TASRS: Towards a Secure Routing System Through Internet Number Resource Certification

[Verisign Labs Technical Report #1130009]

Eric Osterweil
Verisign

Shane Amante
Level(3) Communications

Danny McPherson
Verisign

February 3, 2013

Abstract

The Border Gateway Protocol (BGP) is the Internet's core routing protocol. Today, any BGP speaker can attest to be the origin for any IP prefix, and this results in the implicit assertion that this ASN is the rightful resource holder for that IP prefix. Unfortunately, any BGP speaker can lie about this. Such lies have allowed BGP speakers to "leak routes" and "hijack" each others prefixes. There has never been a programmatic way to securely disambiguate who the rightful resource holder actually is for any given IP prefix, or any Autonomous System Number (ASN). This leaves a very pronounced need for an Internet number resource certification framework. BGP, for example, currently propagates unverified announcements from origin ASNs to their neighbors, who selectively re-announce them to their neighbors in a form of gossip. Meanwhile, none of these parties have any way to properly assure the veracity of the announcements they are working from.

We assert that the Internet needs Internet number resource certification and that routing stability suggests that a proactive approach is paramount. For that, we propose the *Towards A Secure Routing System (TASRS)* architecture, so named because it enables the security of the tried an true operational practice used for origin-based assurances in BGP. The TASRS architecture is evolved from the simple observation that securing the route computation can be done by learning an AS' resources and corresponding routed resource policies, and then proactively verifying the set of allowed states at each BGP peer. Moreover, this proactive practice already exists today, in service providers that maintain a trusted list of clients and policies that are known to be genuine and accurate. With a secure Internet number resource certification framework, routing policy semantics can be securely discovered and used to effectuate verifiable intent of operators in inter-domain routing.

1 Introduction

The Border Gateway Protocol (BGP) [17] is the Internet's core routing protocol. Today, any BGP speaker can attest to be the origin for any IP prefix, and this results in the implicit assertion that this ASN is the rightful resource holder for that IP prefix. Unfortunately, any BGP speaker can lie about this. Such lies have allowed BGP speakers to "leak routes" [13] and "hijack" each others prefixes. There has never been a programmatic way to securely disambiguate who the rightful resource holder actually is for any given IP prefix, or any Autonomous System Number (ASN). This leaves a very pronounced need for an Internet number resource certification framework. BGP, for example, currently propagates unverified announcements from origin ASNs to their neighbors, who selectively re-announce them to their neighbors in a form of gossip. Meanwhile, none of these parties have any way to properly assure the veracity of the announcements they are working from.

Even though the Internet Assigned Numbers Authority (IANA) and Regional Internet Registries (RIRs) are the allocation authorities for Internet number resources, creating a framework that allows users to programmatically verify who the rightful holders of resources are has been elusive. That is, there is no way to build automated systems in which a Relying Party (RP) can securely verify the binding of Internet number resources to the proper resource holder. This underscores the need for RPs to be able to securely query some sort of distributed database that can map Internet number resources into a collision-free/unique namespace, and which can delegate authority for portions of the global Internet number-space to authorized resource holders. With a database like this the policy information for routing policies (such as ASNs that are allowed to announce an IP prefix) can be securely and unambiguously managed and exported by authorized resource holders, and verified by RPs.

In the case of BGP, there is a dire need for BGP speakers to be able to verify who the authorized origin for each routed IP prefixes is. In this work, we categorize approaches to this problem into two fundamentally different perspectives: one can be *proactive*, or one can be *reactive*. The proactive model has been in use throughout the Internet since the 1990s. Many large ISPs generate route filters from their Internet Routing Registry (IRR). This allows resource holders to add their own policy information in their own operational time frames, and allows Level(3) to programmatically ingest policy and construct filters that (among other things) enforce certain routing security semantics at Level(3)'s own operational time frame. The pros of this approach are that it has worked successfully in Internet operations since the 1990s, it achieves failure isolation through loosely coupling different operational processes, it is transparent (users can see what is happening and manually fix any problems they see), and it does not interfere with BGP's control plane (so outages do not interfere with routing). Its cons are that it must know what to generate filters on a priori (i.e. which ASNs to use), and it requires an Internet number resource certification framework for filter generation to be able to know which IRR is authoritative for which Internet resources. That is, if a user wants to put their policy information in their own IRR, how can an ISP find this (because there are many IRRs), and how can they differentiate it from an attacker's attempt to fool them? We claim that it is this last question that begs for an Internet number resource certification framework. With that, a resource holder can securely expose their policy.

The reactive model includes systems like the recently proposed Route Origin Verifier. This system passively monitors BGP updates as they arrive at routers and tries to verify their authenticity while the router is making routing policy computations. In fact, it is very similar to a failed approach tried in 1998 [4]. The pros of reactive approach are that it is simple to discover what to filter on (it just watches update streams). However, its cons are serious: it imposes systemic latency and any failures encountered during the verification process impact the stability of routing. Specifically, its latency is inherent because Relying Parties (RPs) must wait for BGP updates to arrive before they know what information to verify, or where to verify it. This lets the issuers of BGP updates influence the verification process' timing, and this sort of systemic influence has led to attack vectors in other systems (such as the Kaminsky attack in DNS [6]). Additionally, stability is at risk because if a DNS lookup fails, verification fails, and route computation outcomes become non-deterministic.

We assert that the Internet needs Internet number resource certification and that routing stability suggests that a proactive approach is paramount. For that, we propose the *Towards A Secure Routing System (TASRS)* architecture, so named because it enables the security of the tried an true operational practice used for origin-based assurances in BGP. The TASRS architecture is evolved from the simple observation that securing the route computation can be done by learning an AS' resources and corresponding routed resource policies, and then proactively verifying the set of allowed states at each BGP peer. Moreover, this proactive practice already exists today, in service providers that maintain a trusted list of clients and policies that are known to be genuine and accurate. What is needed, however, for an Internet-scale incarnation of this approach is a resource certification framework that allows every resource holder to securely announce their resource and their relevant routing policies, and also allows arbitrary parties in the Internet to learn this same information about any AS, in a verifiable way. We argue that what is needed for this is an Internet-scale distributed database that maps the authority of resource holders to the authority to update their corresponding portion of the database, and a conflict-free name space that all RPs can query to unambiguously and securely lookup resource information that is maintained by the authentic resource holders. For this, we turn to the Domain Name System (DNS) [14], an incumbent system that already contains a large portion of the necessary mechanisms and delegation data to enable a more secure routing system. In addition, DNS operations have years of experience maintaining and operating its infrastructure, and the DNS Security Extensions (DNSSEC) [1] add security to the name space.

With a secure Internet number resource certification framework, routing policy semantics can be securely discovered and used to effectuate verifiable intent of operators in inter-domain routing.

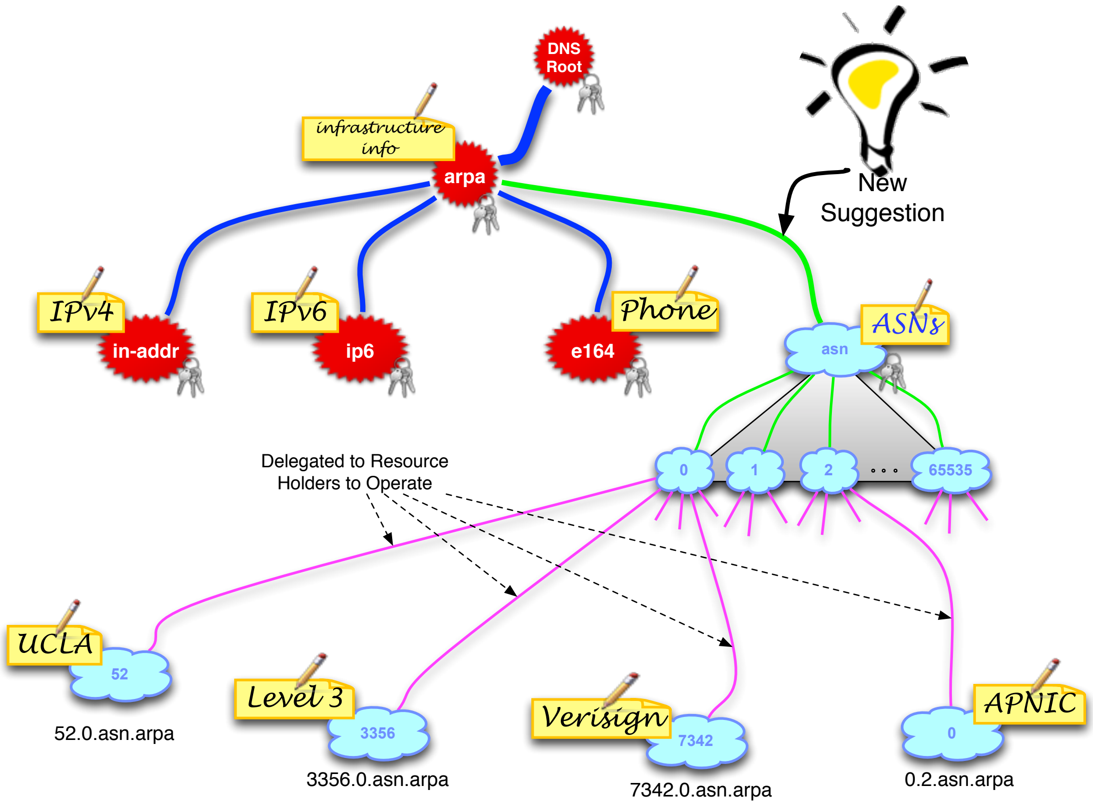

Figure 1: A depiction of how `asn.arpa` might be deployed, managed, and used.

1.1 Using DNS + DNSSEC as the Solution

In today's Internet, we rely on the ubiquitous Domain Name System (DNS) [14] to play some role in almost every transaction we conduct online. Recently, the DNS Security Extensions (DNSSEC) [2] have added origin authenticity, data integrity, and secure denial of existence to the DNS. What's more, the long standing practice of allocated "reverse DNS" zones for IP space (under the `in-addr.arpa` and `ip6.arpa` zones) to IP space holders has meant that those who hold certain IP space are granted authority for annotating the corresponding portions of DNS.

Some have claimed that while the *policy* of delegating reverse DNS space to IP resource holders exists, the *practice* of actually doing it has not kept up with IP allocations. However, in recent work [15], we presented measurements that indicated that the reverse DNS delegation tree does, indeed, seem to closely match the IP allocation space.

What we have today is a way for IP addresses (IPv4 and IPv6) to be mapped to DNS names, and for those names to be answered by authorities designated by the IP resource holders. Just as this facility has existed for some time, we propose that a similarly themed tree for Autonomous System Numbers (ASNs) would serve an equally useful role. We envision a zone called `asn.arpa`, in which resource holders are able to manage delegations that correspond to their ASNs. Figure 1 illustrates how the `.arpa` tree already exists, is a distributed database, embodies an administrative delegation hierarchy that follows that resource allocation hierarchy that exists today, and is extensible to present and future Internet resource certification needs. With this, DNS clients would be able to query for information that ASN holders have chosen to associate with their resource. For example, an ASN holder may want to indicate policy semantics, or where to find information about their policy semantics (such as their IRR of choice), or certificates or public keys to verify their resources, etc.

2 High Level Requirements

In this Section, we present the set of high-level requirements that frame the design of TASRS. The goal of these requirements is to allow the resource holders of IP resources (IPv4 and IPv6 address blocks) to securely advertise the aspects of their routing policies that prevent BGP origin hijacks, and suballocation hijacks.

1. All data retrieved from an Internet number resource certification framework must provide global uniqueness, authenticity, integrity, and secure denial of existence guarantees. This is a requirement in order to assure that verified data was delivered to RPs just as it was encoded by the resource holder, and neither the data, nor any lack thereof has been modified in flight.

2. RPs must be able to use either an IPv4 or an IPv6 prefix to unambiguously determine what ASN(s) (if any) the authorized resource holder has specified as being authorized to originate, or unambiguously determine if the authorized resource holder has not specified anything

3. Suballocation of resources (where feasible) should only involve the receiving party and their immediate predecessor. Hierarchical predecessors beyond the immediate parent of an allocated resource must have no *direct* influence over the suballocation of a resource. That is, each allocation exists as a policy agreement between a allocator and allocatee. This concept has been dubbed “grandparenting,” in the SIDR working group, and it must not be facilitated by this architecture as the aim here is to minimize unilateral actions that may impact non-adjacent parties [5].

4. RPs must be able to use an ASN to unambiguously determine that ASN’s authorized adjacencies and their import/export policies for specific prefixes. This information may be time dependent (who is a feasible path, and when), and may be topologically dependent (ASN X is an authorized transit next hop for ASN Y).

5. Resource holders must be able to directly create, update, and delete the publication of their own mappings, attestations, and policies without involving an external party or organization.

6. There must be a mechanism through which RPs can *feasibly* learn the list of allocated resources without prior knowledge.[1] This includes learning allocated prefixes, allocated ASNs, etc. The complexity of this process may vary across different types of resources, but consideration for this must be addressed for each type of resource.

3 Related Work

RPKI / BGPSEC The Resource PKI (RPKI) and BGPSEC [9] focus on what is routed on the Internet, rather than how it’s routed. The RPKI creates an X.509 certificate hierarchy that cryptographically certifies all AS number and IP prefix allocations. BGPSEC uses this cryptographic hierarchy and adds router certificates so that BGP AS path elements can be ’forward’ signed in a new secure path elements and corresponding AS paths can be verified accordingly. This approach is very much akin to one previously referred to as S-BGP [10].

This type of approach is proactive model, because all resource certification information for all prefixes is certified by resource holders in advance, and all Relying Parties (RPs) prefetch all information before validation begins. The challenge is that it does not prevent policy issues that may be of significant concern from an operational security perspective [13]. A detailed discussion on how this system scales can be found in [16].

ROVER The Route Origin Verifier is a reactive model because it waits for updates to seed verification. All verification happens during route computation, and external failures directly impact route computation without fallback options (i.e. you can’t decide to retry later unless you want to delay route computation until later too)

[1] Note, security precautions may prompt some resource holders to enforce access restrictions to this mechanism.

soBGP [18]

4 Architecture

TASRS' architecture allows resource holders to configure their own policies and their own information under their own authoritative services. In addition, it embraces the proactive verification model, whereby operators checking the veracity of routed resources (i.e,. Relying Parties, or RPs) can pre-verify information (rather than on demand) through the use of existing filter-list techniques (employed extensively throughout the operational community today).

TASRS' filter generation must be primed with a list of ASNs. Operators may determine the list of ASNs that they want to generate filters for in any number of ways. For example, this list can range anywhere from a hard coded set to a set derived from a provision database that specifies what ASNs to expect at which peers to the entire list of all ASNs in the global routing table. The resource certification framework that TASRS uses allows RPs to pull the entire set of provisioned ASNs, but doesn't mandate that this be the priming step for all RPs. Therefore, RPs *can* generate global filter lists, but *may* choose to specify the list they are interested in (for example for things like origin validation from just their direct customers).

TASRS' distributed Internet number resource certification system retrieves and verifies each AS' routing policy information by keying off of each ASN, and ultimately specifies the prefixes that each ASN originates. Included in the routing policy information will be any IP prefixes announceable by each AS. This binding (prefix to origin) will also be verifiable through the TASRS resource certification system.

Processes TASRS' architecture is composed of three loosely coupled processes, which act as independent control flows:

1. The *Provisioning* process' control flow is orchestrated by resource holders, and allows them to *each* control the details, timing, and operational complexities surrounding their own operational needs.

2. The *Filter Generation / Resource Verification* process' control flow is purposely decoupled from route computation process in order minimize and offset interdependent brittleness tightly coupling of the systems would introduce. While some may choose to couple these two processes directly, this design leaves that as an option, rather than a mandated conflation. It is particularly important to derive persistent policy abstraction and buffer between resource certification and operational routing within the global routing infrastructure.

3. The *Routing* process remains completely managed by the operational practices that exist today. That is, this resource certification framework does not prescribe any mandatory changes to routing. The view that securing routing is a policy issue offsets the ingestion of verification to the policy computation state (the Filter Generation process).

The goal of separating the above set of logic into separate processes (with their own control flows) is primarily to formalize the observation that each of these parts of the secure inter-domain routing and Internet number resource certification landscape is logically independent, and the interdependencies only come when verification draws an intersection. However, the separation offers the additional benefits of failure isolation (for example, filter generation failures do not halt routing), enhanced scalability (for example, not every system must deploy, be synchronized, upgraded, etc. at the same time), and operational autonomy (that is, even within a single organization, systems operators and routing operators can maintain their separate division of duties). These are discussed in greater detail, below.

4.1 Resource Provisioning

While our overall goal is to facilitate a general framework for Internet number resource certification, we start with a simplified model of Internet resource certification. Specifically, we begin by focusing on IPv4 and IPv6 prefixes, Autonomous System Numbers (ASNs), and the relationships they have in BGP routing. In today's Internet routing system, reachability to IP prefixes is asserted by ASNs. As a result, there is an inherent mapping between an ASN that reports to originate IP prefixes, and those prefixes themselves. We introduce the notion that a *forward mapping* is one in which a Relying Party (RP) uses an ASN as a lookup key to verify the set of IP prefixes it is authorized to announce. Conversely, we defined a *reverse mapping* as one where an IP prefix is used to lookup and verify the set of ASNs that are authorized to announce it. For the forward mapping process, we propose that the `.arpa` branch of DNS (the de facto infrastructure management branch) be augmented with a subtree for ASNs, called `asn.arpa`.

4.1.1 Forward Mapping

In the same vein as the IP branches of `.arpa`, `asn.arpa`'s utility must come from providing resource holders with the ability to manage their own resources and configurations. In addition to that, where possible, requirement 3 must be preserved. That is, while the main goal of `asn.arpa` is to allow RPs to lookup and discover provisioning of an ASN's information (without knowing what registry allocated the ASN), it is important that only the agency who allocated an ASN can manage that allocation. Thus, if APNIC has allocated ASN 2.99, then ARIN must not be able to revoke or reissue it. Also, the global root (IANA) should not be able to override an RIR's allocation in this way either. For this reason, `asn.arpa` should use standard dotted AS notation (as prescribed by the transition from 2-byte to 4-byte ASNs).

The dotted AS notation was initially used as a rollout mechanism for 4-byte ASNs. One fortunate side effect of this transition is that going forward ASNs can be (and are being) aggregated by the RIR that allocated them. That is, the allocation of legacy 2-byte ASNs, by the RIRs are all intermingled in irregular ranges. So, it is never clear (by inspection) which RIR allocated any ASN just by the number. By contrast, the 4-byte ASN space above the legacy max of 65,536 is all divided by RIR along octet boundaries. For example, only APNIC can allocate an ASN with the number $2.X$. Therefore, any allocated ASN above 65,536 is (automatically) immune to grandparenting. As seen in Figure 1, this would create a two tier hierarchy, where each tier would be quite manageable, at 65,536 possible delegations. Each ASN would then either be unallocated, or result in a delegation to a resource holder. As a result, clients could issue simple formatted queries for an ASN, and be delegated to that resource holder's DNS name servers. This would allow a resource holder to securely manage his/her own AS information in his/her DNS zone. This has the added operational benefit that it avoids mandating such an unmanageably large zone of roughly 4 billion ASNs (if the 4-byte space were every fully allocated). That said, the mapping of ASNs into DNS would almost certainly *not* be a scaling challenge for the foreseeable future.

For the forward mapping of routed resources, operators need to add two DNS resource record sets (RRsets) for each ASN that they have. The domain name of these RRsets is the AS dotted notation of the ASN under the `asn.arpa` zone, as seen in Figure 1. One of the two sets is a service pointer to an Internet Routing Registry (IRR), and the other is a public cryptographic key (used to verify signatures over policy statements that are extracted from the IRR instance).

One of the fundamental advantages of the service pointer is that policy changes can be effectuated through updates to the IRR repository, instead of requiring routing operators to exert control over the DNS infrastructure, or to establish operational procedures that allow them to update DNS zones at the same frequency that they may make routing policy changes. They only need DNS zone modifications when new ASNs are provisioned, or new delegation information is needed (like a new IRR or signing key must be deployed). Another fundamental advantage of this model comes from the fact that when operators need to make changes (peering, policy, etc.) no DNS modifications are needed. Rather, updating the IRR repository is all that must be done. In effect, this process follows the same operational best practices that are in use in many of today's network operation centers.

4.1.2 Reverse Mapping

After the initial service location and AS object retrieval, the validation component (filter generation) will need to ensure that the prefix/origin pairs that it has created are, indeed, valid. For this reason, TASRS needs to be able to look up this mapping in the same resource certification framework that it used to lookup ASN information. We dub the prefix lookups as being Reverse Mappings because they are back-verifying information that was derived from the Forward Mapping process (above).

The mechanism to verify authorized origins for IP prefixes is to encode the prefix as a DNS domain name, according to the rules detailed in `draft-gersch-dnsop-revdns-cidr` [8] (briefly summarized below), and then query for the DNS RR type `ROA` (described below). The goal of this approach is to allow IP resource holders (who have the right to manage their own DNS reverse delegation, RFC 2050) to authorize their own ASNs (or manage the allowed set).

4.1.3 Prefix Encoding

From Section 4.1 of [8]:

```
The CIDR to Reverse-DNS name conversion is performed as follows:

   1.  Remove any octets that are not significant.  An octet is
       significant if it includes any part of the network address.  An
       octet is not significant if all bits correspond to the host
       portion of the address.  For example, 129.82.0.0/16 --> 129.82
       and 129.82.160.0/19 --> 129.82.160

   2.  If the prefix falls on an octet boundary: first, invert the
       address and insert a "m" label as the first label to indicate
       this is a prefix name and, then, append in-addr.arpa to the end
       e.g. 129.82 --> m.82.129.in-addr.arpa.

   3.  If the prefix does not fall on an octet boundary:

       A.  Truncate the name to remove the least significant octet.  Add
           a "m" label to this domain name to indicate "mask".

       B.  Convert the least significant octet to binary, separating
           each bit into its own label (with a "." character).

       C.  Truncate the binary labels to the N significant labels that
           correspond to the given prefix\_length.

       D.  Reverse the string and add ".in-addr.arpa."

   Several examples illustrate this algorithm.  These examples show the
   conversion to binary, followed by the truncation, followed by the
   name reversal.

       129.82.0.0/16   --> m.82.129.in-addr.arpa. (at octet boundary)


       129.82.64.0/18  --> 129.82.m.0.1.0.0.0.0.0.0
                       --> 129.82.m.0.1 (N = 18 mod 8 = 2)
                       --> 1.0.m.82.129.in-addr.arpa.


       129.82.64.0/20  --> 129.82.m.0.1.0.0.0.0.0.0
                       --> 129.82.m.0.1.0.0  (N = 20 mod 8 = 4)
                       --> 0.0.1.0.m.82.129.in-addr.arpa.


       129.82.160.0/20 --> 129.82.m.1.0.1.0.0.0.0.0
                       --> 129.82.m.1.0.1.0 (N = 20 mod 8 = 4)
                       --> 0.1.0.1.m.82.129.in-addr.arpa.
```

```
129.82.160.0/23 --> 129.82.m.1.0.1.0.0.0.0.0
                --> 129.82.m.1.0.1.0.0.0.0.0 (N = 23 mod 8 = 7)
                --> 0.0.0.0.1.0.1.m.82.129.in-addr.arpa.


15.192.0.0/12   --> 15.192.m.1.1.0.0.0.0.0.0
                --> 15.192.m.1.1.0.0     (N = 12 mod 8 = 4)
                --> 0.0.1.1.m.15.in-addr.arpa.
```

And from Section 4.2 of [8]:

```
The IPv6 naming convention is similar, with the exception that 4-bit
nibble boundaries are used instead of octets, and "ip6.arpa" is used
as the suffix.

Examples:

        2607:fa88::/32     --> m.8.8.a.f.7.0.6.2.ip6.arpa
           (on nibble boundary)


        2607:fa88:8000::/33 --> 2.6.0.7.f.a.8.8.m.1.0.0.0
                            --> 2.6.0.7.d.a.8.8.m.1    (33 mod 4 = 1)
                            --> 1.m.8.8.a.f.7.0.6.2.ip6.arpa


        2607:fa88:e000::/35 --> 2.6.0.7.f.a.8.8.m.1.1.1.0
                           --> 2.6.0.7.d.a.8.8.m.1.1.1(35 mod 4 = 3)
                           --> 1.1.1.m.8.8.a.f.7.0.6.2.ip6.arpa
```

4.1.4 The ROA Resource Record

In RFC 6482 [11], the contents of Route Origin Authorization (ROA) records are described. ROAs are RPKI signed objects that are encoded in ASN.1, DER. They are defined as

```
RouteOriginAttestation ::= SEQUENCE {

     version [0] INTEGER DEFAULT 0,
     asID  ASID,
     ipAddrBlocks SEQUENCE (SIZE(1..MAX)) OF ROAIPAddressFamily }

  ASID ::= INTEGER

  ROAIPAddressFamily ::= SEQUENCE {
     addressFamily OCTET STRING (SIZE (2..3)),
     addresses SEQUENCE (SIZE (1..MAX)) OF ROAIPAddress }

  ROAIPAddress ::= SEQUENCE {
     address IPAddress,
     maxLength INTEGER OPTIONAL }

  IPAddress ::= BIT STRING
```

These objects can be represented directly as binary objects in DNS. As a result, one can use the ROA RR (temporary type number 65427) to bind a ROA binary object with each of the prefixes that are being announced that it verifies by placing it at the domain name that corresponds to each prefix.

From a ROA that is retrieved and validated through DNSSEC, RPs can use DNS' reverse mapping and the existing ROAs to cross check the policy information from the forward mapping.

4.2 Filter Generation

The filter generation process is basically the flip-side of the same coin described above. Filter generation ingests the provisioning by, first, looking for *where* a resource holder has provisioned their policy information. That means that an ASN is used as the initial lookup key into the TASRS' resource certification system. Figure 2 illustrates a high level view of how resources for a specific neighboring AS are discovered and used to create and verify filters. The two data items contained in the resource certification system for each ASN are a service pointer (a `NAPTR` record) to an IRR repository, and X.509 certificate that can be used to verify the RPSL aut-num statements for that ASN.

We note that some concern has been raised about IRRs, based on perceived historical limitations. In a recent Internet Draft [12], these limitations were cataloged, and a survey is presented that shows how many of those issues that caused IRRs to be unusable have been addressed, and that IRRs are (again) well suited to conveying policy semantics and intent between ASes, in inter-domain routing.

In TASRS, the the `NAPTR` record is used with the S-NAPTR conventions [7], and must specify the protocol and domain name of the *service* object repository that the resource holder provisions their information in. For example, if a resource holder uses an IRR for the data, they create the following entry for ASN 0:

```
$ORIGIN 0.0.asn.arpa.
IN NAPTR 100  10   "s"   "TASRS:irr"              (   ; service
                        ""                            ; regexp
                        _irr._tcp.0.0.asn.arpa.       ; replacement
                                                  )
IN NAPTR 100  20   "s"   "TASRS:whois"            (   ; service
                        ""                            ; regexp
                        _whois._tcp.0.0.asn.arpa.     ; replacement
                                                  )
```

Then `SRV` records would be used at those domain names.

With the public key (contained in the `CERT` RR type), the filter generation component queries the IRR service for an ASN's aut-num, and verifies it with an attached signature and the public key.

As in today's filter generation approach, the statement for each ASN is (ultimately) expanded to prefix/origin pairs. From these, TASRS then does reverse mapping lookups into the resource certification backend to ensure that statements have expanded to valid mappings.

Verified results are then entered into per-peer and/or per-router specific routing policies. Not all policies are global, and local filter policies can be derived this way. Specific filters are then deployed to specific routers as they fit. Policy refresh can be triggered and/or modified by operators or automated.

4.3 Routing

Business as usual!

5 Considerations

With a resource certification framework like TASRS, constructing routing policies can become automatic based on resource certification infrastructure, augment existing mechanisms, and enable additional applications such as inter-domain anti-spoofing techniques, e.g., generation of datapath filtering [3] can be automatically generated, in order to address issues such as the insertion of ASes in the AS_PATH of updates.

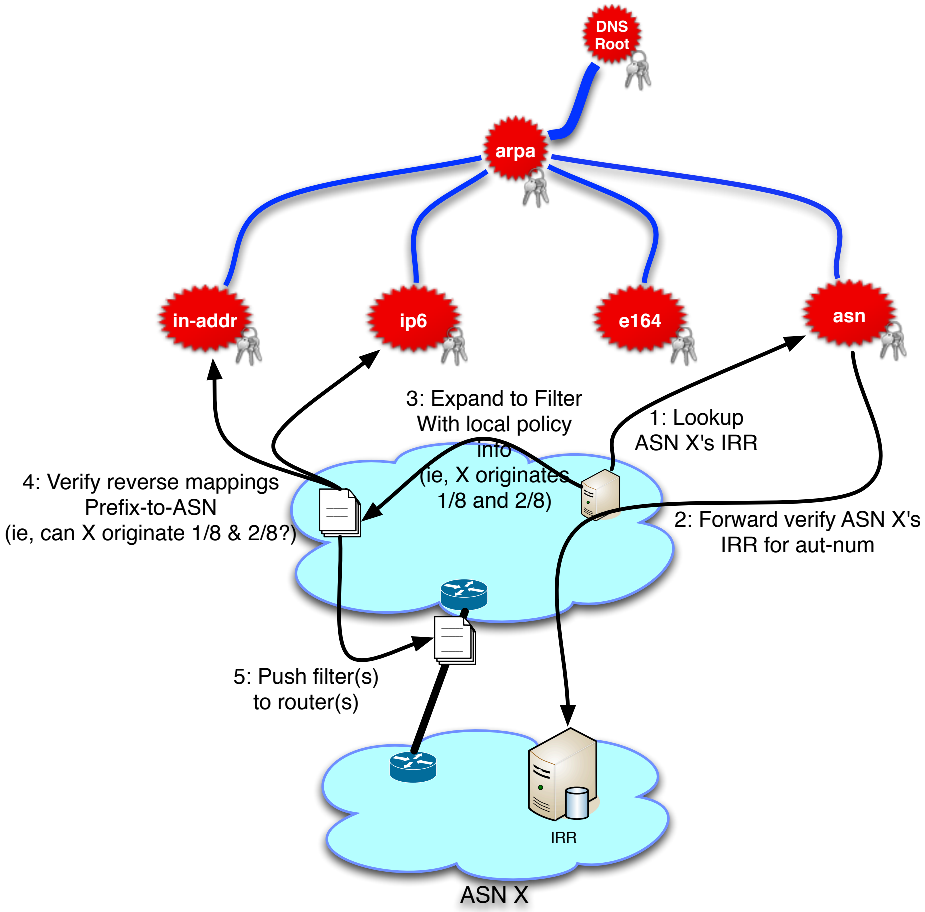
DNS Root
arpa
in-addr
ip6
e164
asn
1: Lookup ASN X's IRR
2: Forward verify ASN X's IRR for aut-num
3: Expand to Filter With local policy info (ie, X originates 1/8 and 2/8)
4: Verify reverse mappings Prefix-to-ASN (ie, can X originate 1/8 & 2/8?)
5: Push filter(s) to router(s)
IRR
ASN X

Figure 2: Filter generation within the TASRS framework.

References

[1] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. DNS Security Introduction and Requirement. RFC 4033, March 2005.

[2] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. Protocol Modifications for the DNS Security Extensions. RFC 4035, March 2005.

[3] F. Baker and P. Savola. Ingress Filtering for Multihomed Networks (BCP 84). BCP 84, March 2004.

[4] Tony Bates, Randy Bush, Tony Li, and Yakov Rekhter. Dns-based nlri origin as verification in bgp. Internet draft, Network WG, 1998.

[5] Kyle Brogle, Danny Cooper, Sharon Goldberg, and Leonid Reyzin. Impacting IP Prefix Reachability via RPKI Manipulations. Technical report, January 2013.

[6] CERT. Cert vulnerability note vu#800113, 2008.

[7] L. Daigle and A. Newton. Domain-Based Application Service Location Using SRV RRs and the Dynamic Delegation Discovery Service (DDDS). RFC 3958, January 2005.

[8] J. Gersch, D. Massey, E. Osterweil, and C. Olschanowsky. Reverse dns naming convention for cidr address blocks v03. Internet draft, Network WG, 2012.

[9] Geoff Huston and Randy Bush. Securing bgp with bgpsec. *The Internet Protocol Forum*, 14(2), june 2011.

[10] Stephen Kent, Charles Lynn, and Karen Seo. Secure border gateway protocol (S-BGP). *IEEE Journal on Selected Areas in Communications*, 18(4):582–592, 2000.

[11] M. Lepinski, S. Kent, and D. Kong. A Profile for Route Origin Authorizations (ROAs). RFC 6482, Feburary 2012.

[12] Danny McPherson, Shane Amante, Eric Osterweil Larry Blunk, and Dave Mitchell. IRR & Routing Policy Configuration Considerations. Internet draft, GROW, 2013.

[13] Danny McPherson, Shane Amante, Eric Osterweil, and Dave Mitchell. Route Leaks & MITM Attacks Against BGPSEC. Internet draft, GROW, 2013.

[14] P. Mockapetris and K. J. Dunlap. Development of the domain name system. In *SIGCOMM '88*, 1988.

[15] Eric Osterweil, Shane Amante, Danny McPherson, and Dan Massey. The Great IPv4 Land Grab: Resource Certification for the IPv4 Grey Market. In *Proc. 10th ACM Workshop on Hot Topics in Networks (Hotnets-X)*, 2011.

[16] Eric Osterweil, Terry Manderson, Russ White, and Danny McPherson. Sizing Estimates for a Fully Deployed RPKI. Technical Report 1120005, Verisign Labs Technical Report, December 2012.

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

[18] Russ White. Architecture and Deployment Considerations for Secure Origin BGP (soBGP). Internet draft, Network WG, December 2006.