            Considerations for Internet Routing Registries (IRRs)
                   and Routing Policy Configuration

Abstract

   The purpose of this document is to catalog issues that influenced the
   efficacy of Internet Routing Registries (IRRs) for inter-domain
   routing policy specification and application in the global routing
   system over the past two decades. Additionally, it provides a
   discussion regarding which of these issues are still problematic in
   practice, and which are simply artifacts that are no longer
   applicable but continue to stifle inter-provider policy-based
   filtering adoption and IRR utility to this day.

Table of Contents

1. Introduction

   The purpose of this document is to catalog issues influencing the
   efficacy of Internet Routing Registries (IRRs) for inter-domain
   routing policy specification and application in the global routing
   system over the past two decades. Additionally, it provides a
   discussion regarding which of these issues still pose problems in
   practice, and which are no longer obstacles, but whose perceived
   drawbacks continue to stifle inter-provider policy-based filtering
   support and IRR utility to this day.

2. Background

   IRRs can be used to express a multitude of Internet number bindings
   and policy objectives, i.e., to include bindings between 1) an origin
   AS and a given prefix, 2) a given AS and its AS and community import
   and export policies, as well as 3) a given AS and the AS macros (as-
   sets in Routing Policy Specification Language (RPSL)) that convey the
   set of ASes that it intends to include in some common group.

   As quoted from Section 7 of "Routing in a Multi-Provider Internet"
   [RFC1787]:

      While ensuring Internet-wide coordination may be more and more
      difficult, as the Internet continues to grow, stability and
      consistency of the Internet-wide routing could significantly
      benefit if the information about routing requirements of various
      organizations could be shared across organizational boundaries.
      Such information could be used in a wide variety of situations
      ranging from troubleshooting to detecting and eliminating
      conflicting routing requirements. The scale of the Internet
      implies that the information should be distributed. Work is
      currently underway to establish depositories of this information
      (Routing Registries), as well as to develop tools that analyze, as
      well as utilize this information.

3. Historical Artifacts Influencing IRR Efficacy

   The term IRR is often used, incorrectly, as a broad catch-all term to
   categorize issues related to the accuracy of data in the IRR, RPSL,
   and the operational deployment of policy (derived from RPSL contained
   within the IRR) to routers. It is important to classify these issues
   into distinct categories so that the reader can understand which
   categories of issues are historical artifacts that are no longer
   applicable and which categories of issues still exist and might be
   addressed by the IETF.

The following sections will separate out challenges related to the
IRR into the following categories: first, accuracy and integrity of
data contained within the IRR; second, operation of the IRR
infrastructure, i.e., synchronization of resources from one IRR to
other IRRs; and finally, this document covers the methods related to
extraction of policy from the IRR and the input, plus activation of
that policy within routers.

4. Accuracy and Integrity of Data Contained within the IRR

   The following section will examine issues related to accuracy and
   integrity of data contained within the IRR.

4.1. Lack of Resource Certification

   Internet number resources include IPv4 addresses, IPv6 addresses,
   Autonomous System Numbers (ASNs), and more. While these resources
   are generally allocated by hierarchical authorities, a general
   mechanism for formally verifying (such as through cryptographic
   mechanisms) when parties have been allocated resources remains an
   open challenge. We generally call such a system a Resource
   Certification System, and we note that some candidate examples of how
   such a general system might be implemented and deployed exist --
   [TASRS], [RC_HotNetsX], and [RFC6480].

   One of the largest weaknesses often cited with the IRR system is that
   the data contained within the IRRs is out of date or lacks integrity.
   This is largely attributable to the fact that existing IRR mechanisms
   do not provide ways for a relying party to (cryptographically) verify
   the validity of an IRR object. That is, there has never existed a
   resource certification infrastructure that enables a resource holder
   to authorize a particular autonomous system to originate network-
   layer reachability advertisements for a given IPv4 or IPv6 prefix.
   It should be noted that this is not a weakness of the underlying RPSL
   [RFC2622], but rather, was largely the result of no clear demand by
   the operator community for Internet Number Resource Registries to
   provide sufficient resource certification infrastructure that would
   enable a resource holder to develop a cryptographic binding between,
   for example, a given AS number and an IP prefix.

   Another noteworthy (but slightly different) deficiency in the IRR
   system is the absence of a tangible tie between the resource and the
   resource holder. That is, generally there is no assurance of the
   validity of objects at their creation time (except for a subset of,
   for example, the RIPE IRR where RPSS [RFC2725] attests for RIPE
   address holders and RIPE ASN holders). If a resource holder's
   authorization cannot be certified, then consumers cannot verify
   attestations made. In effect, without resource certification,

consumers are basically only certifying the assertions that the
creator/maintainer of the resource object has made (not if that
assertion is valid).

The RIPE community addressed this last issue by putting in a
foundation policy [RIPE638], which requires a contractual link
between the RIPE NCC and the end user in direct assignment + ASN
assignment cases, which weren't previously covered by Local Internet
Registry (LIR) contracts for address allocations. There were a
couple of intentions with this policy:

1. There was an encumbrance placed in the policy for the LIR to
    charge the end user for provider-independent (PI) resources.
    This action created a collection mechanism for PI address
    resources (IPv4/IPv6 space, ASNs).

2. It guaranteed that all RIPE NCC allocated/assigned space would be
    subject to a contractual link, and that this contractual chain
    might end up actually meaning something when it came to the issue
    of who made what claim about what number resource.

3. It tied into the RIPE NCC's object grandfathering policy that
    ties the registration details of the end user to the object
    registered in the IRR database.

While this policy specifically addressed PI/portable space holders,
other regions address this issue, too. Further, a tangible tie
between the resource and the resource holder is indeed a prerequisite
for resource certification, though it does not directly address the
IRR deficiencies.

One of the central observations of this policy was that without a
chain-of-ownership functionality in IRR databases, the discussion of
certifying their contents becomes moot.

4.2. Incentives to Maintain Data within the IRR

A second problem with data contained in the IRRs is that the
incentives for resource holders to maintain both accurate and up-to-
date information in one or more IRRs (including deletion of out-of-
date or stale data from the IRRs) can diminish rapidly when changing
their peering policies (such as switching transit providers).
Specifically, there is a very strong incentive for an ISP's customers
to register new routing information in the IRR, because some ISPs
enforce a strict policy that they will only build or update a
customer's prefix-lists applied to the customer's inbound eBGP
sessions based off information found within the IRRs. Unfortunately,
there is little incentive for an ISP's customers to remove out-of-

date information from an IRR, most likely attributed to the fact that
some ISPs do not use, or enforce use of, data contained within the
IRRs to automatically build incoming policy applied to the customer's
eBGP sessions. For example, if a customer is terminating service
from one ISP that requires use of IRR data to build incoming policy
and, at the same time, enabling service with another ISP that does
not require use of IRR data, then that customer will likely let the
data in the IRR become stale or inaccurate.

Further, policy filters are almost exclusively generated based on the
origin AS information contained within IRR route objects and used by
providers to filter downstream transit customers. Since providers
only look for route objects containing the origin AS of their
downstream customer(s), stale route objects with historical and,
possibly, incorrect origin AS information are ignored. Assuming that
the downstream customer(s) do not continue to announce those routes
with historical, or incorrect, origin AS information in BGP to the
upstream provider, there is no significant incentive to remove them
as they do not impact offline policy filter generation nor routing on
the Internet. On the other hand, the main incentive that causes the
Service Provider to work with downstream customer(s) is when the
resultant filter list becomes so large that it is difficult for it to
be stored on PE routers; however, this is more practically an
operational issue with very old, legacy PE routers, not more modern
PE router hardware with more robust control-plane engines.

4.3. Inability for Third Parties to Remove (Stale) Information from the
      IRRs

A third challenge with data contained in IRRs is that it is not
possible for IRR operators, and ISPs who use them, to proactively
remove (perceived) out-of-date or "stale" resources in an IRR, on
behalf of resource holders who may not be diligent in maintaining
this information themselves. The reason is that, according to the
RPSL [RFC2622], only the resource holder ('mntner') specified in a
'mnt-by' value field of an RPSL resource is authorized to add,
modify, or delete their own resources within the IRR. To address
this issue, the 'auth-override' mechanism [RFC2725] was later
developed that would have enabled a third party to update and/or
remove "stale" resources from the IRR. While it is unclear if this
was ever implemented or deployed, it does provide language semantics
needed to overcome this obstacle.

Nevertheless, with such a mechanism in place, there is still a risk
that the original RPSL resource holder would not receive
notifications (via the 'notify' attribute in various RPSL resources)
about the pending or actual removal of a resource from the IRR in
time to halt that action if those resources were still being used.

In this case, if the removal of a resource was not suspended, it
could potentially result in an unintentional denial of service for
the RPSL resource holder when, for example, ISPs automatically
generated and deployed a new policy based on the newly removed
resources from the IRR, thus dropping any reachability announcement
for the removed resource in eBGP.

4.4. Lack of Authoritative IRR for Resources

According to [RFC2622], within an RPSL resource "the source attribute
specifies the registry where the object is registered." Note that
this source attribute only exists within the RPSL resource itself.
Unfortunately, given a specific resource (e.g., a specific IPv4 or
IPv6 prefix), most of the time it is impossible to determine a priori
the authoritative IRR where to query and retrieve an authoritative
copy of that resource.

This makes it difficult for consumers of data from the IRR to
automatically know the authoritative IRR of a resource holder that
will contain the most up-to-date set of resources. This is typically
not a problem for an ISP that has a direct (customer) relationship
with the resource holder, because the ISP will ask the resource
holder which (authoritative) IRR to pull their resources from on, for
example, a "Customer BGP Order Form". However, third parties that do
not have a direct relationship with the resource holder have a
difficult time attempting to infer the authoritative IRR, preferred
by the resource holder, that likely contains the most up-to-date set
of resources. As a result, it would be helpful for third parties if
there were a robust referral mechanism so that a query to one IRR
would be automatically redirected toward the authoritative IRR for
the most up-to-date and authoritative copy of that particular
resource. This problem is worked around by individual IRR operators
storing a local copy of other IRRs' resources, through periodic
mirroring, which allows the individual IRR to respond to a client's
query with all registered instances of a particular IRR resource that
exist in both the local IRR and all other IRRs. Of course, the
problem with this approach is that an individual IRR operator is
under no obligation to mirror all other IRRs and, in practice, some
IRRs do not mirror the resources from all other IRRs. This could
lead to the false impression that a particular resource is not
registered or maintained at a particular IRR. Furthermore, the
authentication process of accepting updates by any given IRR may or
may not be robust enough to overcome impersonation attacks. As a
result, there is no rigorous assurance that a mirrored RPSL statement
was actually made by the authorized resource holder.

4.5. Client-Side Considerations

   There are no provisions in the IRR mode for ensuring the
   confidentiality component for clients issuing queries. The overall
   Confidentiality, Integrity, and Availability (CIA) model of the
   system does lack this component, because the interface to IRRs is
   over an unencrypted TCP connection to port 43. This leaves the
   transaction open to inspection such that an adversary could be able
   to inspect the query and the response. However, the IRR system is
   intended to be composed of public policy information, and protection
   of queries was not part of the protection calculus when it was
   designed, though the use of Transport Layer Security (TLS) [RFC5246]
   would address protections of query information.

4.6. Conclusions with Respect to Data in the IRR

   All of the aforementioned issues related to integrity and accuracy of
   data within the IRR stem from a distinct lack of resource
   certification for resources contained within the IRR. Only now is an
   experimental testbed that reports to provide this function (under the
   auspices of the Resource PKI [RFC6480]) being formally discussed;
   this could also aid in certification of resources within the IRR. It
   should be noted that the RPKI is only currently able to support
   signing of Route Origin Authorization (ROA) resources that are the
   equivalent of 'route' resources in the IRR. There has been some
   sentiment that the RPKI currently is not scoped to address the same
   set of issues and the nuanced policy applications that providers
   leverage in RPSL. It is unclear if, in the future, the RPKI will be
   extended to support additional resources that already exist in the
   IRR, e.g., aut-num, as-net, route-set, etc. Finally, a seemingly
   equivalent resource certification specification for all resources in
   the IRR has already been developed [RFC2725]; however, it is unclear
   how widely it was ever implemented or deployed.

5. Operation of the IRR Infrastructure

5.1. Replication of Resources among IRRs

   Currently, several IRRs [IRR_LIST] use a Near-Real-Time Mirroring
   (NRTM) protocol to replicate each other's contents. However, this
   protocol has several weaknesses. Namely, there is no way to validate
   that the copy of mirrored source is correct, and synchronization
   issues have often resulted. Furthermore, the NRTM protocol does not
   employ any security mechanisms. The NRTM protocol relies on a pull
   mechanism and is generally configured with a poll interval of 5 to 10
   minutes. There is currently no mechanism to notify an IRR when an
   update has occurred in a mirrored IRR so that an immediate update can
   be made.

Some providers employ a process of mirroring an instance of an IRR
that involves downloading a flat text file copy of the IRR that is
made available via FTP [RFC959]. These FTP files are exported at
regular intervals of typically anywhere between 2 and 24 hours by the
IRRs. When a provider fetches those text files, it will process them
to identify any updates and reflect changes within its own internally
maintained database. The use of an internally maintained database is
out of scope for this document but is generally used to assist with
more straightforward access to or modification of data by the IRR
operator. Providers typically employ a 24-hour cycle to pull updated
resources from IRRs. Thus, depending on when resource holders
submitted their changes to an IRR, it may take up to 24 hours for
those changes to be reflected in their policy configurations. In
practice, it appears that the RPKI will also employ a 24-hour cycle
whereby changes in resources are pushed out to other RPKI caches
[RPKI_SIZING].

IRRs originated from Section 7 of [RFC1787], specifically: "The scale
of the Internet implies that the [routing requirements] information
should be distributed." Regardless, the practical effect of an
organization maintaining its own local cache of IRR resources is an
increase in resource resiliency (due to multiple copies of the same
resource being geographically distributed), a reduction in query time
for resources, and, likely, a reduction in inter-domain bandwidth
consumption and associated costs. This is particularly beneficial
when, for example, an ISP is evaluating resources in an IRR just to
determine if there are any modifications to those resources that will
ultimately be reflected in a new routing policy that will get
propagated to (edge) routers in the ISP's network. Cache locality
results in reduced inter-domain bandwidth utilization for each round
trip.

On the other hand, it is unclear from where the current provider
replication interval of 24 hours originated or even whether it still
provides enough freshness in the face of available resources,
particularly in today's environment where a typical IRR system
consists of a (multi-core) multi-GHz CPU connected to a network via a
physical connection of 100 Mbps or, more likely, higher bandwidth.
In addition, due to demand for bandwidth, circuit sizes used by ISPs
have increased to 10 Gbps, thus eliminating bandwidth as a
significant factor in the transfer of data between IRRs.
Furthermore, it should be noted that Merit's Internet Routing
Registry Daemon (IRRd) [MERIT-IRRD] uses 10 minutes as its default
for "irr_mirror_interval".

Lastly, it should be noted that "Routing Policy System Replication"
[RFC2769] attempted to offer a more methodical solution for
distributed replication of resources between IRRs. It is unclear why

that RFC failed to gain traction, but it is suspected that this was
due to its reliance on "Routing Policy System Security" [RFC2725],
which addressed "the need to assure integrity of the data by
providing an authentication and authorization model." Indeed,
[RFC2725] attempts to add an otherwise absent security model to the
integrity of policy statements made in RPSL. Without formal
protections, it is possible for anyone to author a policy statement
about an arbitrary set of resources, and publish it (as discussed
above in Section 4.1.

5.2. Updating Routing Policies from Updated IRR Resources

   Ultimately, the length of time it takes to replicate resources among
   IRRs is, generally, the dominant factor in reflecting changes to
   resources in policy that is eventually applied within the control
   plane of routers. The length of time to update network elements will
   vary considerably depending on the size of the ISP and the number of
   IRR resources that were updated during any given interval. However,
   there are a variety of common techniques, that are outside the scope
   of this document, that allow for this automated process to be
   optimized to considerably reduce the length of time it takes to
   update policies in the ISP's network.

   An ISP will begin the process of updating the policy in its network,
   first by fetching IRR resources associated with, for example, a
   customer ASN attached to its network. Next, the ISP constructs a new
   policy associated to that customer and then evaluates if that new
   policy is different from existing policy associated with that same
   customer. If there are no changes between the new and existing
   policy associated with that customer, then the ISP does not make any
   changes to the policy in their routers specific to that customer. On
   the other hand, if the new policy does reflect changes from the
   existing policy for that customer, then the ISP begins a process of
   uploading the new policy to the routers attached to that customer.

   The process of constructing a new policy involves use of a set of
   programs, e.g., IRRtoolset, that performs recursive expansion of an
   RPSL aut-num resource that comprises an arbitrary combination of
   other RPSL aut-num, as-set, route, and route-set resources, according
   to procedures defined by RPSL. The end result of this process is,
   traditionally, a vendor-dependent configuration snippet that defines
   the routing policy for that customer. This routing policy may
   consist of the set of IPv4 or IPv6 prefixes, associated prefix
   lengths, and AS_PATHs that are supposed to be accepted from a
   customer's eBGP session. However, if indicated in the appropriate
   RPSL resource, the policy may also set certain BGP Attributes, such

as MED, AS_PATH prepend value, LOCAL_PREF, etc., at either the
incoming eBGP session from the customer or on static routes that are
originated by the resource holder.

An ISP's customers may not adequately plan for pre-planned
maintenance, or, more likely, they may need to rapidly begin
announcing a new IP prefix as a result of, for example, an emergency
turn-up of the ISP customer's new downstream customer.
Unfortunately, the routine, automated process employed by the ISP
means that it may not begin updating its routing policy on its
network for up to 24 hours, because the ISP or the IRRs the ISP uses
might only mirror changes to IRR resources once every 24 hours. The
time interval for the routine/automated process is not responsive to
the needs of directly paying customer(s) who need rapid changes in
their policy in rare situations. In these situations, when a
customer has an urgent need for updates to take effect immediately,
they will call the Network Operations Center (NOC) of their ISP and
request that the ISP immediately fetch new IRR objects and push those
changes out to its network. This is often accomplished in as little
as 5 minutes from the time a customer contacts their ISP's NOC to the
time a new filtering policy is pushed out to the Provider Edge (PE)
routers that are attached to that customer's Attachment Circuits
(ACs). It is conceivable that some ISPs automate this using out-of-
band mechanisms as well, although the authors are unaware of any
existing mechanisms that support this.

Ultimately, the aforementioned latency with respect to "emergency
changes" in IRR resources that need to be reflected in near-real-time
in the network is compounded if the IRR resources were being used by
third-party ISPs to perform filtering on their peering circuits,
where typically no such policies are employed today for this very
reason. It is likely that the length of time that it takes IRRs to
mirror changes will have to be dramatically reduced. There will need
to be a corresponding reduction in the time required by ISPs to
evaluate whether those changes should be recompiled and reflected in
router policies that would then get pushed out to Autonomous System
Border Routers (ASBRs) connected to peering circuits on their
network.

6. Historical BGP Protocol Limitations

As mentioned previously, after a resource holder made changes to
their resources in an IRR, those changes would automatically be
distributed to other IRRs, ISPs would then learn of those changes,
generate new BGP policies, and then apply those to the appropriate
subset of routers in their ASes. However, in the past, one
additional step is necessary in order to have any of those new BGP
policies take effect in the control plane and to allow/deny the

updated resource from a customer to their ISP and from their
immediately upstream ISP to the ISP's peers. It was necessary (often
manually) to actually induce BGP on each router to apply the new
policy within the control plane, thus learning of a newly announced/
changed IP prefix (or, dropping a deleted IP prefix). Unfortunately,
most of these methods not only were highly impactful operationally,
but they also affected traffic forwarding to IP destinations that
were unrelated to the most recent changes to the BGP policy.

Historically, a customer would have to (re-)announce the new IP
prefix toward their ISP, but only after the ISP had put the new BGP
policies into effect. Alternatively, the ISP would have to reset the
entire eBGP session from Provider Edge to Customer Edge either by: a)
bouncing the entire interface toward the customer (e.g., shutdown /
no shutdown) or b) clearing the eBGP session toward the customer
(e.g., clear ip bgp neighbor <IP address of CE router>, where <IP
address of CE router> represents a specific IP address). The latter
two cases were, of course, the most highly impactful impact and thus
could generally only be performed off-hours during a maintenance
window.

Once the new IP prefix has been successfully announced by the
customer and permitted by the newly updated policy at the ISP's PEs
(attached to that customer), it would be propagated to that ISP's
ASBRs, attached to peers, at the perimeter of the ISP network.
Unfortunately, if those peers had either not yet learned of the
changes to resources in the IRR or pushed out new BGP policies (and,
reset their BGP sessions immediately afterward) incorporating those
changes, then the newly announced route would also get dropped at the
ingress ASBRs of the peers.

Ultimately, either of the two scenarios above further lengthens the
effective time it would take for changes in IRR resources to take
effect within BGP in the network. Fortunately, BGP has been enhanced
over the last several years in order that changes within BGP policy
will take effect without requiring a service-impacting reset of BGP
sessions. Specifically, BGP soft-reconfiguration (Section 1 of
[RFC2918]) and, later, Route Refresh Capability for BGP-4 [RFC2918]
were developed so that ISPs, or their customers, could induce BGP to
apply a new policy while leaving both the existing eBGP session
active as well as (unaffected) routes active in both the Loc-RIB and,
more importantly, FIB of the router. Thus, using either of these
mechanisms, an ISP or its peers currently will deploy a newly
generated BGP policy, based on changes to resources within the IRR,
and immediately trigger a notification -- which does not impact
service -- to the BGP process to have those changes take effect in
their control plane, either allowing a new IP prefix to be announced
or an old IP prefix to be dropped. This dramatically reduces the

length of time from when changes are propagated throughout the IRRs
to when those changes are actually operational within BGP policy in
an ISP's network.

7. Historical Limitations of Routers

7.1. Incremental Updates to Policy on Routers

   Routers in the mid 1990s rarely supported incrementally updatable
   prefix filters for BGP; therefore, if new information was received
   from a policy or internal configuration database that would impact a
   policy applied to a given eBGP peer, the entire prefix list or access
   list would need to be deleted and rewritten, compiled, and installed.
   This was very tedious and commonly led to leaked routes during the
   time when the policy was being rewritten, compiled, and applied on a
   router. Furthermore, application of a new policy would not
   automatically result in new ingress or egress reachability
   advertisements from that new policy, because routers at the time
   would require a reset of the eBGP sessions for routing information to
   be evaluated by the new policy. Of course, resetting of an eBGP
   session had implications on traffic forwarding during the time the
   eBGP session was reestablished and new routing information was
   learned.

   Routers now support the ability to perform incremental, and in situ,
   updates to filter lists consisting of IP prefixes and/or AS_PATHs
   that are used within an ingress or egress BGP policy. In addition,
   routers also can apply those incremental updates to policy, with no
   traffic disruption, using BGP soft-reconfiguration or BGP Route
   Refresh, as discussed in the previous section.

7.2. Storage Requirements for Policy on Routers

   Historically, routers had very limited storage capacity and would
   have difficulty in storing an extremely large BGP policy on-board.
   This was typically the result of router hardware vendors using an
   extremely limited amount of NVRAM for storage of router
   configurations.

   Another challenge with historical router hardware was that writing to
   NVRAM was extremely slow. For example, when the router configuration
   had changed as a result of updating a BGP policy that needed to
   accommodate changes in IRR resources, this would result in extremely
   long times to write out these configuration changes. Sometimes, due
   to bugs, this would result in loss of protocol keep-alives. This
   would cause an impact to routing or forwarding of packets through the
   platform.

The above limitations have largely been resolved with equipment from
the last few years that ships with increasing amounts of non-volatile
storage such as PCMCIA or USB flash cards, hard disk drives, or
solid-state disk drives.

However, as capacities and technologies have evolved on modern
routing hardware, so have some of the scaling requirements of the
data. In some large networks, configuration growth has begun to
"pose challenges" [IEPG89_NTT]. While the enhancements of hardware
have overcome some historical limitations, evidence suggests that
further optimizations in configuration processing may be needed in
some cases. Some of the more recent operational issues include
scheduler slips and protracted commit times. This suggests that even
though many historical hurdles have been overcome, there are still
motivations to optimize and modernize IRR technologies.

7.3. Updating Configuration on Routers

Historically, there has not been a standardized modeling language for
network configuration or an associated method to update router
configurations. When an ISP detected a change in resources within
the IRR, it would fashion a vendor-dependent BGP policy and upload
that to the router usually via the following method.

First, an updated BGP policy configuration snippet is generated via
processes running on an out-of-band server. Next, the operator uses
either telnet or SSH [RFC4253] to log in to the CLI of a target
router and issue vendor-dependent CLI commands that will trigger the
target router to fetch the new configuration snippet via TFTP, FTP,
or Secure Copy (SCP) stored on the out-of-band server. The target
router will then perform syntax checking on that configuration
snippet and, if that passes, merge that configuration snippet into
the running configuration of the router's control software. At this
point, the new BGP policy configuration snippet is active within the
control plane of the router. One last step remains -- the operator
will issue a CLI command to induce the router to perform a "soft
reset", via BGP soft-reconfiguration or BGP Route Refresh, of the
associated BGP session in order to trigger the router to apply the
new policy to routes learned from that BGP session without disrupting
traffic forwarding.

More recently, operators have the ability to use NETCONF [RFC6241] /
SSH (or, TLS) from an out-of-band server to push a BGP configuration
snippet from an out-of-band server toward a target router that has
that capability. However, this activity is still dependent on
generating, via the out-of-band server, a vendor-dependent XML
configuration snippet that would get uploaded via SSH or TLS to the
target router.

   In the future, the ability to upload new Route Origin Authorization
   (ROA) information may be provided from the RPKI to routers via the
   RPKI-RTR [RFC6810] protocol. However, this will not allow operators
   the ability to upload other configuration information such as BGP
   policy information (AS_PATHs, BGP communities, etc.) that might be
   associated with that ROA information, as they can from IRR-generated
   BGP policies.

8. Summary

   As discussed above, many of the problems that have traditionally
   stifled IRR deployment have, themselves, become historical. However,
   there are still real operational considerations that limit IRR usage
   from realizing its full effectiveness. The potential for IRRs to
   express inter-domain routing policy, and to allow relying parties to
   learn policy, has always positioned them as a strong candidate to aid
   the security postures of operators. However, while routing density
   and complexity have grown, so have some of the challenges facing IRRs
   (even today). Because of this state increase, the potential to model
   all policies for all ASes in all routers may still remain illusive.
   In addition, without an operationally deployed resource certification
   framework that can tie policies to resource holders, there is a
   fundamental limitation that still exists.

9. Security Considerations

   One of the central concerns with IRRs is the ability of an IRR
   operator to remotely influence the routing operations of an external
   consumer. Specifically, if the processing of IRR contents can become
   burdensome, or if the policy statements can be crafted to introduce
   routing problems or anomalies, then operators may want to be
   circumspect about ingesting contents from external parties. A
   resource certification framework should be used to address the
   authorization of IRR statements to make attestations and assertions
   (as mentioned in Section 4.1, and discussed in Section 5.1).

   Additionally, the external and systemic dependencies introduced by
   IRRs and other such systems employed to inform routing policy, and
   how tightly or loosely coupled those systems are to the global
   routing system and operational networks, introduce additional vectors
   that operators and system architects should consider when evaluating
   attack surface and service dependencies associated with those
   elements. These attributes and concerns are certainly not unique to
   IRRs, and operators should evaluate the implications of external
   systems and the varying degrees of coupling and operational buffers
   that might be installed in their environments.

10. Informative References

   [IEPG89_NTT]
              Mauch, J., "NTT BGP Configuration Size and Scope", IEPG
              meeting before IETF 89, March 2014,
              <http://iepg.org/2014-03-02-ietf89/
              ietf89_iepg_jmauch.pdf>.

   [IRR_LIST] Merit Network, Inc., "List of Routing Registries",
              <http://www.irr.net/docs/list.html>.

   [MERIT-IRRD]
              Merit, "IRRd - Internet Routing Registry Daemon",
              <http://www.irrd.net>.

   [RC_HotNetsX]
              Osterweil, E., Amante, S., Massey, D., and D. McPherson,
              "The Great IPv4 Land Grab: Resource Certification for the
              IPv4 Grey Market", DOI 10.1145/2070562.2070574,
              <http://dl.acm.org/citation.cfm?id=2070574>.

   [RFC959]   Postel, J. and J. Reynolds, "File Transfer Protocol",
              STD 9, RFC 959, DOI 10.17487/RFC0959, October 1985,
              <http://www.rfc-editor.org/info/rfc959>.

   [RFC1787] Rekhter, Y., "Routing in a Multi-provider Internet",
              RFC 1787, DOI 10.17487/RFC1787, April 1995,
              <http://www.rfc-editor.org/info/rfc1787>.

   [RFC2622] Alaettinoglu, C., Villamizar, C., Gerich, E., Kessens, D.,
              Meyer, D., Bates, T., Karrenberg, D., and M. Terpstra,
              "Routing Policy Specification Language (RPSL)", RFC 2622,
              DOI 10.17487/RFC2622, June 1999,
              <http://www.rfc-editor.org/info/rfc2622>.

   [RFC2725] Villamizar, C., Alaettinoglu, C., Meyer, D., and S.
              Murphy, "Routing Policy System Security", RFC 2725,
              DOI 10.17487/RFC2725, December 1999,
              <http://www.rfc-editor.org/info/rfc2725>.

   [RFC2769] Villamizar, C., Alaettinoglu, C., Govindan, R., and D.
              Meyer, "Routing Policy System Replication", RFC 2769,
              DOI 10.17487/RFC2769, February 2000,
              <http://www.rfc-editor.org/info/rfc2769>.

   [RFC2918] Chen, E., "Route Refresh Capability for BGP-4", RFC 2918,
              DOI 10.17487/RFC2918, September 2000,
              <http://www.rfc-editor.org/info/rfc2918>.

   [RFC4253] Ylonen, T. and C. Lonvick, Ed., "The Secure Shell (SSH)
              Transport Layer Protocol", RFC 4253, DOI 10.17487/RFC4253,
              January 2006, <http://www.rfc-editor.org/info/rfc4253>.

   [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security
              (TLS) Protocol Version 1.2", RFC 5246,
              DOI 10.17487/RFC5246, August 2008,
              <http://www.rfc-editor.org/info/rfc5246>.

   [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed.,
              and A. Bierman, Ed., "Network Configuration Protocol
              (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011,
              <http://www.rfc-editor.org/info/rfc6241>.

   [RFC6480] Lepinski, M. and S. Kent, "An Infrastructure to Support
              Secure Internet Routing", RFC 6480, DOI 10.17487/RFC6480,
              February 2012, <http://www.rfc-editor.org/info/rfc6480>.

   [RFC6810] Bush, R. and R. Austein, "The Resource Public Key
              Infrastructure (RPKI) to Router Protocol", RFC 6810,
              DOI 10.17487/RFC6810, January 2013,
              <http://www.rfc-editor.org/info/rfc6810>.

   [RIPE638] RIPE NCC, "Autonomous System (AS) Number Assignment
              Policies", March 2015,
              <https://www.ripe.net/publications/docs/ripe-638>.

   [RPKI_SIZING]
              Osterweil, E., Manderson, T., White, R., and D. McPherson,
              "Sizing Estimates for a Fully Deployed RPKI", Verisign
              Labs Technical Report 1120005 version 2, December 2012,
              <http://techreports.verisignlabs.com/
              tr-lookup.cgi?trid=1120005&rev=2>.

   [TASRS]   Osterweil, E., Amante, S., and D. McPherson, "TASRS:
              Towards a Secure Routing System Through Internet Number
              Resource Certification", Verisign Labs Technical
              Report 1130009, February 2013,
              <http://techreports.verisignlabs.com/
              tr-lookup.cgi?trid=1130009&rev=1>.

Authors' Addresses

   Danny McPherson
   Verisign, Inc.

   Email: dmcpherson@verisign.com

   Shane Amante
   Apple, Inc.

   Email: amante@apple.com

   Eric Osterweil
   Verisign, Inc.

   Email: eosterweil@verisign.com

   Larry J. Blunk
   Merit Network, Inc.

   Email: ljb@merit.edu

   Dave Mitchell
   Singularity Networks

   Email: dave@singularity.cx