion of the Internet in 1997, caused the widespread use of prefix filtering. Additionally, the Max Prefix Limit parameter was introduced to provide an upper limit on the number of prefixes a router will accept from a peer. In the mid 90's Internet routing was plagued by excessive routing instability. The route flapping mechanism [4] was introduced to address this problem and is used to suppress the number of advertisements of an unstable network. Recently, a flaw in the TCP protocol, used by BGP, allowed in theory a remote host to tear-down the BGP connection between two peers. This lead to the broad use of MD5 signatures [3] to secure the peering connections, and the introduction of the TTL security mechanism [6] (GTSM), a simple but powerful mechanism to prevent BGP spoofing. These mechanisms have notably improved the overall security of BGP, and combined with the use of the best common practices by the network operators can severely limit possible damage to BGP routing.

On the other hand, the significance of BGP dictates that more actions should be taken to improve its security. The lack of any mechanisms to verify the correctness of the routing information has been scrutinized recently [11] [12] [15] [30] [13]. Potentially, it can lead to problems like IP hijackings, DDOS etc. A number of proposals exist to add to BGP capabilities to verify the routing information. Next, we present in more detail two of the most popular, S-BGP and SoBGP.

S-BGP:S-BGP has three main security mechanisms. First, they use Public Key Infrastructure (PKI) to authenticate every aspect of a routing message, like the ownership of an IP address, the ownership of the Autonomous System (AS) that originates that prefix, the identity of the AS, and so on. Second, they introduce a new optional transitive attribute to carry digital signatures. A router can use the signatures and the information in PKI to validate the BGP update information. Third, they are using IPsec for point-to-point security. The main characteristic is the number of digital signatures required to verify an update. They need to verify every AS in the path, and the origin AS. Additionally, when an AS sends an update to its neighbors, their approach needs to calculate a new one for every neighbor that it has. In their approach they can use cache to improve performance, but when we have changes in either the topology, or in the policy they need to recompute them. They mention that for a router with 30 peers, they can achieve 9 operations per second, with no cryptographic hardware installed in the router.

SoBGP: The SoBGP approach has two main goals. First, to validate the authorization of an AS to advertise an IP prefix. Second, to prove that there exist at least one valid path to the destination. SoBGP is using three certificates to sign information. The certificates are the Entity certificate, which is used to authenticate the identity of an autonomous system. This certificate can be signed by any organization that the receiver trusts. The authorization certificate ties an AS to the IP prefixes that is allowed to advertise. This certificate can be signed by the organization that delegated the IP prefix to the AS. Last, they have the policy certificate, which describes the policies related to the IP prefixes and the topology of the advertising AS. This certificate is signed using the private key of the AS. The keys and certificates, form a web of trust, and no central repository of keys exist, or PKI infrastructure like the one S-BGP requires. The certificates are per Autonomous System, and every AS is responsible for storing its own database of keys and certificates.

In order to prove that the path exist, they are building a directed graph using the information stored in the policy certificates, and check if the path is feasible. Another important component of their approach is that

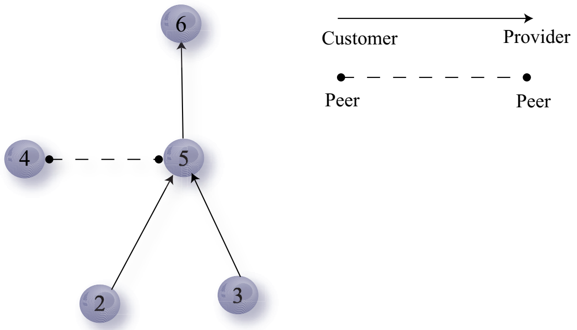

Fig. 1. A simple AS level topology.

they don't require any encryption processing to be done on the routers. The processing can either be done on the routers or it can be done by servers in an offline manner. Additionally, in order not to delay the speed of converge, the operators can choose either to first advertise the updates and then validate them, or first validate them and then advertise them. Their proposal leaves a vast number of choices to the operator, and has many different modes of operation. This is criticized in [15] by the authors of S-BGP and they mention that this can in-adversely impact the interoperability of the protocol.

C. Internet Routing Registries and RPSL

The need for cooperation between Autonomous Systems is fulfilled today by the **Internet Routing Registries (IRR)** [31]. ASes use the **Routing Policy Specification Language (RPSL)** [32] [33] to describe their routing policy, and router configuration files can be produced from it. At present, there exist 62 registries, which form a global database to obtain a view of the global routing policy. Some of these registries are regional, like RIPE or APNIC, other registries describe the policies of an Autonomous System and its customers, for example, cable and wireless CW or LEVEL3. The main uses of the IRR registries are to provide an easy way for consistent configuration of filters, and a mean to facilitate the debugging of Internet routing problems.

The design goal of RPSL is twofold. First, RPSL provides a standard, vendor independent language, so that the policy of an AS can be published in an easy to understand format. Second, RPSL provides high level structures for a more convenient and compact policy specification. RPSL provides an abstract representation of policy, but still the policy described is based on filters on routes, on regular expressions on the AS_PATH, and on communities. There exist 12 different classes of records, that either describe portion of a policy, or describe who is administering this policy. In figures 1 and 2, we have an example topology and the corresponding RPSL records for an Autonomous System. The route class is used to register the IP prefixes or routes an AS owns and originates. The as-set and route-set classes are high level structures that can be used to group routes. For example an AS can create a route-set that will contain the routes of its customers. Finally, the aut-num class contains the import and the export policies for every neighbor of the AS. Note that every class has a mnt-by attribute that specifies the maintainer of the record. This is done for security reasons so that only the maintainer can update that record. There exist additional attributes, not shown in the figure, like the source attribute that specifies in which registry the record exists, and the changed attribute that provides the date that the record was either last updated or created. In our previous work [25], we have developed a methodology to analyze the policy register in the registries. Using our tool we can reverse engineer the policy of an Autonomous System, and check for possible errors.

```
as-set:     AS-5
members:    AS5, AS5:AS-CUSTOMERS
mnt-by:     AS5-MNT

as-set:     AS5:AS-CUSTOMERS
members:    AS2,AS3
mnt-by:     AS5-MNT

route:      199.237.0.0/16
origin:     AS5
mnt-by:     AS5-MNT

aut-num:    AS5
import:     from AS6 action pref = 100; accept ANY
import:     from AS4 action pref = 90;
                 accept <^AS4+ AS4:AS-CUSTOMERS*$>
import:     from AS2 action pref = 80;  accept AS2
import:     from AS3 action pref = 80;  accept AS3
export:     to AS6 announce AS-5
export:     to AS4 announce AS-5
export:     to AS2 announce ANY
export:     to AS3 announce ANY
mnt-by:     AS5-MNT
```

Fig. 2. Example of RPSL policy for Autonomous System 5

III. FRAMEWORK FOR SECURITY

We develop a framework to detect abnormal routing behavior by using the Internet Routing Registries. We first present an overview of our framework and then discuss in detail its two main components. The first component is how we process the registered policy in IRR. The second component is how we discover abnormal routing using the registered policy.

A. Problem overview

The problem we are trying to solve is the following. Assume that a router receives an update from a peer for the prefix $62.1.0.0/16$ with path $\{15623\ 702\ 1241\ 8573\}$. We want to check if this update is legitimate and valid. To verify the update we must answer the following questions:

- Does the destination AS have the authorization to advertise the IP prefix? In our case is $AS8573$ authorized to advertise the prefix $62.1.0.0/16$? There can be three different valid cases. First, the AS was assigned the IP space directly from an authority like RIPE. Second, the AS is using the space that is owned by one of its providers. Third, the AS that originates the IP prefix has aggregated many shorter IP prefixes, usually of its customers, and appears to be the origin AS.
- Does the path towards the destination exist? Two conditions must be met in-order for this to be true. First, every node in the path must have as neighbors the ASes that appear to be adjacent to it. For example, $AS702$ must connect to both $AS15623$ and $AS1241$. Second, the filters of every AS in the path should allow the propagation of the destination prefix. This means that the filters should allow $62.1.0.0/16$ to be imported and exported.
- Does the path conform to the local policies of the ASes? BGP routing is based on the business relations between the peering ASes. For example, based on the valley free path concept, $AS1241$ can not have both $AS702$ and $AS8573$ as its providers. If it is true, it shouldn't advertise the prefix to 702.
- Does the update conflict with any prefix we already have in the routing table. For example, if we have in the routing table the IP prefix $62.1.9.0/24$ with path $\{15623\ 8223\}$. This prefix is a more specific prefix than the $62.1.0.0/16$ we want to validate. In order for the update to be valid, $AS8573$ and $AS8223$ must be connected. Consider the scenario that the IP prefix $62.1.9.0/24$ is withdrawn temporary, packets towards that prefix will be routed to $AS8573$ which must be able to reach 8223.

If we answer positively in all these questions, we consider the update to be legitimate.

B. Intended policy of an AS

We use the RPSL records that exist in the various IRRs to find the intended policy of an AS. The policy of an AS is the end-product of the business relations it has with its neighbors. An AS can adjust its policy by manipulating the import and export filters. The import filters describe what it expects or allows to import from its neighbors, and the export filters can be used to selectively specify what the AS wants to transit. Additionally, the AS can assign a preference value on what it imports usually on a neighbor basis. The common practice is to prefer the routes from your customers, then your peers and last your providers. The vast majority of the ASes in IRR describe their policy at the granularity of AS numbers, and we will describe the policy of the AS also at that level.

Let us first make some definitions. For every AS A given its policy as described in the route, aut-num and set records, we collect the following information.

- $Origin[A]$: The list of IP prefixes AS A registers, by using the route records.
- $Links[A]$: The list of neighbors AS A registers.
- $Import[A, B]$: For every neighbor B of A, $import[A, B]$ is the list of ASes that A will import from B.
- $Export[A, B]$: For every neighbor B of A, $Export[A, B]$ is the list of ASes that A is exporting to neighbor B.
- $Policy[A, B, C]$: We use the $Policy[A, B, C]$ to describe the list of ASes that A is importing from C and exporting to B.

To compute the $Policy[A, B, C]$ relation, we will use the methodology and tool we developed in our earlier work [25]. Computing the Policy is far from trivial, and there exist many details that we can not describe here. In a nutshell, our tool is taking advantage of the hierarchy build in the sets and limits the members of the sets that are visible to an AS. These tables can completely describe the policy of an AS, and can be used to detect abnormal routing behavior.

C. Detect abnormal routing behavior

Given a router C and its routing table, and the IRR that describes the policies, we want to find whether an update for prefix I and path $P_I = \{a_1, a_2, ..., a_n\}$ is valid. The tests are the following:

- a_n can be the origin of I. Either one of the following three options must be true. $Origin[a_n]$ contains I. If $Origin[i]$ contains I, then either $Links[a_n]$ contains i, or $Links[i]$ contains a_n.
- for every a_i in P:
    - $Links[a_i]$ contains both a_{i-1} and a_{i+1}. This means that a_i registers a_{i-1} and a_{i+1} as its neighbors.
    - $Import[a_i, a_{i+1}]$ contains a_n, which means that the import filter a_i uses on its neighbor a_{i+1} allows a_n to be imported.
    - $Export[a_i, a_{i-1}]$ contains a_n. The export filter that a_i uses to describe what it exports to a_{i-1} contains a_n.

  - $Policy[a_i, a_{i+1}, a_{i-1}]$ contains a_n. In other words, a_i will act as a transit for a_{i+1} for AS a_n.
- For every prefix E with path $P_E = \{b_1, b_2, ..., b_n\}$ in C, such that prefix E contains prefix I, check that there exist in IRR a path between a_n and b_n.

We consider the policy of an AS a_i to be fresh and correct if for all the updates and routing tables we analyze, all the requirements we mention above are met.

IV. CASE STUDY: EUROPEAN INTERNET ROUTING(RIPE)

In this section, we show how our approach can be used to check the consistency of the European Internet routing. We start with presenting the data sets that we use and an overview of the data we process. Next, we check the origin AS of the updates, and show that RIPE contains accurate information. Finally, we check the validity of the path and we present our results.

A. Data and Methodology

We process the RIPE registry and the RIS [24] router rrc03 at AMS-IX in Amsterdam for a period of 10 days starting at June, 03, 2004. The rrc03 router had 86 active peers during that time period, and it is the best connected router among all other routers that are part of the RIS project. We start with the routing table of rrc03 collected at June, 03, 2004, and we apply the updates that the router received for the next 10 days. Additionally, during these 10 days, we download and process the IRR registries daily so that changes in IRR reflect back to our model of the intended policy. For our analysis, we are only interested for the prefixes that are assigned to RIPE by IANA [29]. The address space chunks we monitor are the following: 62/8, 80/5, 88/8, 193/8, 194/7, 212/7, 217/8. In order to analyze the prefix and path tuple, we check if the prefix is part of the prefixes administered by RIPE. We analyze the tuple only if the prefix is part of the RIPE prefixes, given that we are interested on the European Internet routing.

Using this methodology, we observe $23,210$ distinct prefixes during the time period of 10 days. In figure 3, we plot the number of prefixes that appear in the routing table of rrc03[1]. It is worth noting the difference after the 5th day, where in the duration of the next two days almost $1,000$ new prefixes were added to the routing table. The reason for this increase is that a number of ASes started advertising more specific prefixes together with the less specific one. When we started our experiment, the routing table had $21,811$ distinct prefixes during the first day, and $22,864$ during the last one. In figure 4, we plot the number of updates the router at rrc03 receives per day that are relevant to the RIPE prefixes. The peak is on the 9th day with close to $670,000$ updates, while the lowest number of updates is on the 4th day with a little over $226,000$ updates. In total, during these 10 days we processed $4,156,340$ updates plus the original $400,025$ prefix-path tuples of the routing table.

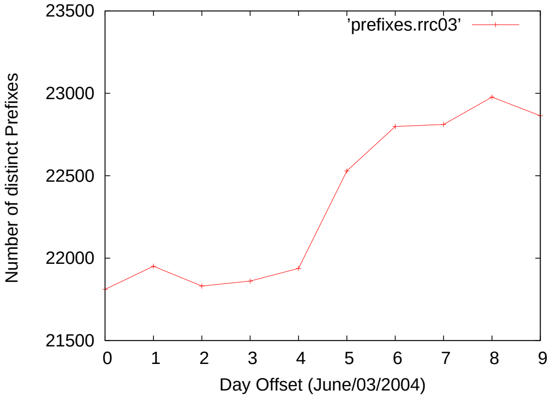

Fig. 3. The number of RIPE prefixes found in rrc03 per day.

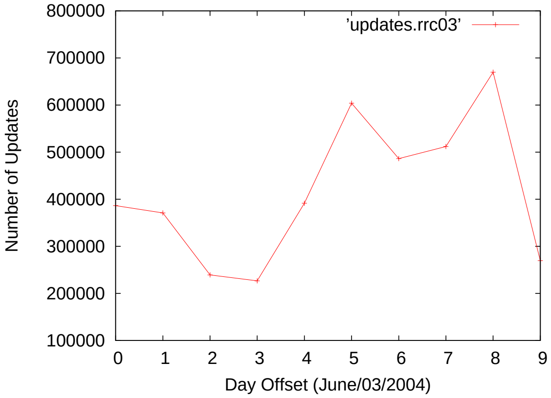

Fig. 4. The number of updates for the RIPE prefixes that we analyze.

B. Origin validation

Next, we study whether we can verify with our intended policy model, the origin AS of every prefix-path tuple. In figure 5, we have the evolution of the number of prefix-origin tuples where the origin can be validated. The total number of the tuples that their origin can be validated is $22,791$. This means that over 97% of the tuples can be validated using the RIPE registry. As we can see in the figure, the number of tuples that we can validate is increasing with time. This is happening because in the same time period the number of prefixes we observe is increasing.

---

[1] Note that we compute the routing table by applying the updates

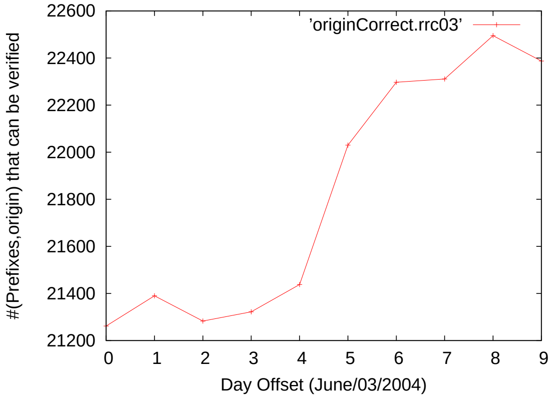

Fig. 5. The evolution of the number of prefix,origin that can be verified in RIPE.

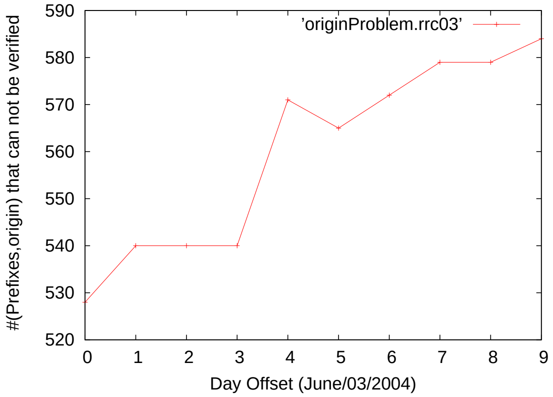

Fig. 6. The evolution of the number of prefix,origin that can not be verified in RIPE.

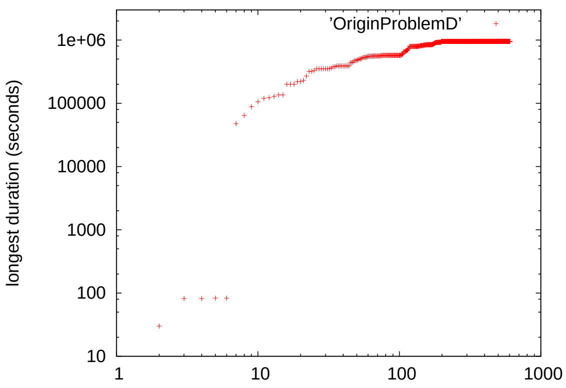

Fig. 7. The longest duration in seconds of the prefix,origin tuple that can not be verified in RIPE.

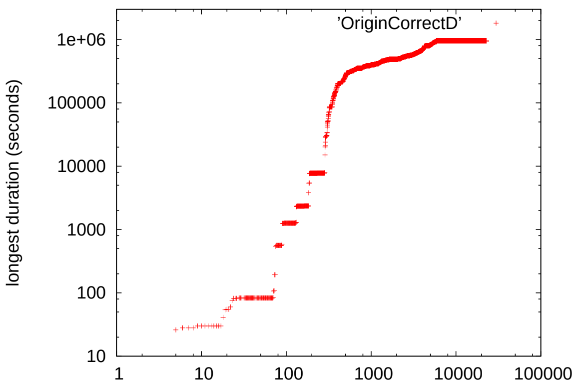

Fig. 8. The longest duration in seconds of the prefix,origin tuple that can be verified in RIPE.

In figure 6, we have the evolution of the number of prefix-path tuples that we can not validate their origin AS. The total number of these cases is 612. As with the previous figure, we see that the number of tuples is increasing with time, again this is happening because we have more prefixes. Additionally, it seems that the problems seems to be persistant, something that indicates that we can not validate them because the registry doesn't contain the appropriate route records.

Next, we want to understand better the persistence of the errors. In figure 7, we plot the maximum continuous time we observe a prefix-path tuple with an origin mismatch. We find that only 5 cases can be classified as short-lived, something that can classify them as possible errors. These five instances appear in the routing table for less than two minutes. The next problematic origin appears continuously for over 13 hours. In figure 8, we plot the maximum continuous time we observe the prefix-path tuple for the cases where we can validate the origin AS. Again as with the previous figure, we can see that some prefix-path tuples last for an extremely small amount of time. We have 70 cases where the prefix is observed for less that 100 seconds. Currently, we don't have any explanation, but it could be interesting to try to understand why this phenomenon is happening.

To summarize our results on the origin validation phase, the percentage of 97% of the prefixes that can be validated, shows that the route records in the RIPE registry are meticulously maintained. There exist records that contain inaccurate information, but the vast majority of the records are kept accurate. One of the reasons is that the European operators use the RIPE registry to automate the generation of filters. Usually, one of the requirements for peering is to maintain route records in RIPE. Another reason is that RIPE requires the ASes to publish their route records. Additionally, they have a number of projects to check the consistency of their registry.

C. *Path validation*

The next step is to check if the path of the prefix-path tuple is valid. There exist a number of tests. First, we check if the path is feasible, and then if the path is valid based on the policy of the ASes.

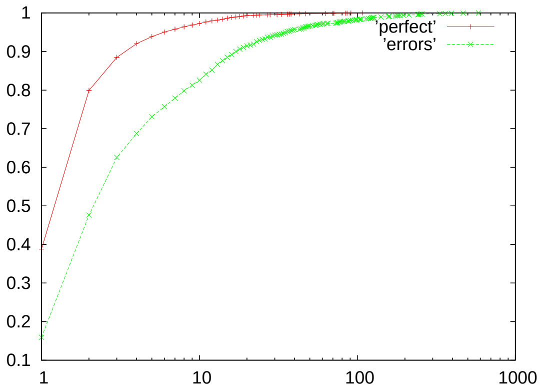

Fig. 9. The CDF of the degree for the ASes that all their links exist in IRR (perfect), and for the ASes that only some exist (errors)

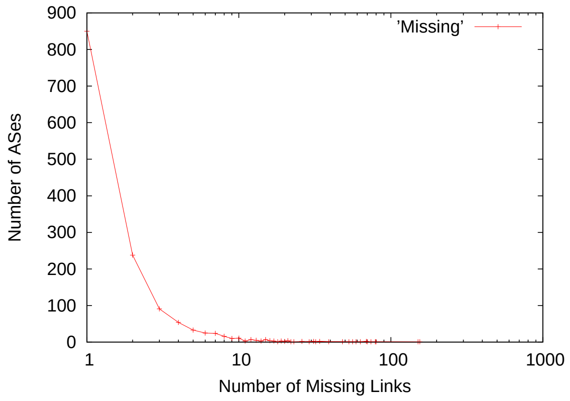

Fig. 11. Number of ASes we could analyze if we allow a number of missing links.

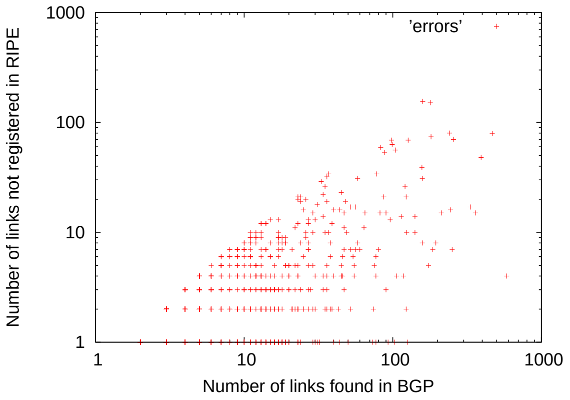

Fig. 10. Number of links not registered in RIPE versus the total number of links found in BGP.

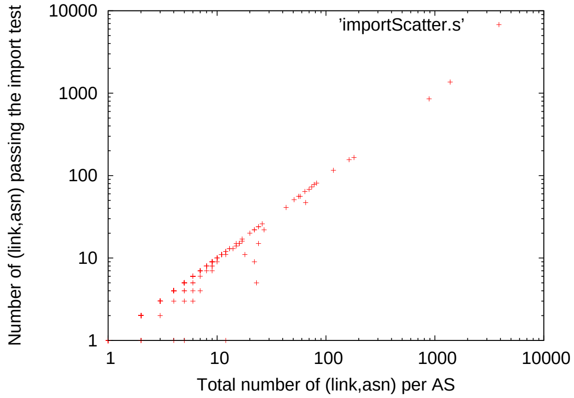

Fig. 12. Scatter plot of the (link,asn) that pass the import test versus the total number per AS.

In order to check the path feasibility, we first need to check if all the ASes register all the links we find in BGP. We find that for 65% of the European ASes, $3,488$ ASes, all their links in BGP are registered in the RIPE registry. In figure 9, we plot the degree CDF of the ASes that register all their links, the perfect line, and the degree CDF of the ASes that don't register all links, the errors line. As shown in the figure, most of the ASes that register all their links have relative small to medium size. Only 3% of these ASes have more than 10 neighbors, and the largest AS which registers all the links has degree 108. On the other hand, 10% of the ASes that don't register all links have a degree of 18 or larger. To better analyze the ASes that don't register all links, we plot in figure 10 the number of links that are not registered versus the total number of links that an AS has in BGP. As we can see from the figure there exist wide variations among the ASes that have some links missing. It is worth noting that the AS with the highest degree in BGP register 580 out of the 584 links. If we allow a number of missing links, we can analyze many more ASes. In figure 11, we plot the number of ASes that we can analyze versus the number of missing links that we should ignore. As you can see in the figure, the vast majority of the ASes misses a few links. For example, if we allow two missing links, we can processed more than $1,000$ additional ASes. On the other hand, if an AS doesn't register all the links, we can not analyze the filters on that link or the policies, and so for the remaining of this section we will ignore the ASes that don't register all links, and focus only on those that register all.

Next, we validate the filters. We find that most of the import and export filters can be validated. In figure 12, we plot per AS the number of link-asn tuples that pass the import test versus the total number of link-asn tuples of an AS. Out of the 522 ASes that have import filters[2], 47 ASes fail in one or more tuples. This mean that we can not find in their import filters all the ASes that they should import. It is worth noting that most of the ASes

---

[2] We can check the import filters only for transit ASes, this is why we only have 522 ASes

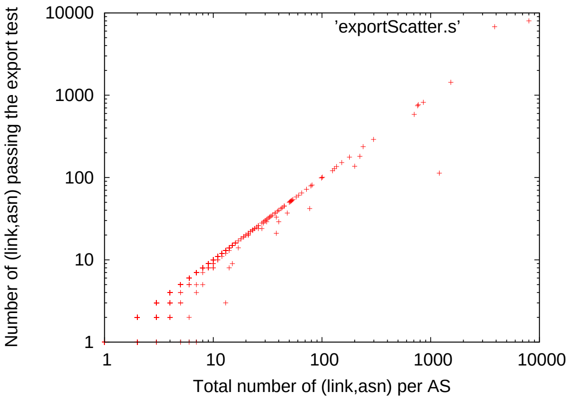

Fig. 13. Scatter plot of the (link,asn) that pass the export test versus the total number per AS.

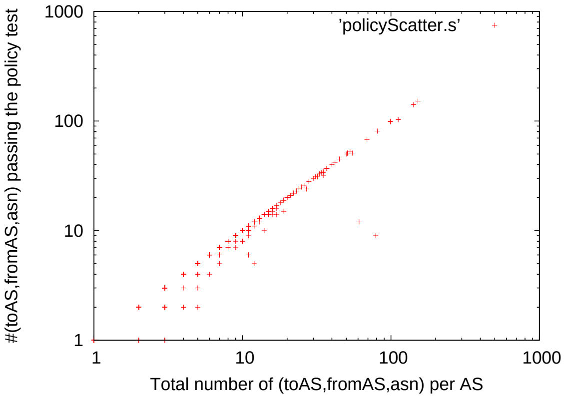

Fig. 14. Scatter plot of the (toAS,fromAS,asn) that pass the policy test versus the total number per AS.

that fail have a small number of tuples.

Next, we check the export filters. We have $3,488$ ASes that register all the links and have exports. In figure 13, we plot per AS the number of link-asn tuples that pass the export test versus the total number of tuples of an AS. Out of the $3,488$ ASes, 3308 ASes have no problem with their export filters. Again, as with the previous figure we observe that most of the ASes with problems have a small number of export tuples.

In order to check if the policy of an AS is correct, it must register all the links and must not have problems with its import and export filters. This holds for $3,281$ ASes. In figure 14, we plot the number of tuples that pass the policy test versus the total number of tuples per AS. Out of $3,281$ ASes, 58 fail to pass this test. This leaves us with $3,223$ ASes that pass all our tests, or 60% of all the European ASes. Next, we want to check why some ASes failed in this new test. In figure 15, we plot the CDF of the longest appearance of the tuples that have policy problems. As we can see from the figure, unlike the similar origin plots, a significant number of problematic cases last for a very small amount of time. We observe that over 40% of the errors last for less than 100 seconds. This short appearance and disappearance makes them a candidate for abnormal routing behavior, and we can categorize them as transient errors.

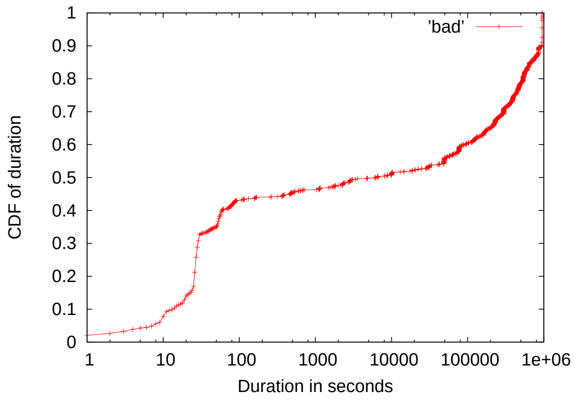

Fig. 15. The CDF of the maximum number of seconds a bad policy lasts.

To summarize our results in the path validation, we see that when all the links are registered only a small percentage of these ASes fail on the tests, and the failure is mostly due to the poor description in RPSL of the policy of the AS(180 ASes fail on the export test).

An important question remains why the larger ASes don't register all their neighbors? They typically register the majority of their neighbors but not all. One reason might be that the large ASes need to update their policy much more frequently than the smaller ASes. Since the policy as stored in IRR is not used in any tangible way other than in general debugging of Internet routing problems, the larger ASes don't bother updating their policies very frequently. Another factor can be that they hesitate to disclose their local policies, which they consider to be confidential, especially if there is no benefit for them. In any case, 60% of the ASes is a significant portion of the European ASes.

We should note here that the last test we analyzed, the cases where we have more specific prefixes with different origin AS than the less specific prefix, didn't produce any new problems. For all the cases we examined, 587, we have either classified them as an error with the previous analysis, or a path existed between the two origins.

V. DISCUSSION

In this section, we discuss the vision that we have on how our approach can be deployed. In addition, we mention the advantages and benefits of our approach.

Deploying our approach: the vision. First, we need to clarify that our approach encourages and relies to some extent on collaboration between ASes, but it does not need a centrally controlled Internet. Clearly, a centrally managed Internet could be made secure if it could overcome scalability issues. However, the Internet is distributedly run for a variety of civil, business and operational reasons. Our approach is aligned with this requirement.

In our vision, IRR could become a more sophisticated database, where multiple views and various levels of access to information could be provided. For example, an AS operator could be allowed to retrieve more information about a neighbor AS and less information about a distant unrelated AS. Similarly, a network operator could have more clearance and access to details than a researcher. In other words, we can shift the security and privacy issues to the access of the IRR registry, which is something that falls into the database security and information access category. This way, IRR could be implemented in a variety of ways, such as a distributed database with multiple views of access. Alternatively, IRR could consist of multiple physically different databases, and an AS should be updating each one separately.

Our approach could significantly benefit from the addition of automated consistency checking in the registries. The more accurate information the better we can detect routing problems. To this effect, the registries can have automated tools for consistency checks. For example, when one AS registers a link, while the neighbor AS does not. Note that many such checks are easy to automate [34] [35] and they can even generate notifications in a web-log or email form.

In a nutshell, the point of this work is to show the power of information sharing and collaboration. Having this, and the appropriate tools, we could automate and speed up the detection of routing errors. Implementing a secure and privacy-aware IRR infrastructure is a separate and technically feasible issue.

The advantages of our approach. We list several advantages that our approach provides. First, by automating the update validation, we decrease the window of opportunity for malicious users. If we can detect abnormal routing fast enough, we can limit the profits from illegal routing. After that, it is up to the community to find ways to act or enforce a solution through recovery mechanisms or business practices. For example, today, a spammer can hijack a route, or an AS number to send spam for a number of days or weeks, until either he is discovered, or the routes he uses are blacklisted. At that point it just hijacks another route. Second, it can limit human errors indirectly by encouraging the use of IRR and the related tools that come with it. Finally, our approach can offer limited protection against malicious users, for example terrorists, which may attempt a massive routing attack. Again, our approach could provide a quick detection of the problem and a potentially fast response, even in the form of a shutdown of affected parties.

VI. CONCLUSIONS

We develop an approach to improve the security and robustness of Internet routing with the tools that exist today, the Internet Routing Registries. Our approach has a large number of benefits. First, no changes are required in the routing protocol and therefore it can be used with minimal disruption. Second, there is no need for global cooperation, and conformance. Networks that publish their policies can use our approach. Third, we increase the accountability of Internet routing and automate the discovery of routing anomalies. Fourth, monitoring of Internet routing can help us separate hype from reality. Which problems are real, how often do they appear? Convery et.al. [36], showed that even though theoretically it is possible for an external attacker to create problems like BGP spoofing, in reality it is extremely difficult to make a succesful attack.

Other practitioners have been interested in similar approaches. For example RIPE has developed a prototype, myAS [37], for a similar purpose. Their tool allows administrators to manually register the routes they want to safeguard, and their upstream providers. They use the RIS monitors to detect deviations from the registered policy, and inform the network administrator of the problems. Our approach is much more ambitious and is motivated by this question: why not use the actual RPSL records described in the RIPE registry for route validation?

We believe that our approach can be used *today* towards a more secure Internet routing. The different elements needed by our approach already exist. In conclusion, our approach can be used to protect Internet routing and automatically evaluate, with little or no human intervention, the extent of the problem before deciding to take extra steps to add security within the Internet infastructure.

Our contributions can be summarized as follows:

- We propose a new approach to improve the security and robustness of BGP by monitoring its operation.
- We demonstrate the efficacy of our approach by applying it to RIPE to validate the European Internet Routing.
- We analyze for 10 days the European Internet routing and examine over 4 million updates. This allow us to check the sanity of $23,210$ distinct

European IP prefixes. We find that for 97% of these prefixes we can validate their origin AS in the RIPE registry.
- After cleaning and processing the data, we could check the policy of 65% of the European ASes. We find that for 60% of the European ASes, a suprising high percentage, we could find no routing deviations during the 10 days we examined. Additionally, we find that 40% of the policy deviations are transient, and last for less than 100sec.

Through this analysis, we get strong evidence of the effectiveness of our approach. We believe that IRR has an important role in the future operation of the Internet. Using IRR or something similar could be the basis for a deployable approach to improving the Internet today. This can be the first step until proactive approaches and next generation hardware are ready to be deployed.

REFERENCES

[1] "Routing protocols security working group," http://www.rpsec.org/.

[2] Y. Rekhter and T. Li, "A Border Gateway Protocol 4 (BGP-4)," RFC 1771, 1995.

[3] A. Heffernan, "Protection of BGP sessions via the TCP MD5 signature option," RFC2385.

[4] C. Villamizar, R. Chandra, and R. Govindan, "BGP route flap damping," RFC2439.

[5] R. Chandra, P. Traina, and T. Li, "BGP Communities Attribute," RFC1997.

[6] V. Gill, J. Heasley, and D. Meyer, "The generalized TTL security mechanism (GTSM)," RFC3682.

[7] Stephen Misel, "Wow, as7007!," http://www.merit.edu/mail.archives/nanog/1997-04/msg00340.html.

[8] W. B. Norton, "The art of peering: The peering playbook," *Draft*.

[9] R. Mahajan, D. Wetherall, and T. Anderson, "Understanding BGP misconfigurations," *ACM Sigcomm*, 2002.

[10] N. Feamster, J. Winick, and J. Rexford, "A model of BGP routing for network engineering," *IEEE Sigmetrics*, 2004.

[11] Sandra Murphy, "Routing protocol threat analysis," INTERNET DRAFT, 2003.

[12] Sandra Murphy, "BGP security vulnerabilities analysis," INTERNET DRAFT, 2003.

[13] S. Convery, D. Cook, and M. Franz, "An attack tree for the border gateway protocol," Internet Draft.

[14] Stephen Kent, Charles Lynn, and Karen Seo, "Secure border gateway protocol (S-BGP) architecture," *IEEE JSAC Issue on Network Security*, 2000.

[15] Stephen Kent, "Securing the border gateway protocol: A status update," *Seventh IFIP TC-6 TC-11 Conference on Communications and Multimedia Security*, 2000.

[16] Ng James, "Extensions to BGP to support secure origin BGP (sobgp)," Internet Draft, 2002.

[17] Geoffrey Goodell, William Aiello, Timothy Griffin, John Ioannidis, Patrick McDaniel, and Aviel Rubin, "Working around BGP: An incremental approach to improving security and accuracy of interdomain routing," *Symposium on Network and Distributed Systems Security*, 2003.

[18] Yih-Chun Hu and Adrian Perrig, "SPV: A Secure Path Vector Routing Scheme for Securing BGP," *ACM Sigcomm*, 2004.

[19] Lakshminarayanan Subramanian, Volker Roth, Ion Stoica, Scott Shenker, and Randy H. Katz, "Listen and whisper: Security mechanisms for BGP," *First Symposium on Networked Systems Design and Implementation*, 2004.

[20] Xiaoliang Zhao, Dan Pei, Lan Wang, Dan Massey, Allison Mankin, S. Felix Wua, and Lixia Zhang, "Detection of invalid routing announcement in the internet," *International Conference on Dependable Systems and Networks (DSN'02)*, 2002.

[21] R. White and N. Feamster, "Considerations in validating the path in routing protocols," *IETF Draft, Work in Progress*, 2003.

[22] Leslie Nobile and Leo Vegoda, "Address space & as number hijacking," *Ripe-48*, 2004.

[23] University of Oregon Route Views Project, "Online data and reports," http://www.routeviews.org/.

[24] "Routing information service(ris)," www.ris.ripe.net.

[25] Georgos Siganos and Michalis Faloutsos, "Analyzing BGP policies:methodology and tool," *IEEE Infocom*, 2004.

[26] Georgos Siganos, "Nemecis: A tool to analyze the IRR registries," *Nanog 30, Miami, http://www.nanog.org/mtg-0402/siganos.html*, 2004.

[27] Georgos Siganos, "Nemecis: A tool to analyze the Internet Routing Registries," *RIPE 48, Amsterdam, http://www.ripe.net/ripe/meetings/ripe-48/eof.html*, 2004.

[28] Y. Rekhter, B. Moskowitz, D. Karrenberg, G.J. de Groot, and E. Lear, "Address allocation for private internets," RFC-1918.

[29] "Internet Protocol V4 Address Space assignments," http://www.iana.org/assignments/ipv4-address-space.

[30] Ola Nordstrom and Constantinos Dovrolis, "Beware of BGP attacks," *ACM Sigcomm Computer Communication Review*, 2004.

[31] "Internet Routing Registries," http://www.irr.net/.

[32] Alaettinoglu, C. Villamizar, E. Gerich, D. Kessens, D. Meyer, T.Bates, D. Karrenberg, and M.Terpstra, "Routing Policy Specification Language(RPSL)," RFC2622.

[33] D.Meyer, J.Schmitz, C.Orange, M. Prior, and C. Alaettinoglu, "Using RPSL in practice," RFC2650.

[34] "Nemecis tool website," http://ira.cs.ucr.edu:8080/Nemecis/.

[35] "Routing Registry Consistency Check project," http://www.ripe.net/ripencc/pub-services/db/rrcc/index.html.

[36] Sean Convery and Matthew Franz, "BGP vulnerability testing: Separating fact from fud v1.00," Nanog 28, 2003.

[37] "MyAS Prototype," RIPE, http://www.ris.ripe.net/myas/.