University of Amsterdam

Detecting routing anomalies with RIPE Atlas

Author:
Todor Yakimov

Supervisor:
dr. J.J. van der Ham
Barry van Kampen

April 2014

Abstract

Routing anomalies are a common occurrence on Today's Internet. Given the vast size of the Internet, detecting such anomalies requires having a large set of vantage points from which to be able to schedule detection tests. An initiative of the RIPE NCC, RIPE Atlas is a globally distributed Internet measurement system that offers a favorable number vantage points. The project examines whether the technical capabilities of RIPE Atlas can be instrumented for the detection of three types of routing anomalies, namely Debogon filtering, Internet censorship and BGP prefix hijacking. By examining existing methodologies for detecting routing anomalies, the project defines a number of tests in RIPE Atlas. The tests examine whether RIPE Atlas is a viable replacement to current detection system and what limitations it presents.

Contents

Chapter 1

Introduction

Routing anomalies are commonly seen on Today's Internet. They range from simple misconfigurations in the internal infrastructure of Internet Service Providers (ISP) to faulty Border Gateway Protocol(BGP) updates in the Internet's control-plane that may cripple the connectivity of entire countries or geographic regions. Amongst the various routing anomalies that exist, the project strives to examine a subset of them that connect to major causes of traffic filtering, misdirection and interception. Depending on the type of routing anomaly and whether it is occurring in the Internet's control- or data-plane, a number of different approaches can be taken to detect it. Traditionally, data-plane anomalies are detected by using tools such as traceroute, ping and DNS queries for verifying two-way reachability, taken paths from sources to destinations and fundamental differences in-between name resources and Layer 3 endpoints. Anomalies in the control-plane are examined by a collection of steps that employ various datasets so that concise detection can be achieved. In both cases, given the vast size of the Internet, it is of utmost importance to have a carefully chosen set of points through which to sense for such anomalies.

RIPE NCC has started a new initiative for a measurement system called RIPE Atlas that offers a high number of publicly-accessible network vantage points, which can be a favorable substitute for components of currently-existing routing anomaly detection systems. By carefully examining the top existing anomalies and their corresponding detection methods, the project strives to examine whether RIPE Atlas is a good substitute. In order to do so, a number of experiments relating to Internet anomalies will be devised and executed by using RIPE Atlas.

1.0.1 Research questions and approach

The main research question of the project follows:

"Is it possible to detect routing anomalies in the Internet's control plane by relying on traceroute data from RIPE Atlas probes?"

Specifically saying:

"Is it possible to detect filtering, MitM(Man-in-the-Middle) routing attacks, eavesdropping or simply routing policy changes by relying on data-mining of Atlas's historical traceroute archives or by using newly-defined active measurements?"

And,

"What other datasets are needed to complement data obtained from RIPE Atlas in the process of accurately detecting the aforementioned Internet routing anomalies?"

Due to limitations presented by the RIPE Atlas system, a number of other fundamental network test utilities were examined such as ping and DNS queries in addition.

1.0.2 Scope

The scope of the research will first explore the technical capabilities of the RIPE Atlas system. Following, a theoretical study looks at whether currently existing detection system may benefit to utilize RIPE Atlas. As a result of the theoretical study, detection methodologies will be synthesized and adapted to fit RIPE Atlas.

1.0.3 Thesis Outline

The rest of the paper is organized as follows. Section 2 provides a top-level view of the current capabilities of RIPE Atlas and its overall architecture. Section 3 explores the different ways in which routing anomaly detection has been done so far. It then elaborates on how RIPE Atlas can be instrumented as a replacement to certain components of existing detection systems. Section 4 lists the methodology and results of performing detection of the different types of routing anomalies. Section 5 presents the conclusions of the conducted work. Section 6 lists further areas of improvement based for methodologies used in Section 4 and elaborates on what systems would benefit from utilizing RIPE Atlas.

Chapter 2

RIPE Atlas

As a side effect, the growth of the Internet increases its overall complexity. This can potentially introduce more forwarding-plane instabilities. Ideally, packets in the forwarding-plane should be delivered reliably and efficiently through the network, however, in many cases, the network paths may not be perfect due to faulty policies within both Interior Gateway Protcol (IGP) as well as Exterior Gateway Protocol (EGP) environments. Therefore, the study of end-to-end network reachability must always examine the way traffic flows in the data-plane. It is therefore of utmost importance to utilize a geographically diverse set of vantage points in such studies through which data-plane probing can be done.

2.1 System specification

Established in 2010 by RIPE NCC, the Atlas initiative aims to provide a global network of vantage points. Its primary aim is to build the largest, publicly-accessible Internet measurement network. In its early stages, the network was established with the goal of further enforcing quality assurance for resources managed by the RIPE NCC region such as studying reachability and and round-trip-times to root name servers, with initial probe hosts being primarily members of the Internet numbering and research community. Probes were hosted mostly in educational networks, where in many cases, a minimal amount of routing anomalies and filtering occurs. Later, Atlas became available to the public at large and many of the probes are now hosted by residential users with leased Internet connections from Tier 3 ISPs. Naturally, such a measurement network far exceeds the initially foreseen applications.

By using the system, numerous questions regarding data-plane routing and end-to-end reachability can be answered such as:

- How does the Internet of a country/city compare to that of another
- How are packets towards certain resources routed on the Internet when there is a disaster somewhere
- Where exactly is filtering performed for a given resource

RIPE Atlas uses a centralized operational model. Specification, reservation and measurement data collection is done by a central authority. All underlying tasks within the specification of a measurement, such as probe selection, picking a measurement type and defining its underlying properties and execution intervals are done via an HTTP REST API. In addition, RIPE also offers a web-based interface that implements the API. Using a centralized model allows RIPE NCC to scale the network to a large number of probes while ensuring ease of maintenance and usage.

The RIPE Atlas measurement network currently consists of more than 5200 active probes [19]. The globally-routed IP addresses of probes show a dispersion throughout 1980 IPv4 ASNs, 628 IPv6 ASNs [1] and a total coverage of 139 countries. In order to connect a probe to the network, a host may either provide the probe with a globally routed IP address, or place it behind a NAT. Probes do not maintain any open ports, therefore making no contribution towards a network attack surface. As their only network requirement, probes need to be able to establish new, outgoing connections to a number of host addresses and ports part of RIPE NCC's network. From this, it can be inferred, that each probe checks upon all of its state changes, such as starting to work on a new measurement that uses it, by routinely engaging into contacting its controller and retrieving data from it that describes its current state.

2.1.1 Credit system

The RIPE Atlas credit system is used both as an incentive for users to participate, as well as for enforcing boundaries upon the usage patterns of its users. Everyone can use any probe from the network to perform his own measurements, however only probe hosts can generate credits that are needed to actually schedule a measurement. If a user's probe is online and accessible by the RIPE Atlas controller for a full 24 hour period, it generates 21,600 credits. As a second condition that limits the usage of the

[1] RFC6793 - BGP Support for Four-octet AS Number Space

system, a user cannot spend more 175,000 credits in a day. Due to the high number of probes part of the system, it is a necessary limitation to prevent malicious usage such as Distributed Denial of Service (DDoS) attacks.

2.2 Measurements

RIPE Atlas User-Defined Measurements (UDM) are comprised of a collection of tests all of which use the same set of probes. Currently the system supports four test types [21]:

Table 2.1: RIPE Atlas test types

- IPv4/6 Ping
- IPv4/6 Traceroute
- DNS lookup
- SSL GET Cert

Tests are executed by the standard set of tools under Linux systems, namely - ping, traceroute, dig, openssl and curl. All of the aforementioned tools have a rich feature-set that can fine-tune their operation, however not all such options available to their Linux counterpart are present. An additional test type is currently undergoing evaluation and development, which would allow users to perform HTTP GET requests. A detailed specification of options and arguments supported by each test type is provided at the RIPE Atlas User-Defined Measurement Wiki [21].

Measurements costs credits. Two types of measurement can be executed based on their longevity and underlying tests. An One-Off measurement is one which includes a single test type and does not repeat itself. The second, more sophisticated type of measurement, can have multiple test types all of which are executed for extended periods of time at a given interval. Additionally, the properties of underlying test can be adjusted individually with such measurements. Given an One-off measurement, credit composition is is based on (1) the type of test and (2) the amount of packets and packet data generated during a single execution of the test as shown in Table 2.2 . The cost composition omits the length of Layer3 IPv4/6 and ICMP headers and only charges upon the actual count of packets and the size of data portions part of the routed protocols Protocol Data Units (PDU).

Table 2.2: RIPE Atlas One-off measurement cost composition

Measurement cost $\equiv$ *Test cost*, where

Ping

$$Test\ cost = N * int(\frac{S}{1500}) + 1, \text{ and}$$

N = number of packets (default 3)
S = size of ICMP data in packet (default 48 octets)

Traceroute

$$Test\ cost = 10 * N * int(\frac{S}{1500} + 1), \text{ and}$$

N = number of packets (default 3)
S = size of ICMP data in packet (default 40 octets)

RIPE Atlas Traceroute uses the traditional operational mode, where probe packets are UDP datagrams with "unlikely" destination ports in the rage 33434 to 33534

DNS

Each DNS UDP query costs 10 credits
Each DNS TCP query costs 20 credits

Repetitive measurements incur an additional cost. In such cases, the overall cost is a product of the cost of a test and the amount of executions each test has performed during the measurement.

2.2.1 Specification

All measurement types can be specified either via a web-interface or by using an HTTP REST Application programming interface (API). The functionality presented by each method is fully interchangeable. Five important API characteristics describe each measurement, which are the measurement type, number of used probes and the last three relate to timing. The first timing characteristic describes the intervals at which the measurement type is executed. For example, a ping that consists of a single ICMP request packet, may be repeated with different intensities such as one second apart or one hour apart. The second timing characteristic is indicative of the overall length of a measurement. For example, by specifying a measurement length of one hour, and a ping request part of the measurement that has an intensity of one minute, would yield

a measurement which performs pings every minute for the duration of one hour. The third timing characteristics specifies whether the measurement will start immediately or at a certain time in future.

Given the vast number of probes in the system, it is of utmost importance to have a number of fine-grained ways of selecting probes. Since probes are simply network nodes, RIPE Atlas assignms a number of important characteristics to them:

- Probe IP address association - in case a probe resides behind NAT, it would be described by both its internal IP address as well as its globally-routed IP address.
- Probe global IP prefix
- ASN in which the probe resides
- Geographic coordinates

In the RIPE NCC region, Atlas can learn the values of all other major probe attributes that relate to location only by examining its globally-routed IP address. This is made possible by objects part of Internet Routing Registries (RFC2725). In the RIPE NCC region, such registries have a very high degree of accuracy and completeness. A study conducted by RIPE NCC [10] in 2012 found the correctness of its WHOIS database to be over 95%. This is due to the fact that all ISPs and LIRs that receive allocations from blocks that are managed by RIPE NCC undergo a mandatory procedure of not only always registering a prefix to an ASN they own, but also inserting all such mappings in RIPE NCC's WHOIS database either manually or by means of RWHOIS servers. Nevertheless, when a host request a probe, it is a necessary condition to specify geographic coordinates. This ensures that in case the host is located outside of the RIPE NCC region, geographic correctness is maintained.

The final important characteristic relates to the way probes can be selected. One can choose to reserve a certain number of probes based on the following origin attributes: Area, Country, Probes, Autonomous System number, Prefix and Existing UDM. The area attribute defines five global areas that correspond to RIPE Atlas controller regions rather than geographical boundaries. The probes attribute allows selecting probes by their identification number. The Existing UDM attribute allows to pick a set of probes that have been used in a previously executed UDM. However, an even geographical distribution is not achieved when reserving probes based on the area and country attributes. With the former, even though multiple countries are part of the same area all returned probes might be from just one country, and even just from one city. The same applies to the country attribute.

Each probe is represented by the following attributes relating to network and physical location:

- IPv4/6 ASN, address and prefix
- Country code
- Latitude, longitude

It is therefore of utmost importance to actually define scripts through which such limitations are defeated.

2.3 Additional system limitations

A number of additional limitations govern how RIPE Atlas is utilized by its users:

- No more than 100 simultaneous measurements for a single user
- No more than 500 probes may be used per measurement
- No more than 10 traceroute or ping UDMs can exist for a given target URL/IP for all users of the system. The condition does not apply to DNS measurements

Due to its centralized design, RIPE Atlas experiences a degree of offskew between measurement scheduling and reservation. Additionally, the execution intervals of a repetitive tests may vary as well.

The set of requested probes during measurement scheduling is applied to all underlying tests (Table 2.2) and cannot be chosen on a per-test basis. The set of allocated probes may be comprised of less probes than requested, due to reserved probes going offline in the interval between measurement reservation and execution. In such cases, no new probes matching the selection criteria are provided and test are launched with a reduced probe set.

Chapter 3

Related work

Routing anomalies may be related to both the control- and data-plane of the Internet. The chapter provides a theoretical study of routing anomalies as well as technical specifications on why they occur. Furthermore, the chapter outlines what are the advantages of using RIPE Atlas for the detection of examined anomalies. The three main categories of examined routing anomalies are improper filtering of IPv4 subnets, conscious filtering due to Internet censorship and BGP prefix hijacking detection.

3.1 Improper filtering of de-bogonised IPv4 blocks

The term bogon prefix draws its meaning from the word "bogus". It is used to describe prefixes which must never appear in the Internet's routing tables, such as those part of reserved or private blocks (RFCs 5737, 6598, 6761, 6890). In addition, packets routed over the public Internet must never have a source address part of a bogon prefix, as such addresses must be restricted to private networks and if seen outside, they are commonly the source of spam and DDos attacks [9]. It is the explicit responsibility of ISPs to enforce and maintain accurate filtering of bogon prefixes on the boundaries of all Autonomous System Numbers (ASN) they maintain. A list of current aggregated bogon prefixes can be seen in Table 3.1. Bogon Filters must be frequently updated to avoid blocking legitimate traffic. There are several organisations on the Internet that provide daily updated bogon and blacklist filters, the most well-known one being the TEAM CYMRU Community services organisation [18].

IANA is the primary organisation that manages delegations from the global IP and AS number spaces to RIRs. Due to the imminent depletion of the IPv4 space, IANA constantly revises the status of otherwise reserved blocks or subprefixes and delegates

Range	Mask
0.0.0.0	8
10.0.0.0	8
100.64.0.0	10
127.0.0.0	8
169.254.0.0	16
172.16.0.0	12
192.0.0.0	24
192.0.2.0	24
192.168.0.0	16
198.18.0.0	15
198.51.100.0	24
203.0.113.0	24
224.0.0.0	4
240.0.0.0	4

TABLE 3.1: List of current bogon prefixes(aggregated)

them to RIRs for further allocation and assignment. Such ranges are referred to as De-bogonised address ranges. Although once they were part of a bogon list, their bogon status is relinquished and their appearance in Internet routing tables allowed. Essentially, when all off the IPv4 address space has been eventually allocated, network operators might consider fully ceasing bogon filtering and only continue filtering statically private and reserved IPv4 blocks.

In order to guarantee quality assurance to LIRs, APNIC and RIPE NCC, launch de-bogon pilots through which the global Internet reachability to such prefixes is tested prior to making any allocations from them. RIPE NCC is the primary RIR who has been establishing such pilots. The pilots use RIPE NCC's BGP Remote Route Collectors (RRCs) for analysing BGP routing convergence. RRCs are Linux-based software routers that collect default free BGP routing information from a number of key Internet Exchange Points around the world in addition to forming interconnect agreement relationships between RIPE NCC's ASN12654 to all major Tier-1,2 providers [1]. This allows RRCs to also be used for the study of global BGP routing convergence on short-lived (route flap damping [15]) or long-lived (de-bogonised prefixes) prefix announcements. Launching such debogon pilots ensures that:

- The global distribution of routes to the pilot prefixes can be compared against the distribution of regular production prefixes. Noticeable differences can then be further analysed to pinpoint ISPs that are filtering routing announcements from the new block.
- Reachability to prominent resources, such as root name servers from the pilot prefixes (B, H root [17]) can be studied

[1] https://stat.ripe.net/widget/asn-neighbours#w.resource=AS12654

After global BGP convergence is ensured, the second step within the process of debogonising new address blocks is to study it's background radiation [14]. The term is used to describe fundamentally non-productive data-traffic the cause of which is primarily malicious - flooding backscatter, worms such as Conficker, vulnerability scans. As previously the blocks' usage must have only occurred in private networks, it is very likely it has been used by software products and systems of different parties. The process ensures that all subprefixes part of a block that are associated with bad traffic are not distributed for allocation and remain part of bogon and blacklist filters instead. A prominent case was seen in 2010 with the distribution of 14.0.0.0/8, where the Conficker worm is present. APNIC reports suggest that parts of the block are not to be distributed [9]. Debogon pilot programmes end as soon as allocations from the new block are made and real production prefixes announced by the ASN of the ISP who received the allocation.

Debogon pilot programmes have primarily focused on examining the global BGP routing converges of newly-released /8 IPv4 blocks in the Internet's control-plane. Although this process ensures that BGP speakers on a global scale are appropriately leaving debogonised prefixes unfiltered, it does not provide any evidence of actual two-way, uninterrupted data-plane connectivity to them from ASNs different from the one a debogonised prefix is assigned to. In addition, due to the vast size of many ASNs, it is sometimes important to not only test for such connectivity from a single prefix of the ASN to the debogon, but from multiple ones instead. Normally traffic to such prefixes in the source ASNs data-plane would be managed by geographically distinct IGP devices. An example is seen in the way an ISP would assign possible different prefixes to be used in different cities. Two-way, data-plane communication from external ASNs to a city-based prefix would ultimately traverse different IGP devices once entering the ISP ASN. Possibly, even the entry point would be different in case of ASNs with multiple External-BGP (EBGP) speakers. As seen in the interesting case of 128.0.0.0/16 [12], some bogon ranges were filtered by network devices on a non-adjustable software level.

This means that although all EBGP speakers in an ASN are not performing any filtering, some of the IGP devices of that ASN may be doing so in the data-plane. Given the inhomogeneous set of devices normally used in carrier environments, some parts of large ASNs where such devices exist may be doing filtering, while other parts are forwarding traffic normally. In order to address such data-plane anomalies, RIRs have actually provided pingable targets from debogonised prefixes while they were still unallocated and being advertised from their own ASNs. However, this approach requires that the everyone in Internet community at large is to be informed about it and issue tests from his/her ASN and prefix to the provided pingable targets. At current, the RIPE Atlas system is a perfect testbed for reversing the process and eliminating the need

for conscious user participation. This is seen in the fact that the system offers a high number of vantage points situated in residential locations as well as an even spread of probes across many ASNs.

3.2 Internet censorship

The pervasive usage of the Internet as an instrument for disseminating information and fostering the formation of communities of any kind is steadily giving rise to robust Internet censorship policies. The initial ways of performing censorship use several approaches to prevent access to resources on the Internet by using technologies at different network layers (application or network and transport) throughout different points such as at a user's machine, at the ISP level or at the endpoint resources themselves. At current, the most pervasive way of establishing censorship is on the ISP level. From a technical standpoint, censorship is established by using a combination of IP blocking, DNS filtering and redirection or URL blocking with a proxy. Newer approaches employ automatic keyword blocking, which blocks access to websites based on the words found in its URLs, page bodies or by examining search engine queries for blacklisted terms that associate to its URLs via any of the aforementioned methods. An ever-increasing number of regional authorities are using this approach, albeit the vast number of false-positives generated by it.

Internet censorship is also a topic of active research. Detailed public information on regional filtering can be obtained from sources such as the OpenNet Initiative, Herdict and Google's Transparency Report, amongst others. Albeit the growing number of public sources on the topic, all detection systems lack the fundamental diversity of vantage points that is needed for swift and truly global detection. Most public services utilize a set of vantage points which are evenly spread across the globe, however such a distribution cannot successfully sense censorship for all macro regions such as districts and/or cities on a global scale. As Internet censorship is primarily established in Tier 3 networks, it is important to have an even coverage of Autonomous Systems and unique prefixes instead. Having vantage points in as many ASNs as possible, guarantees that the detection system essentially can sense for censorship from the perspective of the general public at large. This is due to the fact that ASNs may not be assigned to more than one ISP, and therefore adequately covering a geographic region such as a country requires having vantage points in every ASN of every ISP in the country. In addition, covering as many unique prefixes from an ASN as possible, provides an ever more precise way of pinpointing Internet censorship on a district and/or city level. It also ensures that larger ASNs that span multiple countries are adequately studied. A large-scale research

that employs such a model is the User-Based Internet Censorship Analysis (UBICA) [8] research project funded by Google's Faculty Award 2013. The research project strives to define a censorship detection model that uses either modified firmwares in residential routers or access points in addition to studying the usage of client-based applications. The project encompasses the distribution of either hardware components or software applications on a large scale. This can be a slow, expensive and tedious process. As fundamentally the detection of censorship requires either Layer 3 probing via traceroute and ping to check for IP blocking, or a combination of application-layer DNS and HTTP queries to check for DNS filtering or URL blocking, RIPE Atlas probes can be used as a substitute for vantage points in the project. With a well-established probe base, RIPE Atlas is a cost-effective substitute, which also provides high detection efficiency with its even distribution, especially in the RIPE NCC region.

3.3 MitM routing attacks

The term Man-in-the-Middle routing attack is used to describe a special case of BGP prefix hijacking events. BGP is the standard Inter-Domain routing protocol of the Internet that connects all Autonomous Systems falling within different administrative domains. BGP is a path vector protocol and as such relies on the AS_PATH attribute for disseminating paths to destination address prefixes to EBGP peers. A destination prefix should be usually announced either by the prefix owner itself if it participates in BGP and has an AS number, or by its upstream provider ASN. Two key factors make prefix hijacking possible. First, the initial specifications of the BGP protocol, starting with RFC1771, makes a number of unjustified assumptions:

- Each AS announces only those prefixes for which it has clear ownership
- Source of BGP update has authority to announce the prefix
- Announced AS paths are always correct
- TCP provides a secure transmission between BGP peers

Various improvements and standardizations have addressed each of the issues, however in practice it is impossible for a BGP speaker to be secured from all of them. The primary reason for this is that most of the improvements require an extensive increase in computational capacity for all EBGP devices in an ASN.

A number of these improvements are:

- Signing BGP updates (S-BGP)
- Verifying the AS_PATH attribute (SO-BGP)
- Securing TCP sessions with IPSEC or MD5 validation
- Per-ASN filters to ensure one's neighbours only announce their own space

The second factor that makes prefix hijacking possible is stale and incorrect data in Internet Routing Registry (IRR) databases. In the RIPE NCC region, this is no longer the case due to additional conditions through which WHOIS objects in the region are maintained up to date [10].

3.3.1 Analysis of prefix hijacking

The main way IP prefix hijacking is carried out can be seen in Fig. 3.1. The figure shows an unconverged scenario where ASN1 has just joined BGP and has started advertising prefix p. ASN2 has a provide-customer relationship with ASN1. Likewise ASN3 has a provider-customer relationship with ASNs 2 and 4. ASN5 is the supposable malicious peer which has a peer-peer relationship with ASN3. Additionally, ASN5 is a second upstream provider of ASN4.

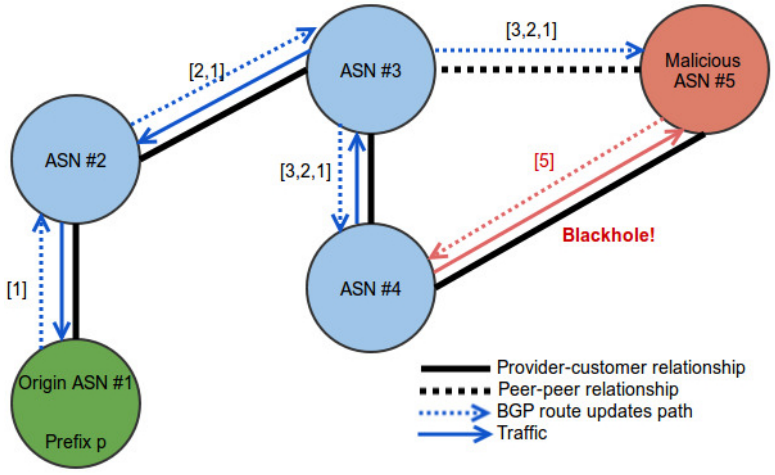

FIGURE 3.1: Prefix hijacking methodology overview

ASN1 advertises its route to prefix p to the rest of the world by using its provider ASN2. This corresponds to an AS_PATH update generated by ASN1 that simply contains [1] as ASN1 is the owner of prefix p. ASN2 propagates this route to its only upstream

provider ASN3, by pre-pending itself in the AS_PATH and sending an update of [2,1]. Consequently, ASN3 does the same and sends an update of [3,2,1] for prefix p to ASN 4 and 5. At this sage, every ASN in the topology knows how to reach prefix p. Given the case with ASN4 which has two providers, it should always choose to reach the prefix via the ASN3 path [3,2,1] rather than the [5,3,2,1] path via ASN5. However, if none of the guidelines outlined in the previous section are in place, ASN5 can start advertising rogue updates that list itself as the only ASN to prefix p with a corresponding AS_PATH update of [5]. Given that ASN4 also does not implement any of the aforementioned security features, it may prefer to send traffic to prefix p via ASN5 as a result of the shorter AS_PATH updates heard from ASN5. This ultimately creates a situation in which traffic originating from ASN4 and destined for prefix p encounters a blackhole due to prefix hijacking.

A number of important observations stem from the operational characteristics of prefix hijacking. The influence of hijacking cannot exceeded beyond the subtree of ASNs to which an attacker ASN connects to. Additionally, this subtree includes all downstream ASNs of the initially affected ASNs. Moreover, two general attack types exist depending on the positioning of the attacker ASN with regards to the attacked ASN. Fig. 3.1 outlines a case where hijacking as an Invalid Origin [2] is conducted by announcing itself to be at the origin of the AS_PATH updates for prefix p. It is also possible for the attacker to announce himself as a transit ASN towards a prefix p, or an Invalid Transit. This case is easily seen in Fig. 3.1 if ASN5 did not have a peering relationship with ASN3 that provides it with a valid route to prefix p, but still made advertisements for the prefix to ASN4.

3.3.2 Classification of prefix hijacking

In order to better understand whether RIPE Atlas can be a substitute for the detection of prefix hijacking, first a taxonomy is provided that classifies the different types and countermeasures.

A prefix should always originate from a single AS on the Internet (RFC1930). Various conditions exist because of which this is not always the case, such as Internet Exchange Points Exchange Points (IXP) and multi-homing via multiple providers without BGP or with private ASN numbers. When a prefix originates from multiple ASNs, Multiple-Origin Autonomous System (MOAS) [2] conflicts start to occur. Additionally, in case a more-specific part of a given prefix is being advertised from a second ASN, subMOAS

conflicts occur. Given, the two types of conflicts based on prefix length and the positioning of the attacker ASN advertising a prefix or a subprefix, the following taxonomy is made:

- Regular prefix hijacking - cases where exactly the same prefix p owned by ASN V is also advertised by ASN X with invalid route advertisements that claim it to be either the owner of the prefix (Invalid Origin) or one of the transit ASNs to the prefix (Invalid Transit)
  - Invalid Origin - the type of attack leads a MOAS conflict seen by all BGP Routing Information System (RIS) peers and BGP anomaly detection systems that stem from them
  - Invalid Transit – the type of attack does not lead to a MOAS conflict, because the attacker ASN does not seem to violate the single origin rule. Additionally, it's not a favourable path due to its increased length, however attackers can hide other hops from the path to make it appear shorter. In such cases, detection must always rely on control-plane passive monitoring
- Sub-prefix hijacking - cases where attacker X starts to advertise prefix p of ASN V claiming to be either the owner of the prefix (Invalid Origin) or one of the transit ASNs to the prefix (Invalid Transit)
  - Invalid Origin - the type of attack can be overlooked by ordinary MOAS-based hijack detection mechanism. [2] states that MOAS conflict would occur unless its super prefixes are examined in the detection process
  - Invalid Transit - the type of attack will not introduce a MOAS or subMOAS conflict because of the longest prefix match rules used in BGP in addition to executing the attack as a transit to the prefix p, hence it is the most difficult attack to detect

In addition to the aforementioned types of prefix hijacking, a new type of attack has been repeatedly reported. After successfully hijacking a prefix p, an attacker can forward hijacked traffic back to the original ASN by means of VPN tunnelling or any other way that maintains valid routes to the original ASN. Such a situation would not disrupt data-plane connectivity in-between third parties part of the affected ASN subtree communicating with hosts from p hence making the interception invisible to them. This type of attack allows for traffic to be eavesdropped, inspected and modified therefore leading to a MitM routing attack. Such a scenario is seen in Fig. 3.1 if ASN5 forwards traffic back to ASN1 instead of forming a blackhole. Depending on the way a valid route towards the original destination is maintained and whether the rogue announcement

is made as an Invalid Transit or Origin, the attack may be executed in a number of different ways [4]. Additionally, various mechanisms may be employed that completely mask the increased data-plane round-trip-times and hop counts.

3.3.3 Prefix hijacking detection

Prefix hijack detection systems are separated in two categories. First, control-plane systems such as Cyclops [5] and PHAS [6] are used to monitor global BGP announcements and seek out MOAS and subMOAS conflicts. They utilize RIPE NCCs BGP Remote Route Collectors (RRC) in addition to various other BGP devices at key locations such as IXPs that have peering relationships with many Tier 1 providers. Three important implications exist with sole control-plane detection. The first two are false-positives stemming from valid MOAS and subMOAS conflicts and false-negatives due to the fundamental difference between BGP AS-level paths and the underlying data-plane paths. Users of such systems need to be the tiebreakers in case of false-positives and determine for themselves whether the MOAS or subMOAS conflict is malicious. Lastly, depending on the BGP feeds used by the system, latest BGP feed data may be obtained hours after a MOAS conflict has occurred.

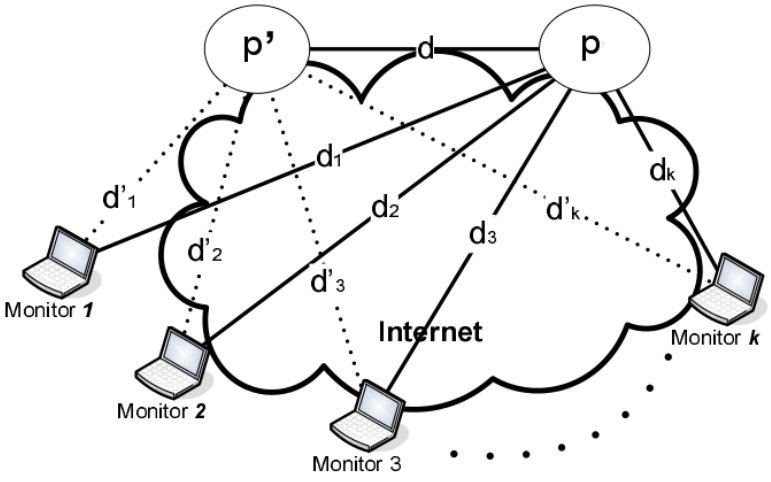

FIGURE 3.2: Hijacked prefix location chage

The second type of detection systems employ data-plane probing [1, 3]. By using a two-step detection approach, they eliminate both false-positives as well as false-negatives stemming from MOAS and subMOAS conflicts. Given that prefixes belong to unique ASNs and normally do not change their location, the first step is to monitor the network location of a prefix (Fig. 3.2). By doing so from a set number of vantage points, concise network distances can be calculated from these points to a given prefix p. However, the target prefix needs to be monitored for an extended period of time so that the

normal path changes in the data-plane can be accounted for when devising distances. Should prefix p be hijacked, a number of the monitoring points would exhibit significant changes between their previous distance to it and the new underlying network path. This condition will apply as long as some of the vantage points are part of the subtree of ASNs that is hijacked. The second condition in data-plane detection schemes, is used to reduce false-negatives that are the result of legitimate network path changes such as failing network nodes. By monitoring a prefix q, which is part of the same ASN as prefix p, path disagreements in detected distances are eliminated.

Robust prefix hijacking detection methodologies must always employ a detection approach that relies on both control- and data-plane analysis [7]. Employing a single approach suffers shortcomings such as invalid and prolonged detection. In the case of data-plane probing, it is also important to actually start the detection prior to an attack being executed in order to detect distance changes. The various limitations have pushed robust prefix hijacking experiments to employ both approaches in addition to historical analysis of IRR databases and relating results to reported spam prefixes.

Given that data-plane probing must rely on a carefully chosen set of vantage points, so that some of them are in the subtree of ASNs polluted by malicious BGP updates, RIPE Atlas is a favourable network of vantage points to be used. Additionally, it presents an environment, which can adapt and change the used vantage points, so that different subtrees of the global ASN structure are examined for occurrences of location changes and path disagreements given a certain target prefix. At current, data-plane prefix detection frameworks [6] use probing locations part of the PlanetLab testbed.

Chapter 4

Experimental study of routing anomalies with RIPE Atlas

The chapter presents the main experiments that were conducted with RIPE Atlas. The chapter is organised as follows. First, each experiment is preceded by a further enforcement of conclusions made in Section 3. Next the methodology through which an experiment is conducted is analysed. Finally, results are discussed for each experiment.

4.1 Debogon filtering

As outlined in Section 3.1, bogon prefixes must not appear on the Internet. It is the responsibility of LIRs to enforce filtering for such prefixes appearing in BGP announcements. The conducted tests for detecting improper filtering of de-bogonised prefix ranges use a data-plane probing approach bypassing the need for examining BGP data. Conducting the test by relying on data-plane probing alone ensures that both BGP dissemination is not hindered by filtering for a given ASN, as well as that all IGP routing in the ASN is sufficient for end-to-end connectivity. For larger ASNs, accurate data-plane detection should utilize multiple probes with distinct prefixes rather than just a single probe from it. Therefore, a favourable condition in choosing the set of Atlas probes from which to test is to simply use as many of them as are online. That implies that a single ASN would be represented by all of its available probes. However, this would also include probe that share a prefix, which in most cases implies they are sharing either the exact same physical location in the case of residential users or having a long common subpath part of their origin ASN. Therefore, the set of used probes in all debogon experiments does not include multiple probes from the same prefix. In addition, the set excludes probes that belong to the same ASN a tested debogon prefix is registered to

as well as all of the ASNs immediate peers. The exact number of probes used in each test varies due to filtering rules and because the separate debogon tests were conducted in successive order and some of the probes from one test might have been offline when devising the set of probes for another test. These numbers are given on a per-test basis in each section that follows.

The used data-plane probing approach can utilized both ping and traceroute. However, the use of traceroute was not possible due to credit limitations presented by RIPE Atlas and all experiments use ping as their base test. Traceroute is only used as part of verifying results that indicate improper debogon filtering.

All tested debogon prefixes are part of de-bogonised /8 blocks that were the last delegations APNIC and RIPE NCC have received from IANA in 2011 and 2010 [11, 14]. At current, hundreds of more-specific prefixes in the range of /15 to /24 have been allocated to various LIRs. Depending on the chosen subprefix towards which a test is conducted, a limitation exist. Each such test is indicative of two-way reachability of only particular sets of more-specific masks and not the entire /8 block. As an improvement to this, a number of different subprefixes were chosen for some of the /8 blocks part of their lower and upper boundaries.

Testing a prefix from a de-bogonised block for reachability utilizes a chosen set of probes by defining two different pings test from each as seen in Fig. 4.1. First, each probe performs pings with a predetermined intensity towards a host address from the de-bogonised prefix. This serves as the primary detection mechanism of filtering. However, by just issuing a ping towards the de-bogonised prefix, it is not possible to state with certainty that the unreachability is the result of improper debogon filtering. It can also be the result of suboptimal global BGP convergence with regards to prefixes disseminated by the destination ASN. This is a common scenario of human mis-configuration on the BGP level. In addition, it could also be the result of improper IGP routing policies. Therefore, each probe used within a test also issues a second ping, with a lower intensity, towards another prefix that is part of the destination ASN and also shares the debogon's geographical destination. Ensuring the geographic proximity for the chosen reference points was further enforced by performing traceroute tests towards the pairs of de-bogon and reference hosts.

Pings towards hosts in the debogon ranges all share an intensity interval of 20 minutes, whereas pings towards the reference points are scheduled 60 minutes apart. The difference in the intensities allows to further test for what is the continuous connectivity in-between a pair of source and destination prefixes. However, this is considered future work and further described in Section 6.

Choosing the pingable targets for each experiment was done by discovering DNS servers. The rationale for choosing DNS servers was based on the fact that DNS uses UDP for its underlying transmission protocol with server instances traditionally always running on port 53. This ensures that, for the very least, UDP communication towards port 53 of a given DNS server is always allowed. It does not, however, guarantee that UDP datagrams can be sent to any other ports. A necessary condition in the selection process is that traceroute towards each chosen host is possible, so that in case of detecting improper debogon filtering, traceroute can be conducted during result verification. As both ping and traceroute as offered by RIPE Atlas, use the traditional ICMP model with "unlikely" ports in the range 33434-33534, it was manually ensured that ping and traceroute are possible towards all chosen hosts. Therefore, all pinglable targets were discovered manually by checking what are the authoritative name servers for given reverse *in-addr.arpa* pointers and retrieving their IPs.

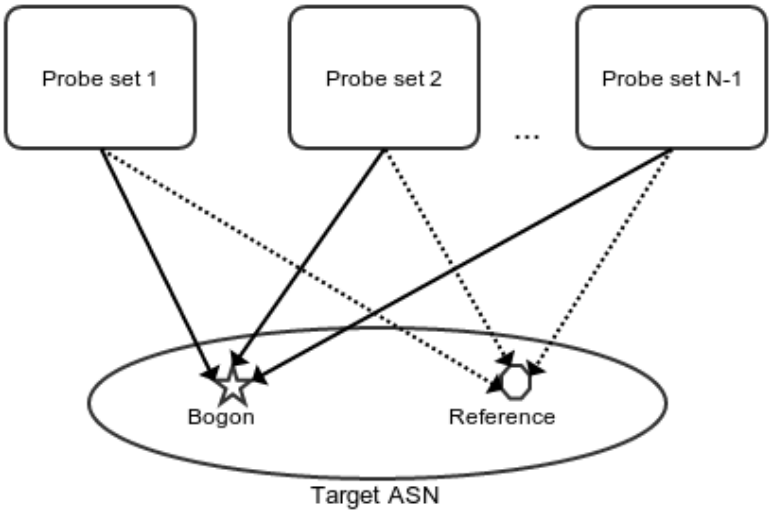

FIGURE 4.1: Conceptual overview of how a de-bogonised block is tested for filtering with regards to a single subprefix from it.

Each debogon test was done with a duration of 12 hours. This ensures that short-lived network outages were not interfering with the results and generating false-positives. In case a single ping was issued towards the two targets of each debogon tests, then many of the results might have indicated both targets as unreachable due to the presence of intermittent network outages. Taking the aforementioned in mind, the condition for detection of filtering should be readjusted. Cases in which a probe is capable of reaching a reference point, i.e. it has received a ping reply for at least one ping request, however incapable of reaching the host from the debogon for the entire duration of a test(12 hours) are confirmations of improper debogon filtering.

4.1.1 Obtained dataset

RIPE Atlas presents the limitation that a single UDM cannot use more than 500 probes. At the time of scheduling each debogon experiment, there were approximately 3200 probes that matched all required probe selection filters. In total, six sets of 500 probes and one set of 200 probes were used in each experiment. Each probe set issues pings to two predetermined destinations. Therefore, the generated results are described by two pairs of seven sets of pings tests for each destination. The result of each measurement is represented as a JSON object in the RIPE Atlas system, which can be retrieved via the API. Each underlying ping test part of a measurement has the following structure:


```
{
  "af":4,
  "avg":79.102666666666664,
  "dst_addr":"185.24.0.1",
  "dst_name":"185.24.0.1",
  "dup":0,
  "from":"193.19.124.139",
  "fw":4580,
  "group_id":1423636,
  "lts":17,
  "max":79.396000000000001,
  "min":78.802000000000007,
  "msm_id":1423636,
  "msm_name":"Ping",
  "prb_id":10087,
  "proto":"ICMP",
  "rcvd":3,
  "result":[
             {"rtt":79.109999999999999},
             {"rtt":78.802000000000007},
             {"rtt":79.396000000000001}
           ],
  "sent":3,
  "size":48,
  "step":1320,
  "timestamp":1391144459,
  "type":"ping"
}
```


Given that the ping measurements use certain intensity, time slices can be defined for the different parts of each measurements result. Pings directed towards de-bogonised prefixes are executed every 20 minutes from each probe set for a duration of 12 hours. Therefore, it can be said that a time slice in this measurement is 20 minutes and that there are 36 different time slices. For the control ping, conducted every 60 minutes, the amount of slices is 12. A time slice contains one ping test for each probe of every probe set.

Given the chronological ordering of each experiment's results, a regrouping was performed by using probe IDs, ping test timestamps and time-slices. On a probe ID basis, the ping tests of a probe were extracted for all time-slices and grouped under its ID. In addition, by using the timestamp part of a ping tests result, chronological ordering was maintained. Due to offskew in the scheduling of ping tests from the same probe, and prior to grouping all pings under a common probe ID, timestamp readjustment was performed on the basis that an offskew higher than +/-10ms was not observed in the results by performing standard deviation checks.

4.1.2 Reachability conditions

Two important attributes of each ping test define its reachability condition, namely the received ICMP duplicates and round-trip-times in the result attribute. The length of the result attribute should always be three, as three ping requests are issued as part of a ping test. Having a longer length is a condition that can only exist when duplicates have occurred. The duplicate attribute is used to describe a number of network conditions, such as more than one host with the same IP address or a misbehaving NAT node. Additional conditions that may produce duplicate ICMP relies exist, however they are not relevant within the context of the executed measurements. Some of these are, pinging a broadcast address, a node which has multiple non-routed return paths or one which performs NIC bonding.

Although receiving a duplicate as part of a ping test can be indicative of two-way reachability, such results are not considered, because determining this requires additional steps to verify the exact condition that is generating the duplicate. This verification process needs to, at best, perform a separate ping test towards each hop along the path to a destination that yielded the duplicate in the first place. All such ping tests need to be scheduled at approximately the same time as the original ping, so that any unknown network condition that may be the cause of the duplicate is captured. However, this is not possible due to the fact that there is no knowledge of the precise hops a ping has taken, as well as because ping tests are evaluated for duplicates only after an overall debogon measurement has finished executing. Therefore, each ping test evaluation regards the presence of duplicate ICMP replies as a case of an unsuccessful ping.

Apart from duplicate ICMP replies, two more cases exist, namely Type 0 and Type 11 ping replies. Type 0 replies are indicative of two-way reachability and are expressed as having a round-trip-time field part of its result attribute. This is the only positive outcome that is considered to signal reachability in-between a probe and its particular

destination. The second one, is a ping request for which the reply has exceeded the specified maximum reception time. In all tests, 3000ms is maximum round-trip-time. This signifies a failed ping and its marked in retrieved objects as having a star in in its result attribute.

Given a particular source probe, various network conditions may influence its capability of reaching a certain destination. As the experiments strive to capture such conditions only if they are related to globally-routed network nodes, all results that have ICMP error messages (Type 3 reply) are filtered, because it cannot be appropriately determined whether they are the result of failures in a probes private network or occurring along a pings globally-routed hops without issuing traceroutes to each such hop. Therefore, the occurrence of all Type 3 error codes are removed from the dataset.

4.1.3 Targets

The following table provides an overview of all tested prefixes. Both debogon prefixes as well as their references are from the same ASN.

TABLE 4.1: Debogon blocks and selected underlying sub-prefixes and reference points

	Debogon		
Block	**Prefix**	**Prefix source ASN**	**Reference prefix**
103.0.0.0/8	103.1.0.0/22	12654 - RIPE NCC	84.205.83.0/24
	103.23.28.0/24	18229 - CtrlS Datacenters	202.65.145.0/24
	103.247.191.0/24	24282 - KAGOYA JAPAN	203.142.192.0/20
128.0.0.0/8	128.0.144.0/21	29066 - VELIANET	85.195.64.0/18
185.0.0.0/8	185.2.136.0/22	13213 - UK2NET	83.170.64.0/18
	185.24.0.0/22	21268 - IP Fjarskipti	46.239.192.0/18

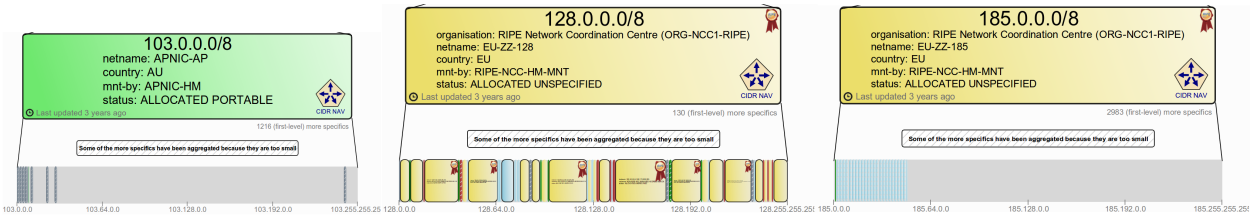

FIGURE 4.2: Currently allocated subprefixes from each block [20]

In April 2011, APNIC reached its final /8 block of IPv4 addresses, which is 103.0.0.0/8. The prefix has not officially underwent any de-bogonising [14] tests by using data-plane probing, however APNIC R&D has deemed it as fully allocatable after a BGP debogon pilot using the now historical APNIC Debogon Project ASN9838.

The 128.0.0.0/16 block is special due to being the lowest Class B address when CIDR and classless IP address space were not used. In 2002, RFC3330 revised its status and

made is subject to allocation by RIRs. RIPE NCC conducted a BGP debogon pilot for the prefix in 2011 by using their own ASN for advertising the /16 prefix as well as various other more-specific prefixes. After the allocations from the block were made, some of the receiving parties discovered that Juniper devices running the latest JUNOS 11.1 at the time and lower were automatically filtering all subprefixes from the 128.0.0.0/8 block. The block has also underwent data-plane probing tests via RIPE Atlas, the last one of which is in 2012 [13]. The study showed that less than 5% of probes were incapable of reaching the debogon out of 1500 used.

The 185.0.0.0/8 block is RIPE NCC's last received IPv4 block by IANA in 2011. The prefix has underwent a BGP debogon pilot, however no data-plane probing via RIPE Atlas.

As of now, all block have less-specific prefixes from them allocated as Provider-Independent or Provider-Aggregateable addresses. Choosing the debogon prefix and reference hosts relied on using DNS to determine pingable targets. The 128.0.144.0/21 and 103.1.0.0/22 are notable exceptions with the former prefix still having ping hosts available for data-plane de-bogonising and the latter still being advertised from RIPE NCC's ASN. Therefore, for the 128.0.0.0/8 block, the used targets were the ones used during the last data-plane connectivity test set up by RIPE NCC in cooperation with the German LIR CtrlS Datacenters. Unfortunately, only one of the set up debogon targets was still available, namely *ripe-test.my-wire.de*.

4.1.4 Results

The following section presents the main results of debogon experiments. Each subsection provides precise details on:

- The examined de-bogonsised prefixes
- The overall number of scheduled probes after RIPE Atlas measurement reservation
- The number of probes removed due to being incapable of reaching both targets
- The Atlas probe IDs in cases where improper debogon filtering was detected
- Plots of the number of probes reaching a debogon prefix in each time slice. The results exclude the few cases of detected debogon filtering as well as probes that could not reach the targets for all time slices

For each plot in this Section, a number of important characteristics need to be known:

- All debogon experiments were carried out on a consecutive, per-prefix basis for a duration of 12-13 hours, and their time intervals and steps normalized before being plotted
- All debogon experiments were reserved with a request for 3300 probes, however all scheduled probe set were smaller. As a result, the starting values of plot entities may significantly differ
- RIPE Atlas presents an offskew in measurement reservation and scheduling (Section 2.3). Hence, even after normalisation, some of the plot entries may either start earlier or end earlier. Additionally, as is the case with 103.1.0.0/22 in Fig. 4.3, some of the probes in a measurement might finish executing their scheduled routine before others from the same test
- All ping test have a hard timeout limit of 3000ms. This causes the plots to be uneven and show large spikes and plunges (approximately 30-70 probes) at each measurements subtest execution. Given that probe sets are build up from an international pool, the timeout value is too restrictive and can make results look peculiar. For example, reference pings and/or bogon pings may show fewer successful executions as opposed to their counterpart in a given time-slice albeit being physically and logically close to one another

As stated in Section 4.1.3, the process of detecting debogon filtering relies on the following conditions:

- At least three ICMP requests have been answered by the reference point for the duration of the entire measurement
- No ICMP requests were answered by the debogon host for the duration of the entire measurement

4.1.4.1 103.0.0.0/8

TABLE 4.2: Probes used in test and detected improper debogon filtering

	Probe(s)		
Measurement	**№ used**	**№ removed**	**Detected filtering**
103.1.0.0/22	3219	28	156, 12007
103.23.28.0/24	3275	26	156, 12007
103.247.191.0/24	3279	26	156, 12007

Probe ID	**Source ASN**	**Last hop IP**	**Last hop ASN**
156	51127	178.255.0.33	51127
12007	45050	Local LAN	45050

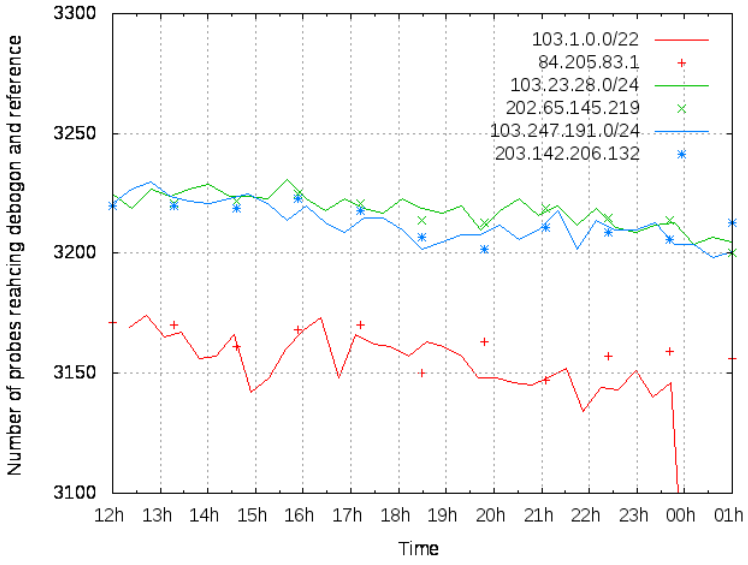

FIGURE 4.3: Total probes capable of reaching debogon and its reference in each time slice

Table 4.2 shown an overview of tested prefixes from the 103.0.0.0/8 block. Every measurement reservation was performed with a request for 3300 probes, however in each case a smaller set of variable size was returned. Probes 156 and 12007 from ASNs 51127 and 45050 respectively were identified as capable of reaching the reference point for all three prefixes throughout all time intervals of the measurements, however incapable of reaching the host from the bogon prefix. In order to verify that the reachability failure is not due to network failures rather than filtering, UDP traceroutes were scheduled towards the three destinations from the two probes for a period of six hours. It could be seen that the traceroutes were failing in both cases at the first hop, which is indicative of faulty firewall policies within the probe's network of residence. However, only probe 156 had an IP address from a reserved class C range. Probe 12007 was using a globally routed public IP address from an assigned Provider Aggregateable address block. It could also be seen in traceroutes from the same probe towards the reference point that

the immediate next hop is part of the same routed PA address block, therefore it is unlikely that faulty firewall rules in-between had applied any filtering. Unfortunately, both of the probes were the only ones from their ASN and no further examination could be made from each of the ASNs. Although the gathered results are not indicative of the entire ASN 45050, it can be concluded that hosts residing in its 178.255.0.0/21 prefix (which incidentally is the only one), are experiencing reachability issues to the bogon in specific while being capable of reaching its reference.

4.1.4.2 128.0.0.0/8

TABLE 4.3: Probes used in test and detected improper debogon filtering

	Probe(s)		
Measurement	**№ used**	**№ removed**	**Detected filtering**
128.0.144.0/21	3214	101	None

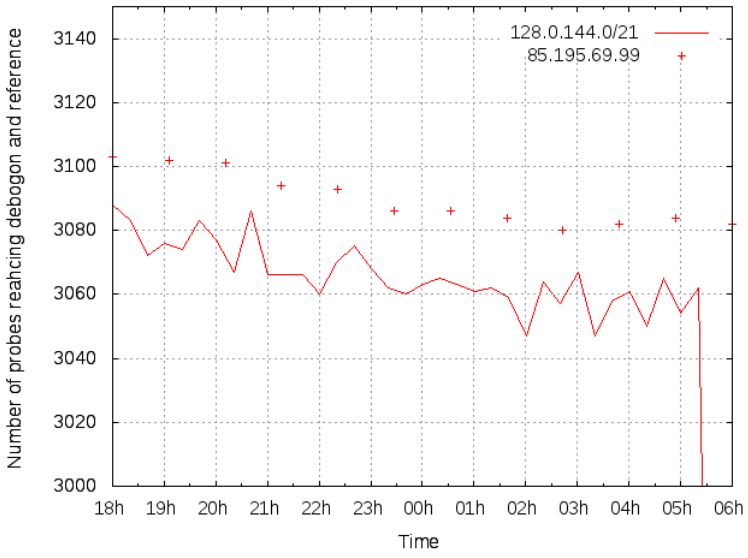

FIGURE 4.4: Total probes capable of reaching debogon in each time slice

No filtering was detected for the chosen prefix from the 128.0.0.0/8 block. As there were significant connectivity issues with assignments from the block, numerous quality assurance tests were conducted by RIPE NCC via RIPE Atlas and via their BGP Debogon programme [11]. The last such test used approximately half the number of probes and found that around 5% were incapable of reaching many of the subprefixes from the block [13]. Although no comparison is done on how the diversity of the probe set changed from the conducted RIPE NCC test to this one, a positive change is seen in the results. However, this test includes the highest number of probes that were incapable of reaching either target. This may imply that besides bogon filtering along the path,

there also may be other problems close to the destination ASN that are responsible for the high count. In Figure 4.4 the high discrepancy between the initial count of probes used for pinging the bogon and the reference is due to the two ping test being scheduled separately and allocated probe sets having different sizes.

4.1.4.3 185.0.0.0/8

TABLE 4.4: Probes used in test and detected improper debogon filtering

	Probe(s)		
Measurement	**№ used**	**№ removed**	**Detected filtering**
185.2.136.0/22	3242	27	156, 12007
185.24.0.0/22	3194	28	156, 3892, 4532, 12007

Probe ID	Source ASN	Last hop IP	Last hop ASN
156	51127	178.255.0.33	51127
3892	50473	31.15.115.5	56704
4532	2818	31.15.115.5	56704
12007	45050	Local LAN	45050

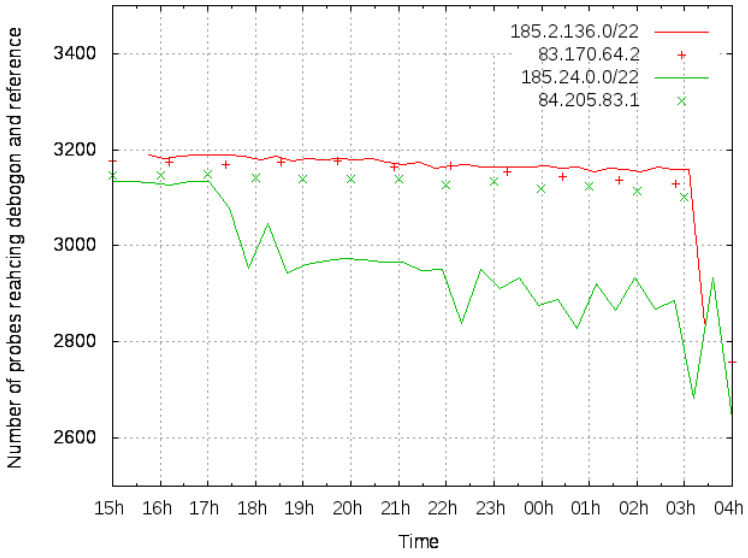

FIGURE 4.5: Total probes capable of reaching debogon in each time slice

The most severe results were seen for the 185/16 block. Besides the singular probes from ASNs 51127 and 45050, two other probes were detected to be along paths where debogon filtering occurs. The two new probes belong to different ASNs, however in both cases the traceroutes used for verification were having as a last hop the same gateway from ASN56704. Unfortunately, no probes from RIPE Atlas reside in that ASN.

Probe 4532 is the only RIPE Atlas probe from ASN2818 (BBC Internet Services UK) and therefore no further tests were conducted with it.

Probe 3892 is part of ASN50473 (Altagen CJSC RU) and has a globally-routed IP address in the 87.236.20.0/22 prefix. Furthermore, three probes from RIPE Atlas reside in this ASN, with two sharing the previous /22 prefix and a third one, probe 4308, which has a unique prefix of 46.151.152.0/21. After scheduling an additional measurement from the two probes with unique prefixes towards the 185.24.0.0/22 prefix, it was established that although probe 3892 cannot reach the debogon host, probe 4308 successfully receives ICMP replies. This is in line with previous results as probe 4308 was part of the measurements initial probe set(after checking). In both cases, as per the detection criteria, the probes can reach the reference points. This situation, coupled with the fact that probes 3892, 4532 can reach one subprefix from the 185.0.0.0/8 block but not the other, clearly expresses the paradigm that examining BGP routing converges alone is not enough to guarantee end-to-end connectivity.

Notably, the common hop at which traceroutes from probes 3892 and 4532 fail is not along the path traffic towards 185.2.136.0/22 get routed through.

The largely failing pings towards the end of the experiment as seen in Fig. 4.5 are due to cases where targets have started to drop a large amount of the ICMP requests for the high-intensity bogon pings. This starts to occur after the sixth round (e.g. the second hour) of pings. As no explicit permission was granted to use the hosts from debogon ranges nor their references as ping targets, but only best-practice guidelines for discovering ICMP-enabled hosts were used, the high amount of dropped ICMP requests is an anticipated occurrence.

4.1.5 Summary

The debogon experiments presented a number of interesting results. Four RIPE Atlas probes were detected to be along paths where improper debogon filtering occurs late after the debogon status of the prefixes has been relinquished and ranges from them allocated to LIRs. As shown in the 103.0.0.0/8 and 185.0.0.0/8 experiments, such filtering may occur anywhere along the path from a source to a destination.

All results were verified with additional ping test lasting from two to three days and executed at one hour intervals. Supporting traceroutes from all four probes have shown that last-hop ASNs at which filtering occurs were not too large and consumer-oriented [1].

The RIPE Atlas system establishes a complicated process of relating the results of measurements. In addition, limitations on the amount of probes a measurement can reserve almost always results in a single experiment being split up in multiple measurements.

[1] https://stat.ripe.net/56704#tabId=at-a-glance

Therefore, when analysing results, it is important to only consider the union of probes reserved within the different sub-tests of a measurement.

The composite credit cost of the multiple measurements a single debogon experiment executes is 1660750 credits.

4.2 Internet censorship detection

As outlined in Section 3.2, Internet censorship may be carried out in different ways along the path from users to endpoint resources. However, due to being regulated by governments and private organisations, it is mostly the responsibility of ISPs to enforce such demands. Technically, censors are established by using IP blocking, DNS filtering and redirection or URL blocking with a proxy. In the case of IP blocking, Internet users that query their ISPs name server get the real IP address of a webresource, however Layer 3 end-to-end reachability is not possible due to their packets being sent to a black-hole destination managed by the ISP. The next two cases, related to either completely blocking DNS queries that request censored URLs or to providing IP addresses in DNS results that do not correspond to the real IP address of a resource.

Therefore, the complete detection of Internet censorship needs to rely on three detection criteria:

- Probing via tools such as traceroute or ping in the case of IP blocking
- Probing via tool such as curl and wget for retrieving webpage headers in the case of redirection. In this case a real IP is served and the client thinks its talking to the real resource, however packets are rewritten en route to be delivered to another Layer 3 IP address
- Probing via DNS queries for the remaining two cases

As of this writing, complete Internet censorship detection via RIPE Atlas is not possible due to the lack of HTTP tools. However, in the near future the system will introduce a new measurement type for the purpose.

4.2.1 Methodology

As in the case of debogon filtering, a favourable condition in the vantage point selection process is to have as many points as possible. Therefore, the procedure via which probe sets are build relies on obtaining all probes from unique prefixes present in the Atlas network. Namely, such a set would include many probe from the same ASN. This ensures that censors for larger ASNs, parts of which fall under different government regulations, are concisely captured on a geographic scale.

All tests were schedule as One-Off traceroute tests, although continuous probing would ensure that conditions such as network failures and name server failures are not part of the resulting dataset.

Each traceroute is scheduled with a URL as a destination rather than a host address. As on normal UNIX systems, the traceroute tool resolves the destination URL by using the naming facilities defined in the probes OS. The condition ensures that probes hosted by residential users are indeed relying on resolving domain names at their ISPs DNS servers, unless explicitly configured otherwise by users.

The main detection criteria possible with the available measurements are the following:

- IP blocking - via traceroute hops. For this type of detection, first it has to be determined if ICMP and/or UDP datagrams can be sent towards the resources in question. Additionally, all other hops along the path need to allow this
- DNS filtering, URL blocking - via the response of the underlying name server queries executed as a sub-measurement to traceroute

4.2.2 The Pirate Bay

4.2.2.1 Obtained dataset

Around 1800 European probes were used in the experiment. Given the UDM probe count limitations, measurements were scheduled based on the maximum probe set length (Fig. 4.1). Retrieving results encompassed utilizing the RIPE Atlas API. Afterwards, results were combined into one set and their base64-encoded name server replies expanded. The retrieved dataset underwent filtering on the following criteria:

- DNS responses that list the local host (or an IP from the local hosts subnet) as the queried name server in the initial request are not considered. Only responses

from globally-routed IP addresses are considered, however no mapping is created for whether the IP belongs to the same ISP as the globally-routed IP of the probe

- DNS queries may fail due to being incapable of reaching their resolver. Given residential probes, unreachability may be the result of the ISP (1) provided default-gateway being down or (2) queried DNS server timing out. Both conditions are expressed by different DNS error messages and are filtered out
- A lot of results had error messages that did not relate to DNS. It was established that they are the result of bugs with the DNS tests themselves [22] and thus filtered out

4.2.2.2 Results

ThePirateBay owns AS51040, with all its webservers located in the 194.71.107.0/24 prefix (as per observed global DNS responses). Due to both ICMP and UDP datagrams being blocked for all resources in the subnet, it is impossible to determine endpoint reachability without using specialised tools such as TCP traceroute. Therefore, in order to verify whether an ASN performs IP blocking while still providing users with valid TPB IP addresses in DNS responses, first it was noted what are the last observed hops in UDP traceroutes.

Evaluating the dataset on the aforementioned criteria, resulted in the map seen in Fig. 4.6 of European countries where censorship occurs. The map was built with data obtained prior to Dutch court ruling in favour of not blocking the website in early February 2014.

The map includes results exclusively for DNS filtering, redirection and URL blocking with the majority of results accounted for by DNS filtering and redirection. Most ISPs opt for a solution where they continue to server actual TPB IPs in DNS queries, however users are redirected towards a webserver where a page shows a warning about the initially requested resource being blocked. No cases were observed where IP blocking occurs at the source ASN after a valid DNS record is returned.

Two ASNs were detected to be doing filtering in one geographic region, but not in another. These regions were governed by the regulations of different countries. Namely, UPC and Tele2 were doing DNS redirection in the Netherlands and Belgium, but not in Germany.

Comparing the results with public Internet censorship resources such as the Opennet Initiative, reveals a far-greater and accurate insight into where filtering is occurring.

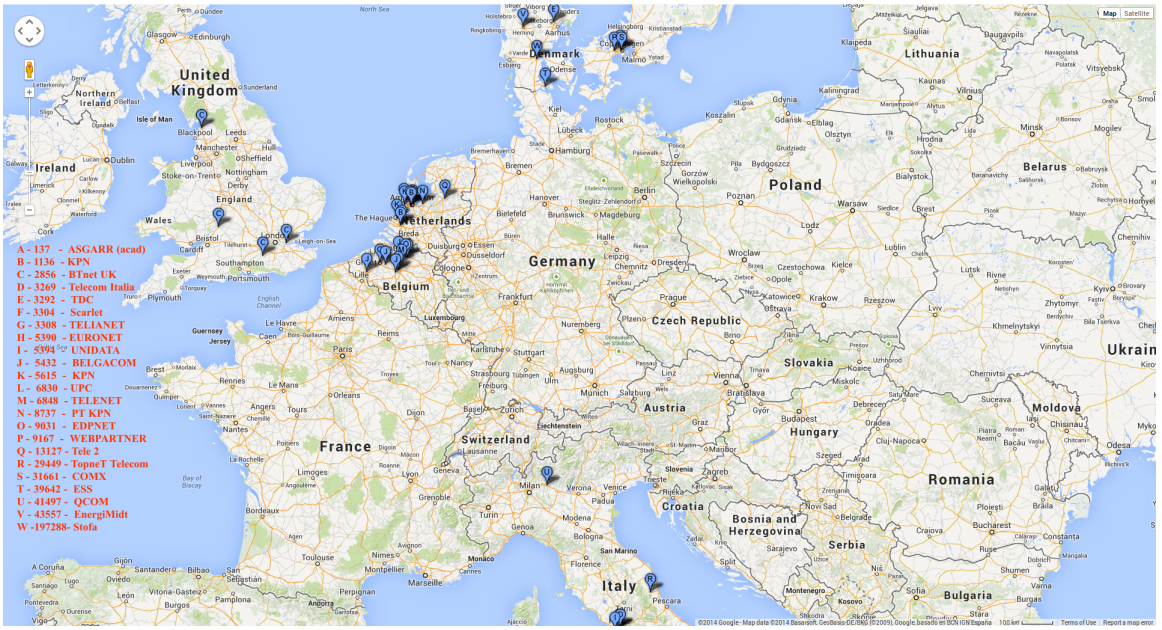

FIGURE 4.6: ThePiratebay DNS filtering, redirection and URL blocking in Europe

The Opennet Initiative lacks any data on current filtering happening in Italy, where ThePirateBay has been officially blocked in the past, however the ban has been lifted in 2011 [2].

4.2.3 LiveJournal and Greenpeace Russia experiments

To further enforce results on Internet censorship, two additional experiments were carried out that connect to recent events. Reportedly, the Greenpeace website was blocked in Russia on numerous occasions in the fall of 2013. By using a probe set with only Russian probes, tests showed that no blocking is in place at the time. Likewise is the case with LiveJournal. These experiments used a slightly different approach from the one used with ThePirateBay, as the resources were not hosted in ASNs and prefixes owned by them, but rather were use hosting companies. Their websites are accessible via a single load-balanced IP address. Therefore, also detecting IP blocking was possible, however no such cases were observed.

4.2.4 Social media filtering

A number of tests were planned to be executed against social media networks. This was impossible due to the Atlas limitation which prevents more than 10 measurements to be executed against the same URL or IP address. Although larger social network such as

[2] http://en.wikipedia.org/wiki/Countries_blocking_access_to_The_Pirate_Bay

Facebook own ASNs and serve their websites via a multitude of IPs and prefixes, it is still a favourable condition not to test against arbitrary IPs owned by the organisation, but rather against its URL as served by DNS name servers. At current, users of RIPE Atlas have defined continuous tests against most major social networks and scheduling further tests for them by using their URL is impossible.

4.2.5 Summary

Effective Internet censorship may occur on various levels in the process of establishing end-to-end host communication. The vast number of probes and the suitable test types in RIPE Atlas make it a robust and versatile tool to enforce the swift detection of censorship. The introduction of the newly-planned measurements (Section 2.2) would cover all possible vectors of Internet censorship, including website-specific ones where blocking occurs on the Application Layer and blocks users based on properties such as login credentials.

A number of countries were detected where no government regulation exists on accessibility of ThePirateBay, however censorship occurs nonetheless. Such cases may be due to the specific regulations of individual ISPs. RIPE Atlas can aid the process of determining such ISP-specific regulations.

The composite amount of credits consumed spent during the Internet censorship experiments is negligible and in the order of tens of thousands(or a bit more).

4.3 Prefix hijacking discussion

As discussed in Section 3.3.3, detecting prefix hijacking via data-plane probing alone relies on a two-step detection mechanism. First, one needs to monitor the path distance to a prefix from a given number of vantage points. And second, path disagreement detection needs to be established by monitoring another prefix part of the same ASN and physically close in the network, in terms of routing and network hops, to the monitored prefix. Ideally, in a scenario where an organisation has a single IGP router towards its upstream provider, an IP address from the link between the provider and customer IGP devices can be monitored. However, this implies that in case of multiple IGP devices each probe needs to monitor each one as well as its reference, with every such reference being different. Otherwise false-positives will be generated, because packets towards the target prefix may be routed over any of the IGP devices given normal network failures, such as nodes along the path(s) to one of the IGP entry points going offline. Such cases can quickly become overwhelming in terms of credit consumption.

The first scheduled prefix hijacking experiment tried to monitor Facebook for hijacking. The test was not possible due to a large number of IGP devices that distribute traffic towards all prefixes owned by the social network. It was determined that 54 unique IPs are part of the first-hop in Facebook's ASN after examining traceroute results.

The second scheduled experiment examined whether prefixes that belonged to smaller organisations, such as the one previously described, can be monitored from RIPE Atlas for prefix hijacking. In a simple experiment, the OS3 prefix 145.100.96.0/20 was used as a target. Additionally, 145.145.19.186 which is part of the last hops before OS3s Border-Service-Router (BSR) in SUFNnet's network was used as a reference target. The test used an One-Off traceroute measurement with around 1200 probes with administrative access to one of the probes and its network. Having access to one of the probes made it possible to test condition 1 of the the detection process. By defining a static route on the default-gateway servicing the probes network of residence that points traffic for 145.100.96.0/20 towards another destination, it can be seen whether such information is visible in the resulting RIPE Atlas traceroute results. By manual inspection, i.e. inspecting only traceroute outputs from the particular probe whose private network was tampered, implementing condition 1 of the detection process is satisfied. However, since all used devices in the test are part of production networks, it was impossible to extend the experiment to include verification of condition 2 from the detection process, namely tracking path disagreements stemming from valid network changes in the data-plane. Additionally, a robust data-plane detection mechanism needs to keep a large amount of historical data on the previously observed path distances from every probe, so that detection can be accurate, but devised scripts did not create such historic archives to compare newly-obtained data against. Nonetheless, both requirements can be satisfied. The second detection rule can again be observed in returned traceroute measurements with all other necessities being a matter of programming.

RIPE Atlas offers a highly optimal selection process of vantage point locations. The geo-location, ASN and prefix attributes associated with its probes can be combined with other datasets such as RIPE NCC's ASN peer overview archive [3] in addition to data from RouteViews and tools such as BGPlay to devise optimal probe placement. That way, the needed number of vantage points can be highly reduced while ensuring that the dispersion of vantage points is such, that a large amount of foreign ASN subtrees are covered given an origin prefix p. Albeit its limited results, the previous experiment showed that establishing personal prefix hijacking systems by using RIPE Atlas is a viable area for engineering. As argued in [6], only the origin itself (prefix owner) can easily and accurately distinguish between a legitimate origin change and a prefix hijack, however he first needs to be aware of it. RIPE Atlas can help exactly in that area,

[3] https://stat.ripe.net/widget/asn-neighbours

becoming aware of the hijack or MitM routing attack in the first place, by relying on personal and highly-customised tools that can generate notifications in a timely and accurate manner. Both personal as well as industry-grade hijacking systems can benefit from utilizing RIPE Atlas and its probe selection utilities within their detection mechanisms. Personal solutions can leverage information from RIR databases to reduce the needed number of probes and hence credit consumption. A personal detection system using a number of probes in the order of hundreds and an intensity of 30 minutes between traceroutes can be easily sustained to run for extended periods of time under a normal RIPE Atlas account. Additionally, industry-grade solutions which use both control- and data-plane probing in their detection criteria can leverage RIPE Atlas in the last-mile detection process.

Chapter 5

Conclusions

During this project, RIPE Atlas was examined as a viable system for detecting Internet routing anomalies. The focus was on establishing whether the technical specifications of the measurement system enable the tracking of Debogon filtering, Internet Censorship and MitM routing attacks. By providing a literal study of currently used detections methodologies and synthesizing their detection criteria, a number of experiments were devised with RIPE Atlas to test it as a viable replacement. The experiments study three important characteristics of the RIPE Atlas measurement system. First, they examine whether the synthesized detection criteria are possible within the scope of RIPE Atlas. Second, an analysis of the way measurements are scheduled by the system and the limitations that exist with it is performed. Third, the way resulting measurement datasets are composed is observed. Finally, by relying on gathered results, the project provides ways of using RIPE Atlas for the detection of the aforementioned anomalies. Results shows that the system is a viable candidate for detecting anomalies both in the data-plane as well as in the control-plane of the Internet. It can be used as a replacement for various subcomponents of large-scale detection systems that target different suspicious network events.

The conducted experiments have shown that RIPE Atlas offers a very large network of global vantage points and that probe sets can be build-up upon sophisticated criteria. In addition, the system's network measurement facilities enable the detection of all examined routing anomalies.

Debogon experiments show that global reachability problems may still exist with debogonised prefixes long after their status as such has been relinquished. However, ASNs at which such filtering occurs, were not part of crucial consumer-based provider networks.

Internet censorship experiments show that RIPE Atlas can be used as a robust tool for detecting censors on both the regional as well as global scale.

Finally, personalised detection systems for prefix hijacking are possible with RIPE Atlas and through the usage of other numbering resources made available by RIPE NCC.

Given that RIPE Atlas provides a historical archive of all previously conducted UDMs that are marked as public, there may be cases where the archives can be used to study known network events and anomalies. Mining procedures may seek out public entries in the system which relate to network test conducted against a certain target of interest.

Chapter 6

Future work

Currently, no Internet filtering detection systems exist that are based on RIPE Atlas. Given that it has a good ASN coverage, an Internet censorship detection system based on it is a favourable future work. [8] can directly benefit from using RIPE Atlas in its detection process. Additionally, with the introduction of wget and curl, all vectors of Internet censorship detection will be possible, such as ones where application-specific censors are applied that block users based on their login credentials.

The data-plane debogon experiments are an interesting topic that can be further extend upon with other special prefixes, such as the remainder of last allocation APNIC has received [14, 16] or any of the other LIRs. Debogon experiments were using pings targets without any prior consent from. As a result, observations were made that a number of the high-intensity pings directed towards bogons might have been dropped after their first few initial execution intervals. However, there were cases where a much lower amount of successful pings were carried out in one time interval and a much higher one in the next. Re-establishing measurements to use traceroute might provide insights into whether this condition is due to network path changes, and whether the paths to which the network switches contain hops at which improper debogon filtering exists.

Another important area of future work is to better define a prefix hijacking system based on RIPE Atlas that ensures optimum coverage of remote ASN sub-trees and low credit consumption. Also, a more in-depth examination of RIPE Atlas applicability within the scope of [6, 7] is another area of improvement.

Appendix A

RIPE Atlas UDM IDs for each experiment

This Appendix lists the measurement IDs of most main conducted measurement. For verification measurement of debogon tests, please refere to the RIPE Atlas archives of user r dot yakimov at os3 dot nl. This Appendix lists the measurement IDs of most main conducted measurement. For verification measurement of debogon tests, please refere to the RIPE Atlas archives of user r dot yakimov at os3 dot nl.

Debogon filtering

Test	Targets	RIPE Atlas measurement IDs
103.1.0.0/22	103.1.0.1	1425147, 1425149, 1425151, 1425153, 1425155, 1425157, 1425159
	84.205.83.1	1425148, 1425150, 1425152, 1425154, 1425156, 1425158, 1425160
103.23.28.0/24	103.23.28.101	1429510, 1429457, 1429499, 1429501, 1429503, 1429506, 1429508
	202.65.145.219	1429456, 1429458, 1429500, 1429502, 1429504, 1429507, 1429509
103.247.191.0/24	103.247.191.11	1428391, 1428393, 1428395, 1428397, 1428399, 1428401, 1428403
	203.142.206.132	1428392, 1428394, 1428396, 1428398, 1428400, 1428402, 1428404
128.0.144.0/21	128.0.144.145	1425236, 1425238, 1425240, 1425242, 1425244, 1425246, 1425248
	85.195.69.99	1425237, 1425239, 1425241, 1425243, 1425245, 1425247, 1425249
185.2.136.0/22	185.2.137.24	1425343, 1425345, 1425347, 1425349, 1425351, 1425353, 1425356
	83.170.64.2	1425344, 1425346, 1425348, 1425350, 1425352, 1425354, 1425357
185.24.0.0/22	185.24.0.1	1423636, 1423638, 1423640, 1423642, 1423644, 1423646, 1423648
	84.205.83.1	1423637, 1423639, 1423641, 1423643, 1423645, 1423647, 1423649

Internert censorship

Experiment	RIPE Atlas Measurement IDs
ThePirateBay	1425252, 1425253, 1425254
LiveJournal	1425199
GreenPeace	1425164, 1425165
Facebook	1421842, 1421843, 1421845, 1421846, 1421847, 1421848, 1421849

Prefix hijacking: 1421897, 1421923

Appendix B

Code listings debogon filtering

The provided snipplet are just stencils and might have underwent additional modifications for the context of a given measurement. Additionally, further BASH processing might have been applied to the generated output by the scripts.

B.1 Measurement reservation

The following code snipplet was used for devising probe sets and reserving debogon measurements.

```
#!/usr/bin/env python

import urllib2, urllib, json, time, sys
from random import choice

#Build probe set
#---------
#Get data from RIPE ATlas REST
result = list()
for i in range(0,7500,100):
  response = urllib.urlopen("https://atlas.ripe.net/api/v1/probe/?fields=status,id,country_code,prefix_v4,asn_v4&limit=7700&offset="+str(i))
  jsondata = json.loads(response.read())
  (true,false,null) = (True,False,None)
  result.extend([eval(str(jsondata))])

#Retain only object part of result entries
probe = list()
for i in result:
  probe.extend(i.get("objects"))

#Retain only online probes
probe = [i for i in probe if i.get("status") == 1]

#Retain only probes from unique prefixes by creating a dict with prefixes as keys. This will ultimately drop every >1 insertion with the same key
probeDict = {}
for i in probe:
  if i.get("prefix_v4") and i.get("asn_v4"):
    probeDict[i.get("prefix_v4")] = [i.get("id"), i.get("asn_v4")]
```

```
29
30 #Remove probes from RIPE NCC's ASNs
31 ripeASNs = [2121,3333,12654,12898,34964,196615]
32 keysToRemove = [k for k,v in probeDict.items() if int(v[1]) in ripeASNs]
33 for key in keysToRemove:
34   if key in probeDict.keys():
35     del probeDict[key]
36
37 #Retain only probe IDs
38 probe = [v[0] for v in probeDict.values()]
39
40 #Order probes within lists of size <=500. Required due to limitation in atlas UDM
41 lol = lambda lst, sz: [lst[i:i+sz] for i in range(0, len(lst), sz)]
42 probe = lol(probe, 500)
43 #---------
44
45 #Schedule measurement from each probe
46 #The measurement has the following properties:
47 #separate measurement for each ~500 hosts in the network
48 #intensity: 20 minutes
49 #duration: 24 hours
50 #Packets per/MSM: 3
51 #Packet size: 48
52 #
53 #
54 #Also launches a control ping towards Routing Beacon RRC3 AMS-IX
55 #intensity: 60 minutes
56 #duration: 24 hours
57 #Packets p/mm: 3
58 #packet size: 48
59 target = {"128.0.0.0/16"  :"128.0.144.145",
60           "185.1.0.0/21"  :"185.1.1.1",
61           "185.1.24.0/24 ":"185.24.0.1"}
62 for prefix in target.keys():
63   #Base Restful API requirements
64   key = "20cf1905-41c9-4e3d-b83f-93c06769caa7"
65   url = "https://atlas.ripe.net/api/v1/measurement/?key=%s" % key
66   request = urllib2.Request(url)
67   request.add_header("Content-Type", "application/json")
68   request.add_header("Accept", "application/json")
69   chunkID=0
70   for chunk in probe:
71     data = { "definitions":
72              [
73                {
74                  "af": "4",
75                  "type": "ping",
76                  "target": target.get(prefix),
77                  "packets":3,
78                  "size": 48,
79                  "interval": 1320,
80                  "is_oneoff": False,
81                  "is_public": True,
82                  "description": ""#"Debogon "+prefix+" with reference point 84.205.83.1 chunk"+str(chunkID)
83                },
84                {
85                  "af": "4",
86                  "type": "ping",
87                  "target": "84.205.83.1",
88                  "packets":3,
89                  "size": 48,
90                  "interval": 3307,
91                  "is_oneoff": False,
92                  "is_public": True,
93                  "description": ""#"Debogon "+prefix+" reference point chunk"+str(chunkID)
94                }
95              ],
96              "probes":
97              [
98                {
99                  'requested': '500',
100                 'type': 'probes',
101                 'value': ",".join(str(v) for v in sorted(chunk))
102               }
103             ],
```

```
            "start_time": str(int(time.time())),
            "stop_time": "1391032800"
          }
    try: conn = urllib2.urlopen(request, json.dumps(data))
    except urllib2.URLError, e:
      print e.reason
    chunkID=chunkID+1
    time.sleep(20)
```

B.2 Measurement results aggregation

The following code snipplet was used for downloading and combining the datasets of debogon tests:

```
#!/usr/bin/env python
import urllib2, urllib, json, time, pickle

#Convert to form {PID1:[ping,ping,...], PID2:[ping,ping,...]}
def joinProbePings(rawMSMdata):
    ping = dict()
    for pingResult in rawMSMdata:
        prb_id = pingResult.pop("prb_id")
        if prb_id in ping.keys():
            ping[prb_id].append(pingResult)
        else:
            ping[prb_id] = [pingResult]
    return ping

#Download results
#-------------------------------------------------------------------------------
#Download all parts of pings towards the bogon. They are part of MSMIDs:
#   1425343,1425345,1425347,1425349,1425351,1425353,1425356         |  185.2.137.24
#   1425344,1425346,1425348,1425350,1425352,1425354,1425357         |  83.170.64.2
data = list()
result = list()
for i in [1425344,1425346,1425348,1425350,1425352,1425354,1425357]:
    response = urllib.urlopen("https://atlas.ripe.net/api/v1/measurement/"+str(i)+"/result/")
    jsondata = json.loads(response.read())
    ##Convert to python objects
    #(true,false,null) = (True,False,None)
    result.extend(eval(str(jsondata)))
    time.sleep(1)
data = joinProbePings(result)

#Write data to file
with open("83.170.64.2.pyMeasurementData", 'wb') as f:
    pickle.dump(data, f)
```

B.3 Measurement results analysis

The following code snipplet was used to analyze the aggregate results of individual debogon tests

```
#!/usr/bin/env python
import urllib2, urllib, json, datetime, pickle, itertools
from operator import itemgetter


with open("185.2.137.24.pyMeasurementData", 'rb') as f:
  bogonPing = pickle.load(f)
with open("83.170.64.2.pyMeasurementData", 'rb') as f:
  referencePing = pickle.load(f)

#Order values in both sets by timestamp
#------------------------------------------------------------
for k,v in bogonPing.items():
  sortedList = sorted(v, key=itemgetter('timestamp'))
  bogonPing[k] = sortedList
for k,v in referencePing.items():
  sortedList = sorted(v, key=itemgetter('timestamp'))
  referencePing[k] = sortedList
#------------------------------------------------------------
#Put all pings within the same timeslices (normalize), by getting first timestamp and modifying
#all others to be a subsum of it
#------------------------------------------------------------
startTimestamp = 1391442334
endTimestamp = 1391484663
for pings in bogonPing.values():
  step = 0
  #Iterate over all ping results and change the
  for j in pings:
    j["startTimestamp"] = startTimestamp +(step*1200)
  step +=1
#------------------------------------------------------------
#Filter results from probs that can't reach (ICMP unreachable) both the bogon and the reference
#for the entire duration, or have errors in their result#
#------------------------------------------------------------
#First from the referene pings get all probe IDs, from which the target was unreachable for the
#duration of the entire test
probeIDs1 = list()
for prb_id, pingData in referencePing.items():
  #Get all keys associated with values containing 3 starts (in the result section) for the duration of all pings
  unreachable = 0
  for pingResult in pingData:
    if str(pingResult.get("result")).count("*") == 3:
      unreachable +=1
  if unreachable == len(pingData):
    probeIDs1.append(prb_id)
    unreachable = 0
  else:
    unreachable = 0

#Next, from the bogon pings, check the pings results of all previously collected probe IDs
probeIDs2 = list()
for prb_id in probeIDs1:
  #Count unreachable ping results in corresponding reference set
  unreachable = 0
  for pingData in bogonPing.get(prb_id):
    if str(pingData.get("result")).count("*") == 3:
      unreachable +=1
  if unreachable == len(bogonPing.get(prb_id)):
    probeIDs2.append(prb_id)
  unreachable = 0

#Remove measurements for probes that could not reach either target for the entire duration
toBeRemoved = list(set(probeIDs1).intersection(probeIDs2))
for i in toBeRemoved:
  bogonPing.pop(i, None)
```

```
  referencePing.pop(i, None)
#-------------------------------------------------------------
#Get all cases in which reference was pingable but bogon wasnt
#By filtering on start, it is ensured that error ICMP replies are not considered. Previous section counts them
#-------------------------------------------------------------
bogonProbe = list()
for prb_id, pingData in bogonPing.items():
  #Get all keys associated with values containing 3 starts (in the result section) for the duration of all pings
  unreachable = 0
  for pingResult in pingData:
    if str(pingResult.get("result")).count("*") == 3:
      unreachable +=1
  if unreachable == len(pingData):
    bogonProbe.append(prb_id)
  unreachable = 0
print bogonProbe
#-------------------------------------------------------------
#Create timeline plot dataset, such that output is two columns: (Time, NumberOfProbesReaching)
#-------------------------------------------------------------
#Next, iterave over each timeslice and get active probes within it(at least one ICMP reply received)
sliceCounter = 0
slices = dict() #11 in total
for time in range(1391442334, 1391484663, 1200):
  #Iterate over each probe for each timeslice
  for prb_id, result in bogonPing.items():
    #Iterate over probe results and capture those withime upper and lower timeslice to the new dict in case there is a ping whose packet was anwered
    for ping in result:
      if ping.get("timestamp") >= time and ping.get("timestamp") <=time+1200 and str(ping.get("result")).count("*") < 3:
        if sliceCounter in slices.keys():
          slices[sliceCounter].append(prb_id)
        else:
          slices[sliceCounter] = [prb_id]
  sliceCounter +=1
#Finally output data to a file
startTimestamp = 1391442334
counter = 0
for sliceID, probeIDs in slices.items():
  with open("timeseries185.2.137.24.data", 'a+') as f:
    #time = datetime.datetime.fromtimestamp(startTimestamp+counter*1200)
    #f.write(str(time.strftime('%H:%M'))+"    "+str(len(set(probeIDs)))+"\n")
    f.write(str(startTimestamp+counter*1200)+","+str(len(set(probeIDs)))+"\n")
    counter += 1
#-------------------------------------------------------------
#Create timeline plot of amount of packets received, such that output is two columns(Time, SuccessfulPings)
#-------------------------------------------------------------
sliceCounter = 0
slices = dict() #11 in total
for time in range(1391364032, 1391406395, 1200):
  #Iterate over each probe for each timeslice
  unsuccessfull = int()
  total = int()
  for prb_id, result in bogonPing.items():
    for ping in result:
      if ping.get("timestamp") >= time and ping.get("timestamp") <=time+1200:
        total += 3
        unsuccessfull = unsuccessfull + str(ping.get("result")).count("*")
  if sliceCounter in slices.keys():
    slices[sliceCounter].append([total, unsuccessfull])
  else:
    slices[sliceCounter] = [total, unsuccessfull]
  sliceCounter +=1
print slices
#-------------------------------------------------------------
```

Appendix C

Code listings Internet censorship

C.1 Measurement reservation

The following code snipplet was used to build up a probe set for ThePirateBay measurements. A derivative of the script on page 43 was used for scheduling the measurements in RIPE Atlas.

```
#!/usr/bin/python

import json, urllib
"""
This script retrieves all online EU probes outside of a specified AS and its immediate neighbours
"""

#Get all probes, with their IDs, country codes, status code and ASN numbers
response = urllib.urlopen("https://atlas.ripe.net/api/v1/probe/?fields=status,asn_v4,id,country_code,prefix_v4&limit=7700")
jsondata = json.loads(response.read())
(true,false,null) = (True,False,None)
result = eval(str(jsondata))

#Filter out all non-connected probes
probe = [i for i in result.get("objects") if int(i.get("status")) == 1]

#Filter out non-EU probes
euCountryCodes = ['AD', 'AL', 'AT', 'BE', 'BG', 'BY', 'CZ', 'DE', 'DK', 'EE', 'FI',
                  'FR', 'GR', 'HU', 'IE', 'IS', 'IT', 'LI', 'LT', 'LU', 'LV', 'MK',
                  'MT', 'NL', 'NO', 'PL', 'PT', 'RO', 'RU', 'SE', 'SI', 'SK', 'SM',
                  'UA', 'VA', 'BA', 'HR', 'MD', 'MC', 'ME', 'RS', 'ES', 'CH', 'GB']
probe = [i for i in probe if i.get("country_code") in euCountryCodes]

#Retain only probes from unique prefixes by creating a dict with prefixes as keys. This will ultimately
#drop every >1 insertion with the same key
probeDict = {}
for i in probe:
  if i.get("prefix_v4") and i.get("asn_v4"):
    probeDict[i.get("prefix_v4")] = [i.get("id"), i.get("asn_v4")]

#Retain only probe IDs
probe = [v[0] for v in probeDict.values()]

#Order probes within lists of size <=500. Required due to limitation in atlas UDM
lol = lambda lst, sz: [lst[i:i+sz] for i in range(0, len(lst), sz)]
probe = lol(probe, 500)
print probe
```

C.2 Measurement results aggregation

The following code snipplet was used to retrieve and aggregated the dataset of scheduled measurements.

```
#!/usr/bin/env python
import urllib2, urllib, json, time, pickle


result = list()
for i in [ 1425262,1425263,1425265,1425266,1425269]:
    response = urllib.urlopen("https://atlas.ripe.net/api/v1/measurement/"+str(i)+"/result/")
    jsondata = json.loads(response.read())
    ##Convert to python objects
    #(true,false,null) = (True,False,None)
    result.extend(eval(str(jsondata)))
    #result.extend(jsondata)
    time.sleep(1)


#Write data to file
with open("traceRouteData2", 'w+') as f:
    #pickle.dump(result, f)
    f.write(str(result))
```

C.3 Measurement results analysis

The following code snipplet was partially used to analyze ThePiratebay results.

```
#!/usr/bin/env python
import urllib, json, time, pickle, ipaddr, re
from netaddr import *

with open("traceRouteData", 'rb') as f:
  data = pickle.load(f)

#Filter dns error messages and results from badly configured DNS servers
successfull = list()
for i in data:
  if "error" not in str(i.get("result")) and "127.0.0.1" not in i.get("dst_addr"):
    successfull.extend([i])

#Determine probeIDs that do not resolve to pirabay prefix
probe = list()
for i in successfull:
  if ipaddr.IPAddress(i.get("dst_addr")) not in ipaddr.IPNetwork('194.71.107.0/24'):
    probe.extend([i])
print probe


#Get probe geo data, asn name
print "Latitude,Logitude,ASN_Name"
for i in probe:
  response = urllib.urlopen("https://atlas.ripe.net/api/v1/probe/?id="+str(i.get("prb_id"))+"&fields=latitude,longitude,asn_v4")
  jsondata = json.loads(response.read())
  result = eval(str(jsondata))
  print result.get("objects")[0].get("latitude"), result.get("objects")[0].get("longitude"), result.get("objects")[0].get("asn_v4")

#Get probes where URL resolves to a valid TPB prefix, however, the last hop isn't a TPB prefix
probe = dict()
for traceroute in successfull:
```

```
  prb_id = traceroute.get("prb_id")
  if ipaddr.IPAddress(traceroute.get("dst_addr")) in ipaddr.IPNetwork('194.71.107.0/24'):
    for result in traceroute.get("result")[::-1]:
      if "from" in str(result):
        print traceroute.get("prb_id")
        matches = re.findall('u\'from\': u\'\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}\',', str(result))
        for i in matches:
          ip = i.split("'")[3]
          if not IPAddress(str(ip)).is_private() and not IPAddress(str(ip)).is_reserved()
             and not IPAddress(str(ip)).is_multicast() and not IPAddress(str(ip)).is_loopback():
            if prb_id in probe.keys():
              probe[prb_id].extend([ip])
            else:
              probe[prb_id] = [ip]
    break
for k,i in probe.items():
  for j in i:
    print j, k
```

Bibliography

[1] Tongqing Qiu, Jia Wang, Lusheng Ji, Dan Pei, Hitesh Ballani, "Locating Prefix Hijackers using LOCK", 2009

[2] Xin Hu, Mao Z.M., "Accurate real-time identification of IP Hijacking", IEEE Security and Privacy 2007

[3] Zheng, Changxi and Ji, Lusheng and Pei, Dan and Wang, Jia and Francis, Paul, "A Light-Weight Distributed Scheme for Detecting IP Prefix Hijacks in Realtime", SIGCOMM 2007

[4] Khin Thida Latt, Yasuhiro Ohara, Satoshi Uda and Yoichi Shinoda, "Analysis of IP Prefix Hijacking and Traffic Interception", 2010

[5] Chi, Ying-Ju and Oliveira, Ricardo and Zhang, Lixia, "Cyclops: the AS-level connectivity observatory", SIGCOMM Comput. Commun. Rev. 2008

[6] Mohit Lad, Dan Massey, Dan Pei, Yiguo Wu, Beichuan Zhang, Lixia Zhang, "PHAS: A Prefix Hijack Alert System", 2006

[7] Yang Xiang, Zhiliang Wang, Xia Yin, Jianping Wu, "Argus: An accurate and agile system to detecting IP prefix hijacking", Tsinghua Nat. Lab. for Inf. Sci. Technol. (TNList), Beijing, China, 2011

[8] Antonio Pescapè, Giuseppe Aceto, "UBICA: User-based Internet Censorship Analysis", University of Napoli "Federico II", 2013

[9] Geoff Huston, George Michaelson, "Traffic in Network 14.0.0.0/8 and 223.0.0.0/8", APNIC RD, 2010

[10] Mirjam Kühne, "Interesting Graph - How Complete is the RIPE Routing Registry", RIPE Labs, 2010

[11] Daniel Karrenberg, "De-Bogonising New Address Blocks", RIPE Community, 2005

[12] Emile Aben, "The Curious Case of 128.0/16", RIPE Labs, 2011

[13] Kjell Leknes, "De-bogonising 128.0.0.0/16", RIPE Labs, 2012

[14] APNIC Resource Reachability Testing, 2014

[15] RIPE RIS Routing Beacons, 2013

[16] IANA IPv4 Address Space Registry

[17] INTERNIC Root Name Servers reference

[18] Team CYMRU Community Servics, The Bogon Reference

[19] Global RIPE Atlas Network Coverage

[20] RIPE NCC Address Space Hierarchy and allocations

[21] RIPE Atlas - User-Defined Measurements

[22] RIPE Atlas bugs with DNS measurement