

Network Working Group
Internet-Draft
Intended status: Informational
Expires: January 9, 2014

G. Huston
G. Michaelson
APNIC
July 8, 2013

RPKI Validation Reconsidered
draft-huston-rpki-validation-00.txt

Abstract

This document reviews the certificate validation procedure specified in RFC6487 and highlights aspects of operational management of certificates in the RPKI in response to the movement of resources across registries, and the associated actions of Certification Authorities to maintain certification of resources during this movement. The document describes an alternative validation procedure that reduces the operational impact of certificate management during resource movement.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 9, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

Table of Contents

1. Introduction	3
2. Operational Considerations	4
3. A Specific Resource RPKI Certificate Validation Process	6
3.1. Resource Transfers and Specific Resource Certificate Validation	7
3.2. A Specification of Specific Resource Validation	8
4. Local Repository Cache Maintenance	10
5. Security Considerations	10
6. IANA Considerations	11
7. Acknowledgements	11
8. References	11
8.1. Normative References	11
8.2. Informative References	11
Authors' Addresses	11

1. Introduction

This document reviews the certificate validation procedure specified in [RFC6487] and highlights aspects of operational management of certificates in the RPKI in response to the movement of resources across registries, and the associated actions of Certification Authorities to maintain certification of resources during this movement. The document describes an alternative validation procedure that reduces the operational impact of certificate management during resource movement.

As currently defined in section 7.2 of [RFC6487], validation of PKIX certificates that conform to the RPKI profile relies on the use of a path validation process where each certificate in the validation path is required to meet the certificate validation criteria. This can be considered to be a recursive validation process where, in the context of an ordered sequence of certificates, as defined by common Issuer and Subject Name pairs, a certificate is defined as valid if it satisfies basic validation criteria relating to the syntactic correctness, currency of validity dates and similar properties of the certificate itself, as described in [RFC5280], and also that it satisfies certain additional criteria with respect to the previous certificate in the sequence, and that this previous certificate is itself a valid certificate using the same criteria. This definition applies recursively to all certificates in the sequence apart from the initial sequence element, which is required to be a Trust Anchor.

For RPKI certificates, the additional criteria relating to the previous certificate in this sequence is that the certificate's number resource set, as defined in [RFC3779], is "encompassed" by the number resource set contained in the previous certificate.

Because [RFC6487] validation demands that all resources in a certificate be valid under the parent (and recursively, to the root), a digitally signed attestation, such as a Route Origin Authorization (ROA) object [RFC6482], which refers only to a subset of RFC3779-specified resources from that certificate chain can be concluded to be invalid, but not by virtue of the relationship between the RFC3779 extensions of the certificates on the putative certificate validation path and the resources in the ROA, but by other resources described in these certificates where the "encompassing" relationship of the resources does not hold. Any such invalidity along the certificate validation path can cause this outcome, not just at the immediate parent of the end entity certificate that attests to the key used to sign the ROA.

For example, in the certificate sequence:

Certificate 1:

Issuer A, Subject B, Resources 192.0.2.0/24, AS64496-AS64500

Certificate 2:

Issuer B, Subject C, Resources 192.0.2.0/24/24, AS64496-AS64511

Certificate 3:

Issuer C, Subject D, Resources 192.0.2.0/24

Certificate 3 is considered to be an invalid certificate, because the resources in Certificate 2 are not encompassed by the resources in Certificate 1, by virtue of certificate 2 holding the resources of the range AS64501 - AS64511 in this RFC3779 resource extension. Obviously, these Autonomous Systems numbers are not related to the IPv4 resources contained in Certificate 3.

2. Operational Considerations

The operational consideration described here relates to the situation where a registry withdraws a resource from the current holder, and the resource is transferred to another registry, to be registered to a new holder in that registry. The reason why this is a consideration in operational deployments of the RPKI lies in the movement of the "home" registry of number resources during cases of mergers, acquisitions, business re-alignments, and resource transfers and the desire to ensure that during this movement all other resources can continue to be validated.

If the original registry's certification actions are simply to issue a new certificate for the current holder with a reduced resource set, and to revoke the original certificate, then there is a distinct possibility of encountering the situation illustrated by the example in the previous section. This is a result of an operational process for certificate issuance by the parent CA being de-coupled from the certificate operations of child CA.

This de-coupled operation of CAs introduces a risk of unintended third party damage: since a CA certificate can refer to holdings which relate to two or more unrelated subordinate certificates, if this CA certificate becomes invalid due to the reduction in the resources allocated to this CA relating to one subordinate resource set, all other subordinate certificates are invalid until the CA certificate is reissued with a reduced resource set.

In the above example, all subordinate certificates issued by CA C are invalid until CA B issues a new certificate for CA C with a reduced resource set.

At the lower levels of the RPKI hierarchy the resource sets affected by such movements of resources may not encompass significantly large pools of resources. However, as one ascends through this hierarchy towards the apex, the larger the resource set that is going to be affected by a period of invalidity by virtue of such uncoordinated certificate management actions. In the case of a Regional Internet Registry (RIR) or National Internet Registry (NIR), the potential risk arising from uncoordinated certification actions relating to a transfer of resources is that the entire set of subordinate certificates that refer to resources administered by the RIR or the NIR cannot be validated during this period.

Avoiding such situations requires that CA's adhere to a very specific ordering of certificate issuance. In this framework, the common registry CA that describes (directly or indirectly) the resources being shifted from one registry to the other, and also contains in subordinate certificates (direct or indirect) the certificates for both registries who are parties to the resource transfer has to coordinate a specific sequence of actions.

This common registry CA has to first issue a new certificate towards the "receiving" registry that adds to the RFC3779 extension resource set the specific resource being transferred into this receiving registry. The common registry CA then has to wait until all registries in the subordinate certificate chain to the receiving registry have also performed a similar issuance of new certificates, and in each case a registry must await the issuance of the immediate superior certificate with the augmented resource set before it, in turn, can issue its own augmented certificate to its subordinate CA. This is a "top down" issuance sequence."

It is possible for the common registry to issue a certificate to the "sending" registry with the reduced resource set at any time, but it should not revoke the previously issued certificate, nor overwrite this previously issued certificate in its repository publication point without specific coordination. Only when the common registry is assured that the top down certificate issuance process to the receiving registry CA chain has been completed can the common registry commence the revocation of the original certificate for the sending registry. However, it should not do so until it is assured that the immediate subordinate registry CA in the path to the sending registry has issued a certificate with a reduced resource set, and so on. This implies that on the sending side the certificate issuance and revocation is a "bottom up" process.

If this process is not carefully followed, then the risk is that some or all of the subordinate certificates of this common registry CA will be unable to be validated until the entire process of

certificate issuance and revocation has been completed. While this sequenced process is intended to preserve validity of certificates in the RPKI, it is a complex and operationally cumbersome process.

The underlying consideration here is that the operational coordination of these certificate issuance and revocation actions to effect a smooth resource transfer across registries is mandated by the nature of the certificate validation process described in [RFC6487].

3. A Specific Resource RPKI Certificate Validation Process

The question considered here is: Is there an alternate definition of RPKI certificate validity that could remove the requirement for such careful orchestration of certification actions across the RPKI to support resource transfers?

The general definition of certificate validity as defined in [RFC5280] assumes a validation question relating to the relying party's (RP's) level of trust in a subject's signed material, given knowledge of a subject's name, the subject's public key, the RP's chosen trust anchor(s) and an overall PKI to define the domain of discourse.

The validation question assumed by the [RFC6487] RPKI certificate validation process relates to a RP's level of trust in the combination of some signed material, a certificate that attests to the public key used to sign this material and the set of all number resources that have been assigned or allocated to the subject of the certificate, given knowledge of the certificate, the RP's chosen trust anchor(s), the RPKI, and the application of the same test applied to the superior certificate in the RPKI hierarchy, and so on to a Trust Anchor.

There is a alternative certificate validation procedure that starts with an attestation containing the subject's signed material and an explicit enumeration of a set of number resources. The associated validation question relates to whether a RPKI validation process can attest to the validity of a subject's signed attestation relating to a particular set of number resources, rather than a signed attestation relating to all number resources held by this subject. We will term this alternate certificate validation process "specific resource" validation.

If the certificate validation procedure is specifically restricted to a question of ascertaining the validity of a particular set of number resources in the context of the RPKI, the RPKI validation procedure

need not be as strict as a recursive "encompassing" condition for the resources contained in each pair of certificates in the validation path. It would be sufficient in the context of this "specific resource" validation procedure to require only that each certificate in the validation path has a number resource extension that "encompasses" the specific resources described in the original validation question. Rather than a validation test for all possible questions, this is a specific validation question in the context of specific resources.

This validation question can be informally described as: Given a certificate and a given resource set, is there an Issuer-Subject ordered sequence of certificates from a Trust Anchor to the certificate being validated, where each certificate on this sequence is well-formed, not revoked by a valid CRL, where the certificate's lifetimes are valid, and where the RFC3779 resource extension in the certificate encompass the given resource set?

In the example from Section 1, using a this alternate certificate validation process, a validation question of certificate 3 and the resource 10.0.1.0/24, the validation outcome would be positive, in that certificates 1, 2 and 3 all encompass the specific resource 10.0.1.0/24, assuming that the certificates are valid in all other respects.

3.1. Resource Transfers and Specific Resource Certificate Validation

When considering the transfer of resources across registries, and the associated certification actions, then if the validation process was one of "specific resource" validation, then there is no requirement for synchronized orchestration of the process of certificate issuance and revocation by the CAs involved in this transfer in order to preserve the validity of resources described in these certificates.

Along the chain of the "sending" registry CA hierarchy each registry CA can issue a certificate with a reduced resource set that removes the resource being transferred, and revoke the previously issued certificate without regard to the specific timing of similar actions by either it's superior or its subordinate registry CA.

Similarly, in the "receiving" registry hierarchy each CA can issue a certificate with an augmented resource set that includes the resource being transferred without particular regard to the timing of similar actions by the other superior or subordinate registry CAs.

Validation questions relating to the migrating resource made against certificates on the "sending registry" will return an invalid outcome as soon as any registry CA in this chain has performed revocation of

the original certificate. Validation questions relating to the migrating resource made against certificates on the "receiving registry" will return an valid outcome only when all the registries in this chain have performed certificate re-issuance and included the resource in the new certificate.

Critically, at all times validation questions relating to any other resource using the "specific resource" validation approach will return the same outcomes throughout this issuance and revocation process. This "specific resource" validation process engenders a more robust outcome in RPKI certificate management. Validation questions relating to resources which are not being transferred from one registry to another cannot be compromised by any failure to adhere to a strict process of issuance and revocation relating to the certification of the resources being transferred.

3.2. A Specification of Specific Resource Validation

The following is a amended specification of certificate validation as described in [RFC6487] that describes the proposed "specific resource" certificate validation process.

Validation of signed resource data using a target resource certificate and a specific set of number resources consists of verifying that the digital signature of the signed resource data is valid, using the public key of the target resource certificate, and also validating the resource certificate in the context of the RPKI, using the path validation process. This path validation process verifies, among other things, that a prospective certification path (a sequence of n certificates) satisfies the following conditions:

1. for all 'x' in $\{1, \dots, n-1\}$, the Subject of certificate 'x' is the Issuer of certificate ('x' + 1);
2. certificate '1' is issued by a trust anchor;
3. certificate 'n' is the certificate to be validated; and
4. for all 'x' in $\{1, \dots, n\}$, certificate 'x' is valid.

Certificate validation entails verifying that all of the following conditions hold, in addition to the Certification Path Validation

criteria specified in Section 6 of [RFC5280]:

1. The certificate can be verified using the Issuer's public key and the signature algorithm
2. The current time lies within the certificate's Validity From and To values.
3. The certificate contains all fields that MUST be present, as defined by this specification, and contains values for selected fields that are defined as allowable values by this specification.
4. No field, or field value, that this specification defines as MUST NOT be present is used in the certificate.
5. The Issuer has not revoked the certificate. A revoked certificate is identified by the certificate's serial number being listed on the Issuer's current CRL, as identified by the CRLDP of the certificate, the CRL is itself valid, and the public key used to verify the signature on the CRL is the same public key used to verify the certificate itself.
6. The resource extension data contained in this certificate "encompasses" the entirety of the resources in the specific resource set.
7. The Certification Path originates with a certificate issued by a trust anchor, and there exists a signing chain across the Certification Path where the Subject of Certificate 'x' in the Certification Path matches the Issuer in Certificate 'x + 1' in the Certification Path, and the public key in Certificate 'x' can verify the signature value in Certificate 'x+1'.

A certificate validation algorithm MAY perform these tests in any chosen order.

4. Local Repository Cache Maintenance

This change in the validation process would have some impact on the operation of a local cache of validated RPKI certificates. Given that the validation process requires the specification of a specific resource set, it would appear that it would not be possible to validate an RPKI certificate without also specifying a specific resource set.

However, using a top-down validation process, and an additional local data structure associated with each locally held validated RPKI certificate, it is possible to maintain a local cache of validated certificates, and the set of valid and invalid resources for each certificate.

The additional data structures are the certificate's validated and invalidated resource set. These sets are defined as follows:

- o If the certificate is used as a Trust Anchor, then the local validated resource set is copied from the certificate's RFC3779 resource set. There is no invalid resource set.
- o Otherwise, the certificate's local validated resource set is defined as the intersection of this certificate's RFC3779 resource set and the parent certificate's local validated resource set. The certificate's invalid resource set is the difference between this set and the certificate's RFC3779 resource set.

If the certificate's validated resource set is empty then the certificate is not valid.

If the invalid resource set is not empty, then any resources that are contained in this invalid resource set cannot be valid by virtue of this certificate.

In all operations on the local repository cache, local applications should use the certificate's local validated resource set in place of the resources described in the certificate's RFC3779 extension.

The invalid resource set can be used as a diagnostic aide in local cache management.

5. Security Considerations

The Security Considerations of [RFC6487] apply to the validation procedure described here.

6. IANA Considerations

No updates to the registries are suggested by this document.

7. Acknowledgements

TBA.

8. References

8.1. Normative References

- [RFC3779] Lynn, C., Kent, S., and K. Seo, "X.509 Extensions for IP Addresses and AS Identifiers", RFC 3779, June 2004.
- [RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, May 2008.
- [RFC6487] Huston, G., Michaelson, G., and R. Loomans, "A Profile for X.509 PKIX Resource Certificates", RFC 6487, February 2012.

8.2. Informative References

- [RFC6482] Lepinski, M., Kent, S., and D. Kong, "A Profile for Route Origin Authorizations (ROAs)", RFC 6482, February 2012.

Authors' Addresses

Geoff Huston
Asia Pacific Network Information Centre (APNIC)
6 Cordelia St
South Brisbane, QLD 4101
Australia

Phone: +61 7 3858 3100
Email: gih@apnic.net

George Michaelson
Asia Pacific Network Information Centre (APNIC)
6 Cordelia St
South Brisbane, QLD 4101
Australia

Phone: +61 7 3858 3100
Email: ggm@apnic.net

Network Working Group
Internet-Draft
Intended status: Informational
Expires: August 13, 2014

G. Huston
G. Michaelson
APNIC
February 9, 2014

RPKI Validation Reconsidered
draft-huston-rpki-validation-01.txt

Abstract

This document reviews the certificate validation procedure specified in RFC6487 and highlights aspects of operational management of certificates in the RPKI in response to the movement of resources across registries, and the associated actions of Certification Authorities to maintain certification of resources during this movement. The document describes an alternative validation procedure that reduces the operational impact of certificate management during resource movement.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 13, 2014.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

Table of Contents

1. Introduction	3
2. Operational Considerations	4
3. A Specific Resource RPKI Certificate Validation Process . . .	6
3.1. Resource Transfers and Specific Resource Certificate Validation	8
3.2. A Specification of Specific Resource Validation	8
4. Local Repository Cache Maintenance	10
5. Security Considerations	11
6. IANA Considerations	11
7. Acknowledgements	11
8. References	11
8.1. Normative References	11
8.2. Informative References	12
Authors' Addresses	12

1. Introduction

This document reviews the certificate validation procedure specified in [RFC6487] and highlights aspects of operational management of certificates in the RPKI in response to the movement of resources across registries, and the associated actions of Certification Authorities to maintain certification of resources during this movement. The document describes an alternative validation procedure that reduces the operational impact of certificate management during resource movement. The alternative validation procedure also offers a higher level of robustness in the face of resource inconsistencies in a putative certificate validation path.

As currently defined in section 7.2 of [RFC6487], validation of PKIX certificates that conform to the RPKI profile relies on the use of a path validation process where each certificate in the validation path is required to meet the certificate validation criteria. This can be considered to be a recursive validation process where, in the context of an ordered sequence of certificates, as defined by common Issuer and Subject Name pairs, a certificate is defined as valid if it satisfies basic validation criteria relating to the syntactic correctness, currency of validity dates and similar properties of the certificate itself, as described in [RFC5280], and also that it satisfies certain additional criteria with respect to the previous certificate in the sequence, and that this previous certificate is itself a valid certificate using the same criteria. This definition applies recursively to all certificates in the sequence apart from the initial sequence element, which is required to be a Trust Anchor.

For RPKI certificates, the additional criteria relating to the previous certificate in this sequence is that the certificate's number resource set, as defined in [RFC3779], is "encompassed" by the number resource set contained in the previous certificate.

Because [RFC6487] validation demands that all resources in a certificate be valid under the parent (and recursively, to the root), a digitally signed attestation, such as a Route Origin Authorization (ROA) object [RFC6482], which refers only to a subset of RFC3779-specified resources from that certificate chain can be concluded to be invalid, but not by virtue of the relationship between the RFC3779 extensions of the certificates on the putative certificate validation path and the resources in the ROA, but by other resources described in these certificates where the "encompassing" relationship of the resources does not hold. Any such invalidity along the certificate validation path can cause this outcome, not just at the immediate parent of the end entity certificate that attests to the key used to sign the ROA.

For example, in the certificate sequence:

Certificate 1:

Issuer A, Subject B, Resources 192.0.2.0/24, AS64496-AS64500

Certificate 2:

Issuer B, Subject C, Resources 192.0.2.0/24/24, AS64496-AS64511

Certificate 3:

Issuer C, Subject D, Resources 192.0.2.0/24

Certificate 3 is considered to be an invalid certificate, because the resources in Certificate 2 are not encompassed by the resources in Certificate 1, by virtue of certificate 2 holding the resources of the range AS64501 - AS64511 in this RFC3779 resource extension. Obviously, these Autonomous Systems numbers are not related to the IPv4 resources contained in Certificate 3.

2. Operational Considerations

There are two areas of operational concern with the current RPKI validation definition.

The first is that of the robustness of the operational management procedures in the issuance of certificates. If a subordinate CA issues a certificate that contains an Internet Number Resource (INR) collection that is not either exactly equal to, or a strict subset of, its parent CA, then this issued certificate, and all subordinate certificates of this issued certificate are invalid. These certificates are not only defined as invalid when being considered to validate an INR that is not in the parent CA certificate, but are defined as invalid for all INRs in the certificate. This creates a degree of operational fragility in the issuance of certificates, as all CA's are now required to exercise extreme care in the issuance and reissuance of certificates that they do not overclaim on the resources described in the parent CA, as the consequences of an operational lapse or oversight implies that all the subordinate certificates from the point of mismatch are invalid. It would be preferred if the consequences of such an operational lapse were limited in scope to the specific INRs that formed the mismatch, rather than including the entire set of INRs within the scope of damage from this oversight.

The second operational consideration described here relates to the situation where a registry withdraws a resource from the current holder, and the resource is transferred to another registry, to be registered to a new holder in that registry. The reason why this is

a consideration in operational deployments of the RPKI lies in the movement of the "home" registry of number resources during cases of mergers, acquisitions, business re-alignments, and resource transfers and the desire to ensure that during this movement all other resources can continue to be validated.

If the original registry's certification actions are simply to issue a new certificate for the current holder with a reduced resource set, and to revoke the original certificate, then there is a distinct possibility of encountering the situation illustrated by the example in the previous section. This is a result of an operational process for certificate issuance by the parent CA being de-coupled from the certificate operations of child CA.

This de-coupled operation of CAs introduces a risk of unintended third party damage: since a CA certificate can refer to holdings which relate to two or more unrelated subordinate certificates, if this CA certificate becomes invalid due to the reduction in the resources allocated to this CA relating to one subordinate resource set, all other subordinate certificates are invalid until the CA certificate is reissued with a reduced resource set.

In the above example, all subordinate certificates issued by CA C are invalid until CA B issues a new certificate for CA C with a reduced resource set.

At the lower levels of the RPKI hierarchy the resource sets affected by such movements of resources may not encompass significantly large pools of resources. However, as one ascends through this hierarchy towards the apex, the larger the resource set that is going to be affected by a period of invalidity by virtue of such uncoordinated certificate management actions. In the case of a Regional Internet Registry (RIR) or National Internet Registry (NIR), the potential risk arising from uncoordinated certification actions relating to a transfer of resources is that the entire set of subordinate certificates that refer to resources administered by the RIR or the NIR cannot be validated during this period.

Avoiding such situations requires that CA's adhere to a very specific ordering of certificate issuance. In this framework, the common registry CA that describes (directly or indirectly) the resources being shifted from one registry to the other, and also contains in subordinate certificates (direct or indirect) the certificates for both registries who are parties to the resource transfer has to coordinate a specific sequence of actions.

This common registry CA has to first issue a new certificate towards the "receiving" registry that adds to the RFC3779 extension resource

set the specific resource being transferred into this receiving registry. The common registry CA then has to wait until all registries in the subordinate certificate chain to the receiving registry have also performed a similar issuance of new certificates, and in each case a registry must await the issuance of the immediate superior certificate with the augmented resource set before it, in turn, can issue its own augmented certificate to its subordinate CA. This is a "top down" issuance sequence."

It is possible for the common registry to issue a certificate to the "sending" registry with the reduced resource set at any time, but it should not revoke the previously issued certificate, nor overwrite this previously issued certificate in its repository publication point without specific coordination. Only when the common registry is assured that the top down certificate issuance process to the receiving registry CA chain has been completed can the common registry commence the revocation of the original certificate for the sending registry. However, it should not so until it is assured that the immediate subordinate registry CA in the path to the sending registry has issued a certificate with a reduced resource set, and so on. This implies that on the sending side the certificate issuance and revocation is a "bottom up" process.

If this process is not carefully followed, then the risk is that some or all of the subordinate certificates of this common registry CA will be unable to be validated until the entire process of certificate issuance and revocation has been completed. While this sequenced process is intended to preserve validity of certificates in the RPKI, it is a complex and operationally cumbersome process.

The underlying consideration here is that the operational coordination of these certificate issuance and revocation actions to effect a smooth resource transfer across registries is mandated by the nature of the certificate validation process described in [RFC6487].

3. A Specific Resource RPKI Certificate Validation Process

The question considered here is: Is there an alternate definition of RPKI certificate validity that could remove the requirement for such careful orchestration of certification actions across the RPKI to support resource transfers?

The general definition of certificate validity as defined in [RFC5280] assumes a validation question relating to the relying party's (RP's) level of trust in a subject's signed material, given knowledge of a subject's name, the subject's public key, the RP's

chosen trust anchor(s) and an overall PKI to define the domain of discourse.

The validation question assumed by the [RFC6487] RPKI certificate validation process relates to a RP's level of trust in the combination of some signed material, a certificate that attests to the public key used to sign this material and the set of all number resources that have been assigned or allocated to the subject of the certificate, given knowledge of the certificate, the RP's chosen trust anchor(s), the RPKI, and the application of the same test applied to the superior certificate in the RPKI hierarchy, and so on to a Trust Anchor.

There is a alternative certificate validation procedure that starts with an attestation containing the subject's signed material and an explicit enumeration of a set of number resources. The associated validation question relates to whether a RPKI validation process can attest to the validity of a subject's signed attestation relating to a particular set of number resources, rather than a signed attestation relating to all number resources held by this subject. We will term this alternate certificate validation process "specific resource" validation.

If the certificate validation procedure is specifically restricted to a question of ascertaining the validity of a particular set of number resources in the context of the RPKI, the RPKI validation procedure need not be as strict as a recursive "encompassing" condition for the resources contained in each pair of certificates in the validation path. It would be sufficient in the context of this "specific resource" validation procedure to require only that each certificate in the validation path has a number resource extension that "encompasses" the specific resources described in the original validation question. Rather than a validation test for all possible questions, this is a specific validation question in the context of specific resources.

This validation question can be informally described as: Given a certificate and a given resource set, is there an Issuer-Subject ordered sequence of certificates from a Trust Anchor to the certificate being validated, where each certificate on this sequence is well-formed, not revoked by a valid CRL, where the certificate's lifetimes are valid, and where the RFC3779 resource extension in the certificate encompass the given resource set?

In the example from Section 1, using a this alternate certificate validation process, a validation question of certificate 3 and the resource 10.0.1.0/24, the validation outcome would be positive, in that certificates 1, 2 and 3 all encompass the specific resource

10.0.1.0/24, assuming that the certificates are valid in all other respects.

3.1. Resource Transfers and Specific Resource Certificate Validation

When considering the transfer of resources across registries, and the associated certification actions, then if the validation process was one of "specific resource" validation, then there is no requirement for synchronized orchestration of the process of certificate issuance and revocation by the CAs involved in this transfer in order to preserve the validity of resources described in these certificates.

Along the chain of the "sending" registry CA hierarchy each registry CA can issue a certificate with a reduced resource set that removes the resource being transferred, and revoke the previously issued certificate without regard to the specific timing of similar actions by either it's superior or its subordinate registry CA.

Similarly, in the "receiving" registry hierarchy each CA can issue a certificate with an augmented resource set that includes the resource being transferred without particular regard to the timing of similar actions by the other superior or subordinate registry CAs.

Validation questions relating to the migrating resource made against certificates on the "sending registry" will return an invalid outcome as soon as any registry CA in this chain has performed revocation of the original certificate. Validation questions relating to the migrating resource made against certificates on the "receiving registry" will return an valid outcome only when all the registries in this chain have performed certificate re-issuance and included the resource in the new certificate.

Critically, at all times validation questions relating to any other resource using the "specific resource" validation approach will return the same outcomes throughout this issuance and revocation process. This "specific resource" validation process engenders a more robust outcome in RPKI certificate management. Validation questions relating to resources which are not being transferred from one registry to another cannot be compromised by any failure to adhere to a strict process of issuance and revocation relating to the certification of the resources being transferred.

3.2. A Specification of Specific Resource Validation

The following is a amended specification of certificate validation as described in [RFC6487] that describes the proposed "specific resource" certificate validation process.

Validation of signed resource data using a target resource certificate and a specific set of number resources consists of verifying that the digital signature of the signed resource data is valid, using the public key of the target resource certificate, and also validating the resource certificate in the context of the RPKI, using the path validation process. This path validation process verifies, among other things, that a prospective certification path (a sequence of n certificates) satisfies the following conditions:

1. for all 'x' in $\{1, \dots, n-1\}$, the Subject of certificate 'x' is the Issuer of certificate ('x' + 1);
2. certificate '1' is issued by a trust anchor;
3. certificate 'n' is the certificate to be validated; and
4. for all 'x' in $\{1, \dots, n\}$, certificate 'x' is valid.

Certificate validation entails verifying that all of the following conditions hold, in addition to the Certification Path Validation criteria specified in Section 6 of [RFC5280]:

1. The certificate can be verified using the Issuer's public key and the signature algorithm
2. The current time lies within the certificate's Validity From and To values.
3. The certificate contains all fields that MUST be present, as defined by this specification, and contains values for selected fields that are defined as allowable values by this specification.
4. No field, or field value, that this specification defines as MUST NOT be present is used in the certificate.

5. The Issuer has not revoked the certificate. A revoked certificate is identified by the certificate's serial number being listed on the Issuer's current CRL, as identified by the CRLDP of the certificate, the CRL is itself valid, and the public key used to verify the signature on the CRL is the same public key used to verify the certificate itself.
6. The resource extension data contained in this certificate "encompasses" the entirety of the resources in the specific resource set.
7. The Certification Path originates with a certificate issued by a trust anchor, and there exists a signing chain across the Certification Path where the Subject of Certificate 'x' in the Certification Path matches the Issuer in Certificate 'x + 1' in the Certification Path, and the public key in Certificate 'x' can verify the signature value in Certificate 'x+1'.

A certificate validation algorithm MAY perform these tests in any chosen order.

4. Local Repository Cache Maintenance

This change in the validation process would have some impact on the operation of a local cache of validated RPKI certificates. Given that the validation process requires the specification of a specific resource set, it would appear that it would not be possible to validate an RPKI certificate without also specifying a specific resource set.

However, using a top-down validation process, and an additional local data structure associated with each locally held validated RPKI certificate, it is possible to maintain a local cache of validated certificates, and the set of valid and invalid resources for each certificate.

The additional data structures are the certificate's validated and invalidated resource set. These sets are defined as follows:

- o If the certificate is used as a Trust Anchor, then the local validated resource set is copied from the certificate's RFC3779 resource set. There is no invalid resource set.
- o Otherwise, the certificate's local validated resource set is defined as the intersection of this certificate's RFC3779 resource

set and the parent certificate's local validated resource set. The certificate's invalid resource set is the difference between this set and the certificate's RFC3779 resource set.

If the certificate's validated resource set is empty then the certificate is not valid.

If the invalid resource set is not empty, then any resources that are contained in this invalid resource set cannot be valid by virtue of this certificate.

In all operations on the local repository cache, local applications should use the certificate's local validated resource set in place of the resources described in the certificate's RFC3779 extension.

The invalid resource set can be used as a diagnostic aide in local cache management.

5. Security Considerations

The Security Considerations of [RFC6487] apply to the validation procedure described here.

6. IANA Considerations

No updates to the registries are suggested by this document.

7. Acknowledgements

TBA.

8. References

8.1. Normative References

- [RFC3779] Lynn, C., Kent, S., and K. Seo, "X.509 Extensions for IP Addresses and AS Identifiers", RFC 3779, June 2004.
- [RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, May 2008.
- [RFC6487] Huston, G., Michaelson, G., and R. Loomans, "A Profile for

X.509 PKIX Resource Certificates", RFC 6487,
February 2012.

8.2. Informative References

[RFC6482] Lepinski, M., Kent, S., and D. Kong, "A Profile for Route
Origin Authorizations (ROAs)", RFC 6482, February 2012.

Authors' Addresses

Geoff Huston
Asia Pacific Network Information Centre (APNIC)
6 Cordelia St
South Brisbane, QLD 4101
Australia

Phone: +61 7 3858 3100
Email: gih@apnic.net

George Michaelson
Asia Pacific Network Information Centre (APNIC)
6 Cordelia St
South Brisbane, QLD 4101
Australia

Phone: +61 7 3858 3100
Email: ggm@apnic.net

Internet Engineering Task Force
Internet-Draft
Intended status: Best Current Practice
Expires: January 16, 2014

J. Durand
CISCO Systems, Inc.
I. Pepelnjak
NIL
G. Doering
SpaceNet
July 15, 2013

BGP operations and security
draft-ietf-opsec-bgp-security-01.txt

Abstract

BGP (Border Gateway Protocol) is the protocol almost exclusively used in the Internet to exchange routing information between network domains. Due to this central nature, it's important to understand the security measures that can and should be deployed to prevent accidental or intentional routing disturbances.

This document describes measures to protect the BGP sessions itself (like TTL, MD5, control plane filtering) and to better control the flow of routing information, using prefix filtering and automatization of prefix filters, max-prefix filtering, AS path filtering, route flap dampening and BGP community scrubbing.

Foreword

A placeholder to list general observations about this document.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [1].

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any

time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 16, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Definitions and Acronyms	3
3. Protection of the BGP router	3
4. Protection of BGP sessions	4
4.1. Protection of TCP sessions used by BGP	4
4.2. BGP TTL security	4
5. Prefix filtering	5
5.1. Definition of prefix filters	5
5.1.1. Prefixes that MUST not be routed by definition	5
5.1.2. Prefixes not allocated	5
5.1.3. Prefixes too specific	9
5.1.4. Filtering prefixes belonging to the local AS	9
5.1.5. IXP LAN prefixes	9
5.1.6. The default route	11
5.2. Prefix filtering recommendations in full routing networks	11
5.2.1. Filters with internet peers	12
5.2.2. Filters with customers	13
5.2.3. Filters with upstream providers	14
5.3. Prefix filtering recommendations for leaf networks	14
5.3.1. Inbound filtering	14
5.3.2. Outbound filtering	15
6. BGP route flap dampening	15
7. Maximum prefixes on a peering	15
8. AS-path filtering	16
9. Next-Hop Filtering	17
10. BGP community scrubbing	18

11. Change logs	18
11.1. Diffs between draft-jdurand-bgp-security-01 and draft-jdurand-bgp-security-00	18
11.2. Diffs between draft-jdurand-bgp-security-02 and draft-jdurand-bgp-security-01	19
11.3. Diffs between draft-ietf-opsec-bgp-security-00 and draft-jdurand-bgp-security-02	20
11.4. Diffs between draft-ietf-opsec-bgp-security-01 and draft-ietf-opsec-bgp-security-00	21
12. Acknowledgements	21
13. IANA Considerations	21
14. Security Considerations	22
15. References	22
15.1. Normative References	22
15.2. Informative References	23
Authors' Addresses	25

1. Introduction

BGP [6] is the protocol used in the internet to exchange routing information between network domains. This protocol does not directly include mechanisms that control that routes exchanged conform to the various rules defined by the Internet community. This document intends to both summarize common existing rules and help network administrators apply coherent BGP policies.

2. Definitions and Accronyms

- o Tier 1 transit provider: an IP transit provider which can reach any network on the internet without purchasing transit services
- o IXP: Internet eXchange Point

3. Protection of the BGP router

The BGP router needs to be protected from stray packets. This protection should be achieved by an access-list (ACL) which would discard all packets directed to TCP port 179 on the local device and sourced from an address not known or permitted to become a BGP neighbor. If supported, an ACL specific to the control-plane of the router should be used (receive-ACL, control-plane policing, etc.), to avoid filtering transit traffic if not needed. If the hardware can not do that, interface ACLs can be used to block packets to the local router.

Some routers automatically program such an ACL upon BGP configuration. On other devices this ACL should be configured and maintained manually or using scripts.

The filtering of packets destined to the local router is a wider topic than "just for BGP" (if you bring down a router by overloading one of the other protocols from remote, BGP is harmed as well). For a more detailed recommendation, see RFC6192 [21].

4. Protection of BGP sessions

Current issues of TCP-based protocols (therefore including BGP) have been documented in [27]. The following sub-sections recall the major points raised in this RFC and gives best practices for BGP operation.

4.1. Protection of TCP sessions used by BGP

Attacks on TCP sessions used by BGP (ex: sending spoofed TCP RST packets) could bring down the TCP session. Following a successful ARP spoofing attack (or other similar Man-in-the-Middle attack), the attacker might even be able to inject packets into the TCP stream (routing attacks).

TCP sessions used by BGP can be secured with a variety of mechanisms. MD5 protection of TCP session header [13] is the most common one as it was the first mechanism widely implemented on routers. IPsec or TCP Authentication Option (TCP-AO, [10]) offers stronger protection and should now be preferred.

The drawback of TCP session protection is additional configuration and management overhead for authentication information (ex: MD5 password) maintenance. Protection of TCP sessions used by BGP is thus recommended when peerings are established over shared networks where spoofing can be done (like IXPs).

You SHOULD block spoofed packets (packets with a source IP address belonging to your IP address space) at all edges of your network, making the protection of TCP sessions used by BGP unnecessary on iBGP or eBGP sessions run over point-to-point links.

4.2. BGP TTL security

BGP sessions can be made harder to spoof with the TTL security [9]. Instead of sending TCP packets with TTL value = 1, the routers send the TCP packets with TTL value = 255 and the receiver checks that the TTL value equals 255. Since it's impossible to send an IP packet with TTL = 255 to a non-directly-connected IP host, BGP TTL security effectively prevents all spoofing attacks coming from third parties not directly connected to the same subnet as the BGP-speaking routers. Network administrators SHOULD implement TTL security on directly connected BGP peerings.

Note: Like MD5 protection, TTL security has to be configured on both ends of a BGP session.

5. Prefix filtering

The main aspect of securing BGP resides in controlling the prefixes that are received/advertised on the BGP peerings. Prefixes exchanged between BGP peers are controlled with inbound and outbound filters that can match on IP prefixes (prefix filters, Section 5), AS paths (as-path filters, Section 8) or any other attributes of a BGP prefix (for example, BGP communities, Section 10).

5.1. Definition of prefix filters

This section list the most commonly used prefix filters. Following sections will clarify where these filters should be applied.

5.1.1. Prefixes that MUST not be routed by definition

5.1.1.1. IPv4

IPv4 registry [34] maintains the list of IPv4 special purpose prefixes and their routing scope. Reader will refer to this registry in order to configure prefix filters.

5.1.1.2. IPv6

IPv6 registry [35] maintains the list of IPv6 special purpose prefixes and their routing scope. Reader will refer to this registry in order to configure prefix filters.

At the time of the writing of this document, the list of IPv6 prefixes that MUST not cross network boundaries can be simplified as IANA allocates at the time being prefixes to RIR's only in 2000::/3 prefix [33]. All other prefixes (ULA's, link-local, multicast... are outside of that prefix) and therefore the simplified list becomes:

- o 2001:DB8::/32 and more specifics - documentation [15]
- o Prefixes more specifics than 2002::/16 - 6to4 [3]
- o 3FFE::/16 and more specifics - was initially used for the 6Bone (worldwide IPv6 test network) and returned to IANA
- o All prefixes that are outside 2000::/3 prefix

5.1.2. Prefixes not allocated

IANA allocates prefixes to RIRs which in turn allocate prefixes to LIRs. It is wise not to accept in the routing table prefixes that are not allocated. This could mean allocation made by IANA and/or allocations done by RIRs. This section details the options for building a list of allocated prefixes at every level. It is important to understand that filtering prefixes not allocated requires constant updates as prefixes are continually allocated. Therefore automation of such prefix filters is key for the success of this approach. One should probably not consider solutions described in this section if it is not capable of maintaining updated prefix filters: the damage would probably be worse than the intended security policy.

5.1.2.1. IANA allocated prefix filters

IANA has allocated all the IPv4 available space. Therefore there is no reason why one would keep checking prefixes are in the IANA allocated address space [36]. No specific filters need to be put in place by administrators who want to make sure that IPv4 prefixes they receive have been allocated by IANA.

For IPv6, given the size of the address space, it can be seen as wise accepting only prefixes derived from those allocated by IANA. Administrators can dynamically build this list from the IANA allocated IPv6 space [37]. As IANA keeps allocating prefixes to RIRs, the aforementioned list should be checked regularly against changes and if they occur, prefix filters should be computed and pushed on network devices. As there is delay between the time a RIR receives a new prefix and the moment it starts allocating portions of it to its LIRs, there is no need doing this step quickly and frequently. Based on past experience, authors recommend that the process in place makes sure there is no more than one month between the time the IANA IPv6 allocated prefix list changes and the moment all IPv6 prefix filters are updated.

If process in place (manual or automatic) cannot guarantee that the list is updated regularly then it's better not to configure any filters based on allocated networks. The IPv4 experience has shown that many network operators implemented filters for prefixes not allocated by IANA but did not update them on a regular basis. This created problems for latest allocations and required a extra work for RIRs that had to "de-bogonize" the newly allocated prefixes.

5.1.2.2. RIR allocated prefix filters

A more precise check can be performed as one would like to make sure that prefixes they receive are being originated or transited by autonomous systems entitled to do so. It has been observed in the past that one could easily advertise someone else's prefix (or more specific prefixes) and create black holes or security threats. To overcome that risk, administrators would need to make sure BGP advertisements correspond to information located in the existing registries. At this stage 2 options can be considered (short and long term options). They are described in the following subsections.

5.1.2.3. Prefix filters creation from Internet Routing Registries (IRR)

An Internet Routing Registry (IRR) is a database containing internet routing information, described using Routing Policy Specification Language objects [16]. Network administrators are given privileges to describe routing policies of their own networks in the IRR and information is published, usually publicly. Most of Regional Internet Registries do also operate an IRR and can control that registered routes conform to prefixes allocated or directly assigned.

It is possible to use the IRR information to build, for a given neighbor autonomous system, a list of prefixes originated or transited which one may accept. This can be done relatively easily using scripts and existing tools capable of retrieving this information in the registries. This approach is exactly the same for both IPv4 and IPv6.

The macro-algorithm for the script is described as follows. For the peer that is considered, the distant network administrator has provided the autonomous system and may be able to provide an AS-SET object (aka AS-MACRO). An AS-SET is an object which contains AS numbers or other AS-SETs. An operator may create an AS-SET defining all the AS numbers of its customers. A tier 1 transit provider might create an AS-SET describing the AS-SET of connected operators, which in turn describe the AS numbers of their customers. Using recursion, it is possible to retrieve from an AS-SET the complete list of AS numbers that the peer is likely to announce. For each of these AS numbers, it is also easy to check in the corresponding IRR for all associated prefixes. With these two mechanisms a script can build for a given peer the list of allowed prefixes and the AS number from which they should be originated. One could decide not use the origin information and only build monolithic prefix filters from fetched data.

As prefixes, AS numbers and AS-SETs may not all be under the same RIR authority, a difficulty resides choosing for each object the

appropriate IRR to poll. Some IRRs have been created and are not restricted to a given region or authoritative RIR. They allow RIRs to publish information contained in their IRR in a common place. They also make it possible for any subscriber (probably under contract) to publish information too. When doing requests inside such an IRR, it is possible to specify the source of information in order to have the most reliable data. One could check a popular IRR containing many sources (such as RADB [38], the Routing Assets Database) and only use information from sources representing the five current RIRs.

As objects in IRRs may quickly vary over time, it is important that prefix filters computed using this mechanism are refreshed regularly. A daily basis could even be considered as some routing changes must be done sometimes in a certain emergency and registries may be updated at the very last moment. It has to be noted that this approach significantly increases the complexity of the router configurations as it can quickly add tens of thousands configuration lines for some important peers.

5.1.2.4. SIDR - Secure Inter Domain Routing

An infrastructure called SIDR (Secure Inter-Domain Routing) [22] has been designed to secure internet advertisements. At the time this document is written, many documents have been published and a framework with a complete set of protocols is proposed so that advertisements can be checked against signed routing objects in RIR routing registries. There are basically two services that SIDR offers:

- o Origin validation seeks at making sure that attributes associated with a routes are correct (the major point being the validation of the AS number originating this route). Origin validation is now operational (Internet registries, protocols, implementations on some routers...) and in theory it can be implemented knowing that the proportion of signed resources is still low at the time this document is written.
- o Path validation provided by BGPsec [40] seeks at making sure that no ones announce fake/wrong BGP paths that would attract traffic for a given destination [41]. BGPsec is still an on-going work item at the time this document is written and therefore cannot be implemented.

Implementing SIDR mechanisms is expected to solve many of BGP routing security problems in the long term but it may take time for deployments to be made and objects to become signed. It also has to be pointed that SIDR infrastructure is complementing (not replacing)

the security best practices listed in this document. Authors therefore recommend to implement any SIDR proposed mechanism (example: route origin validation) on top of the other existing mechanisms even if they could sometimes appear targeting the same goal.

If route origin validation is implemented, authors recommend to implement the following rules. Each received route on a router SHOULD be checked against the RPKI data set: if a corresponding ROA is found and is valid then the prefix SHOULD be accepted. If the ROA is found and is INVALID then the prefix SHOULD be discarded. If an ROA is not found then the prefix SHOULD be accepted but corresponding route SHOULD be given a low preference.

5.1.3. Prefixes too specific

Most ISPs will not accept advertisements beyond a certain level of specificity (and in return do not announce prefixes they consider as too specific). That acceptable specificity is decided for each peering between the 2 BGP peers. Some ISP communities have tried to document acceptable specificity. This document does not make any judgement on what the best approach is, it just recalls that there are existing practices on the internet and recommends the reader to refer to what those are. As an example the RIPE community has documented that IPv4 prefixes longer than /24 and IPv6 prefixes longer than /48 are generally not announced/accepted in the internet [29] [30].

5.1.4. Filtering prefixes belonging to the local AS

A network SHOULD filter its own prefixes on peerings with all its peers (inbound direction). This prevents local traffic (from a local source to a local destination) from leaking over an external peering in case someone else is announcing the prefix over the Internet. This also protects the infrastructure which may directly suffer in case backbone's prefix is suddenly preferred over the Internet. To an extent, such filters can also be configured on a network for the prefixes of its downstreams in order to protect them too. Such filters must be defined with caution as they can break existing redundancy mechanisms. For example in case an operator has a multihomed customer, it should keep accepting the customer prefix from its peers and upstreams. This will make it possible for the customer to keep accessing its operator network (and other customers) via the internet in case the BGP peering between the customer and the operator is down.

5.1.5. IXP LAN prefixes

5.1.5.1. Network security

When a network is present on an IXP and peers with other IXP members over a common subnet (IXP LAN prefix), it MUST NOT accept more specific prefixes for the IXP LAN prefix from any of its external BGP peers. Accepting these routes may create a black hole for connectivity to the IXP LAN.

If the IXP LAN prefix is accepted as an "exact match", care needs to be taken to avoid other routers in the network sending IXP traffic towards the externally-learned IXP LAN prefix (recursive route lookup pointing into the wrong direction). This can be achieved by preferring IGP routes before eBGP, or by using "BGP next-hop-self" on all routes learned on that IXP.

If the IXP LAN prefix is accepted at all, it MUST only be accepted from the ASes that the IXP authorizes to announce it - which will usually be automatically achieved by filtering announcements by IRR DB.

5.1.5.2. pMTUd and the loose uRPF problem

In order to have pMTUd working in the presence of loose uRPF, it is necessary that all the networks that may source traffic that could flow through the IXP (ie. IXP members and their downstreams) have a route for the IXP LAN prefix. This is necessary as "packet too big" ICMP messages sent by IXP members' routers may be sourced using an address of the IXP LAN prefix. In the presence of loose uRPF, this ICMP packet is dropped if there is no route for the IXP LAN prefix or a less specific route covering IXP LAN prefix.

In that case, any IXP member SHOULD make sure it has a route for the IXP LAN prefix or a less specific prefix on all its routers and that it announces the IXP LAN prefix or less specific (up to a default route) to its downstreams. The announcements done for this purpose SHOULD pass IRR-generated filters described in Section 5.1.2.3 as well as "prefixes too specific" filters described in Section 5.1.3. The easiest way to implement this is that the IXP itself takes care of the origination of its prefix and advertises it to all IXP members through a BGP peering. Most likely the BGP route servers would be used for this. The IXP would most likely send its entire prefix which would be equal or less specific than the IXP LAN prefix.

5.1.5.3. Example

Let's take as an example an IXP in the RIPE region for IPv4. It would be allocated a /22 by RIPE NCC (X.Y.0.0/22 in our example) and use a /23 of this /22 for the IXP LAN (let say X.Y.0.0/23). This IXP LAN prefix is the one used by IXP members to configure eBGP peerings. The IXP could also be allocated an AS number (AS64496 in our example).

Any IXP member MUST make sure it filters prefixes more specific than X.Y.0.0/23 from all its eBGP peers. If it received X.Y.0.0/24 or X.Y.1.0/24 this could seriously impact its routing.

The IXP SHOULD originate X.Y.0.0/22 and advertise it to its members through an eBGP peering (most likely from its BGP route servers, configured with AS64496).

The IXP members SHOULD accept the IXP prefix only if it passes the IRR generated filters (see Section 5.1.2.3)

IXP members SHOULD then advertise X.Y.0.0/22 prefix to their downstreams. This announce would pass IRR based filters as it is originated by the IXP.

5.1.6. The default route

5.1.6.1. IPv4

The 0.0.0.0/0 prefix is likely not intended to be accepted nor advertised other than in specific customer / provider configurations, general filtering outside of these is RECOMMENDED.

5.1.6.2. IPv6

The ::/0 prefix is likely not intended to be accepted nor advertised other than in specific customer / provider configurations, general filtering outside of these is RECOMMENDED.

5.2. Prefix filtering recommendations in full routing networks

For networks that have the full internet BGP table, some policies should be applied on each BGP peer for received and advertised routes. It is recommended that each autonomous system configures rules for advertised and received routes at all its borders as this will protect the network and its peer even in case of misconfiguration. The most commonly used filtering policy is proposed in this section.

5.2.1. Filters with internet peers

5.2.1.1. Inbound filtering

There are basically 2 options, the loose one where no check will be done against RIR allocations and the strict one where it will be verified that announcements strictly conform to what is declared in routing registries.

5.2.1.1.1. Inbound filtering loose option

In this case, the following prefixes received from a BGP peer will be filtered:

- o Prefixes not routable (Section 5.1.1)
- o Prefixes not allocated by IANA (IPv6 only) (Section 5.1.2.1)
- o Routes too specific (Section 5.1.3)
- o Prefixes belonging to the local AS (Section 5.1.4)
- o IXP LAN prefixes (Section 5.1.5)
- o The default route (Section 5.1.6)

5.2.1.1.2. Inbound filtering strict option

In this case, filters are applied to make sure advertisements strictly conform to what is declared in routing registries (Section 5.1.2.2). In case of script failure each administrator may decide if all routes are accepted or rejected depending on routing policy. While accepting the routes during that time frame could break the BGP routing security, rejecting them might re-route too much traffic on transit peers, and could cause more harm than what a loose policy would have done.

In addition to this, one could apply the following filters beforehand in case the routing registry used as source of information by the script is not fully trusted:

- o Prefixes not routable (Section 5.1.1)
- o Routes too specific (Section 5.1.3)
- o Prefixes belonging to the local AS (Section 5.1.4)
- o IXP LAN prefixes (Section 5.1.5)

- o The default route (Section 5.1.6)

5.2.1.2. Outbound filtering

Configuration should be put in place to make sure that only appropriate prefixes are sent. These can be, for example, prefixes belonging to both the network in question and its downstreams. This can be achieved by using a combination of BGP communities, AS-paths or both. It can also be desirable that following filters are positioned before to avoid unwanted route announcement due to bad configuration:

- o Prefixes not routable (Section 5.1.1)
- o Routes too specific (Section 5.1.3)
- o IXP LAN prefixes (Section 5.1.5)
- o The default route (Section 5.1.6)

In case it is possible to list the prefixes to be advertised, then just configuring the list of allowed prefixes and denying the rest is sufficient.

5.2.2. Filters with customers

5.2.2.1. Inbound filtering

The inbound policy with end customers is pretty straightforward: only customers prefixes MUST be accepted, all others MUST be discarded. The list of accepted prefixes can be manually specified, after having verified that they are valid. This validation can be done with the appropriate IP address management authorities.

The same rules apply in case the customer is also a network connecting other customers (for example a tier 1 transit provider connecting service providers). An exception can be envisaged in case it is known that the customer network applies strict inbound/outbound prefix filtering, and the number of prefixes announced by that network is too large to list them in the router configuration. In that case filters as in Section 5.2.1.1 can be applied.

5.2.2.2. Outbound filtering

The outbound policy with customers may vary according to the routes customer wants to receive. In the simplest possible scenario, the customer may only want to receive only the default route, which can be done easily by applying a filter with the default route only.

In case the customer wants to receive the full routing (in case it is multihomed or if wants to have a view of the internet table), the following filters can be simply applied on the BGP peering:

- o Prefixes not routable (Section 5.1.1)
- o Routes too specific (Section 5.1.3)
- o The default route (Section 5.1.6)

There can be a difference for the default route that can be announced to the customer in addition to the full BGP table. This can be done simply by removing the filter for the default route. As the default route may not be present in the routing table, one may decide to originate it only for peerings where it has to be advertised.

5.2.3. Filters with upstream providers

5.2.3.1. Inbound filtering

In case the full routing table is desired from the upstream, the prefix filtering to apply is the same than the one for peers Section 5.2.1.1 with the exception of the default route. The default route can be desired from an upstream provider in addition to the full BGP table. In case the upstream provider is supposed to announce only the default route, a simple filter will be applied to accept only the default prefix and nothing else.

5.2.3.2. Outbound filtering

The filters to be applied would most likely not differ much from the ones applied for internet peers (Section 5.2.1.2). But different policies could be applied in case it is desired that a particular upstream does not provide transit to all the prefixes.

5.3. Prefix filtering recommendations for leaf networks

5.3.1. Inbound filtering

The leaf network will position the filters corresponding to the routes it is requesting from its upstream. In case a default route is requested, a simple inbound filter can be applied to accept only the default route (Section 5.1.6). In case the leaf network is not capable of listing the prefixes because the amount is too large (for example if it requires the full internet routing table) then it should configure filters to avoid receiving bad announcements from its upstream:

- o Prefixes not routable (Section 5.1.1)
- o Routes too specific (Section 5.1.3)
- o Prefixes belonging to local AS (Section 5.1.4)
- o The default route (Section 5.1.6) depending if the route is requested or not

5.3.2. Outbound filtering

A leaf network will most likely have a very straightforward policy: it will only announce its local routes. It can also configure the following prefixes filters described in Section 5.2.1.2 to avoid announcing invalid routes to its upstream provider.

6. BGP route flap dampening

The BGP route flap dampening mechanism makes it possible to give penalties to routes each time they change in the BGP routing table. Initially this mechanism was created to protect the entire internet from multiple events impacting a single network. Studies have shown that implementations of BGP route flap dampening could cause more harm than they solve problems and therefore RIPE community has in the past recommended not using BGP route flap dampening [28]. Works have then been conducted to propose new route flap dampening thresholds in order to make the solution "usable" [39] and RIPE has reviewed its recommendations in [31]. New thresholds have been proposed to make BGP route flap dampening usable. Authors of this document propose to follow RIPE recommendations and only use BGP route flap dampening with adjusted configured thresholds.

7. Maximum prefixes on a peering

It is recommended to configure a limit on the number of routes to be accepted from a peer. Following rules are generally recommended:

- o From peers, it is recommended to have a limit lower than the number of routes in the internet. This will shut down the BGP peering if the peer suddenly advertises the full table. One can also configure different limits for each peer, according to the number of routes they are supposed to advertise plus some headroom to permit growth.
- o From upstreams which provide full routing, it is recommended to have a limit higher than the number of routes in the internet. A limit is still useful in order to protect the network (and in particular the routers' memory) if too many routes are sent by the

upstream. The limit should be chosen according to the number of routes that can actually be handled by routers.

It is important to regularly review the limits that are configured as the internet can quickly change over time. Some vendors propose mechanisms to have two thresholds: while the higher number specified will shutdown the peering, the first threshold will only trigger a log and can be used to passively adjust limits based on observations made on the network.

8. AS-path filtering

The following rules SHOULD be applied on BGP AS-paths (for both 16 and 32 bits Autonomous System Numbers):

- o From customers, try to accept only AS(4)-Paths containing ASNs belonging to (or authorized to transit through) the customer. If you can not build and generate filtering expressions to implement this, consider accepting only path lengths relevant to the type of customer you have (as in, if they are a leaf or have customers of their own), try to discourage excessive prepending in such paths.
- o Do not advertise prefixes with non-empty AS-path if you do not intend to be transit for these prefixes.
- o Do not advertise prefixes with upstream AS numbers in the AS-path to your peering AS if you do not intend to be transit for these prefixes.
- o Do not accept prefixes with private AS numbers in the AS-path except from customers. Exception: an upstream offering some particular service like black-hole origination based on a private AS number. Customers should be informed by their upstream in order to put in place ad-hoc policy to use such services.
- o Do not advertise prefixes with private AS numbers in the AS-path. Exception: customers using BGP without having their own AS number MUST use private AS numbers to advertise their prefixes to their upstream. This private AS number is usually provided by the upstream.
- o Do not accept prefixes when the first AS number in the AS-path is not the one of the peer. In case the peering is done toward a BGP route-server [11] (connection on an IXP) with transparent AS path handling, this verification needs to be de-activated as the first AS number will be the one of an IXP member whereas the peer AS number will be the one of the BGP route-server.

- o Do not override BGP's default behavior accepting your own AS number in the AS-path. In case an exception to this is required, impacts should be studied carefully as this can create severe impact on routing.

AS-path filtering should be further analyzed when ASN renumbering is done. Such operation is common and mechanisms exist to allow smooth ASN migration [42]. The usual migration technique, local to a router, consists in modifying the AS-path so it is presented to a peer as if no renumbering was done. This makes it possible to change ASN of a router without reconfiguring all eBGP peers at the same time (as this operation would require synchronization with all peers attached to that router). During this renumbering operation, rules described above may be adjusted.

9. Next-Hop Filtering

If peering on a shared network, like an IXP, BGP can advertise prefixes with a 3rd-party next-hop, thus directing packets not to the peer announcing the prefix but somewhere else.

This is a desirable property for BGP route-server setups [11], where the route-server will relay routing information, but has neither capacity nor desire to receive the actual data packets. So the BGP route-server will announce prefixes with a next-hop setting pointing to the router that originally announced the prefix to the route-server.

In direct peerings between ISPs, this is undesirable, as one of the peers could trick the other one to send packets into a black hole (unreachable next-hop) or to an unsuspecting 3rd party who would then have to carry the traffic. Especially for black-holing, the root cause of the problem is hard to see without inspecting BGP prefixes at the receiving router at the IXP.

Therefore, an inbound route policy SHOULD be applied on IXP peerings in order to set the next-hop for accepted prefixes to the BGP peer IP address (belonging to the IXP LAN) that sent the prefix (which is what "next-hop-self" would enforce on the sending side).

This policy MUST NOT be used on route-server peerings, or on peerings where you intentionally permit the other side to send 3rd-party next-hops.

This policy also MUST be adjusted if Remote Triggered Black Holing best practice (aka RTBH [25]) is implemented. In that case one would apply a well-known BGP next-hop for routes it wants to filter (if an internet threat is observed from/to this route for example). This

well known next-hop will be statically routed to a null interface. In combination with unicast RPF check, this will discard traffic from and toward this prefix. Peers can exchange information about black-holes using for example particular BGP communities. One could propagate black-holes information to its peers using agreed BGP community: when receiving a route with that community one could change the next-hop in order to create the black hole.

10. BGP community scrubbing

Optionally we can consider the following rules on BGP AS-paths:

- o Scrub inbound communities with your AS number in the high-order bits - allow only those communities that customers/peers can use as a signaling mechanism
- o Do not remove other communities: your customers might need them to communicate with upstream providers. In particular do not (generally) remove the no-export community as it is usually announced by your peer for a certain purpose.

11. Change logs

11.1. Diffs between draft-jdurand-bgp-security-01 and draft-jdurand-bgp-security-00

Following changes have been made since previous document draft-jdurand-bgp-security-00:

- o "This documents" typo corrected in the former abstract
- o Add normative reference for RFC5082 in former section 3.2
- o "Non routable" changed in title of former section 4.1.1
- o Correction of typo for IPv4 loopback prefix in former section 4.1.1.1
- o Added shared transition space 100.64.0.0/10 in former section 4.1.1.1
- o Clarification that 2002::/16 6to4 prefix can cross network boundaries in former section 4.1.1.2
- o Rationale of 2000::/3 explained in former section 4.1.1.2

- o Added 3FFE::/16 prefix forgotten initially in the simplified list of prefixes that MUST not be routed by definition in former section 4.1.1.2
 - o Warn that filters for prefixes not allocated by IANA MUST only be done if regular refresh is guaranteed, with some words about the IPv4 experience, in former section 4.1.2.1
 - o Replace RIR database with IRR. A definition of IRR is added in former section 4.1.2.2
 - o Remove any reference to anti-spoofing in former section 4.1.4
 - o Clarification for IXP LAN prefix and pMTUd problem in former section 4.1.5
 - o "Autonomous filters" typo (instead of Autonomous systems) corrected in the former section 4.2
 - o Removal of an example for manual address validation in former section 4.2.2.1
 - o RFC5735 obsoletes RFC3300
 - o Ingress/Egress replaced by Inbound/Outbound in all the document
- 11.2. Diffs between draft-jdurand-bgp-security-02 and draft-jdurand-bgp-security-01

Following changes have been made since previous document draft-jdurand-bgp-security-01:

- o 2 documentation prefixes were forgotten due to errata in RFC5735. But all prefixes were removed from that document which now point to other references for sake of not creating a new "registry" that would become outdated sooner or later
- o Change MD5 section with global TCP security session and introducing TCP-AO in former section 3.1. Added reference to BCP38
- o Added new section 3 about BGP router protection with forwarding plane ACL
- o Change text about prefix acceptable specificity in former section 4.1.3 to explain this doc does not try to make recommendations

- o Refer as much as possible to existing registries to avoid creating a new one in former section 4.1.1.1 and 4.1.1.2
 - o Abstract reworded
 - o 6to4 exception described (only more specifics MUST be filtered)
 - o More specific -> more specifics
 - o should -> MUST for the prefixes an ISP needs to filter from its customers in former section 4.2.2.1
 - o Added "plus some headroom to permit growth" in former section 7
 - o Added new section on Next-Hop filtering
- 11.3. Diffs between draft-ietf-opsec-bgp-security-00 and draft-jdurand-bgp-security-02

Following changes have been made since previous document draft-jdurand-bgp-security-02:

- o Added a subsection for RTBH in next-hop section with reference to RFC6666
- o Changed last sentence of introduction
- o Many edits throughout the document
- o Added definition of tier 1 transit provider
- o Removed definition of a BGP peering
- o Removed description of routing policies for IPv6 prefixes in IANA special registry as this now contains a routing scope field
- o Added reference to RFC6598 and changed the IPv4 prefixes to be filtered by definition section
- o IXP added in acronym/definition section and only term used throughout the doc now

11.4. Diffs between draft-ietf-opsec-bgp-security-01 and draft-ietf-opsec-bgp-security-00

Following changes have been made since previous document draft-ietf-opsec-bgp-security-00:

- o Obsolete RFC2385 moved from normative to informative reference
- o Clarification of preference of TCP-AO over MD5 in former section 4.1
- o Mentioning KARP efforts in TCP session protection section in former section 4 and adding 3 RFC as informative references: 6518, 6862 and 6952
- o Removing reference to SIDR working-group
- o Better dissociating origin validation and path validation to clarify what's potentially available for deployment
- o Adding that SIDR mechanisms should be implemented in addition to the other ones mentioned throughout this document
- o Added a paragraph in former section 8 about ASN renumbering
- o Change of security considerations section
- o Added the newly created IANA IPv4 Special Purpose Address Registry instead of references to RFCs listing these addresses

12. Acknowledgements

The authors would like to thank the following people for their comments and support: Marc Blanchet, Ron Bonica, David Freedman, Wesley George, Daniel Ginsburg, David Groves, Mike Hugues, Tim Kleefass, Warren Kumari, Hagen Paul Pfeifer, Thomas Pinaud, Carlos Pignataro, Matjaz Straus, Tony Tauber, Gunter Van de Velde, Sebastian Wiesinger.

13. IANA Considerations

This memo includes no request to IANA.

14. Security Considerations

This document is entirely about BGP operational security. It depicts best practices that one should adopt to secure its BGP infrastructure: protecting BGP router and BGP sessions, adopting consistent BGP prefix and AS-path filters and configure other options to secure the BGP network.

On the other hand this document doesn't aim at depicting existing BGP implementations and their potential vulnerabilities and ways they handle errors. It will not detail how protection could be enforced against attack techniques using crafted packets.

15. References

15.1. Normative References

- [1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, <<http://xml.resource.org/public/rfc/html/rfc2119.html>>.
- [2] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, June 1999.
- [3] Carpenter, B. and K. Moore, "Connection of IPv6 Domains via IPv4 Clouds", RFC 3056, February 2001.
- [4] Huitema, C. and B. Carpenter, "Deprecating Site Local Addresses", RFC 3879, September 2004.
- [5] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, October 2005.
- [6] Rekhter, Y., Li, T., and S. Hares, "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, January 2006.
- [7] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.
- [8] Huitema, C., "Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs)", RFC 4380, February 2006.
- [9] Gill, V., Heasley, J., Meyer, D., Savola, P., and C. Pignataro, "The Generalized TTL Security Mechanism (GTSM)", RFC 5082, October 2007.

- [10] Touch, J., Mankin, A., and R. Bonica, "The TCP Authentication Option", RFC 5925, June 2010.
- [11] , "Internet Exchange Route Server", , <<http://tools.ietf.org/id/draft-ietf-idr-ix-bgp-route-server-00.txt>>.

15.2. Informative References

- [12] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, November 1997.
- [13] Heffernan, A., "Protection of BGP Sessions via the TCP MD5 Signature Option", RFC 2385, August 1998.
- [14] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, May 2000.
- [15] Huston, G., Lord, A., and P. Smith, "IPv6 Address Prefix Reserved for Documentation", RFC 3849, July 2004.
- [16] Blunk, L., Damas, J., Parent, F., and A. Robachevsky, "Routing Policy Specification Language next generation (RPSLng)", RFC 4012, March 2005.
- [17] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 4234, October 2005.
- [18] Blanchet, M., "Special-Use IPv6 Addresses", RFC 5156, April 2008.
- [19] Cotton, M. and L. Vegoda, "Special Use IPv4 Addresses", RFC 5735, January 2010.
- [20] Arkko, J., Cotton, M., and L. Vegoda, "IPv4 Address Blocks Reserved for Documentation", RFC 5737, January 2010.
- [21] Dugal, D., Pignataro, C., and R. Dunn, "Protecting the Router Control Plane", RFC 6192, March 2011.
- [22] Lepinski, M. and S. Kent, "An Infrastructure to Support Secure Internet Routing", RFC 6480, February 2012.
- [23] Lebovitz, G. and M. Bhatia, "Keying and Authentication for Routing Protocols (KARP) Design Guidelines", RFC 6518, February 2012.

- [24] Weil, J., Kuarsingh, V., Donley, C., Liljenstolpe, C., and M. Azinger, "IANA-Reserved IPv4 Prefix for Shared Address Space", BCP 153, RFC 6598, April 2012.
- [25] Hilliard, N. and D. Freedman, "A Discard Prefix for IPv6", RFC 6666, August 2012.
- [26] Lebovitz, G., Bhatia, M., and B. Weis, "Keying and Authentication for Routing Protocols (KARP) Overview, Threats, and Requirements", RFC 6862, March 2013.
- [27] Jethanandani, M., Patel, K., and L. Zheng, "Analysis of BGP, LDP, PCEP, and MSDP Issues According to the Keying and Authentication for Routing Protocols (KARP) Design Guide", RFC 6952, May 2013.
- [28] Smith, P. and C. Panig1, "RIPE-378 - RIPE Routing Working Group Recommendations On Route-flap Damping", May 2006.
- [29] Smith, P., Evans, R., and M. Hughes, "RIPE-399 - RIPE Routing Working Group Recommendations on Route Aggregation", December 2006.
- [30] Smith, P. and R. Evans, "RIPE-532 - RIPE Routing Working Group Recommendations on IPv6 Route Aggregation", November 2011.
- [31] Smith, P., Bush, R., Kuhne, M., Pelsser, C., Maennel, O., Patel, K., Mohapatra, P., and R. Evans, "RIPE-580 - RIPE Routing Working Group Recommendations On Route-flap Damping", January 2013.
- [32] Doering, G., "IPv6 BGP Filter Recommendations", November 2009, <<http://www.space.net/~gert/RIPE/ipv6-filters.html>>.
- [33] , "IANA IPv6 Address Space", , <<http://www.iana.org/assignments/ipv6-address-space/ipv6-address-space.xml>>.
- [34] , "IANA IPv4 Special Purpose Address Registry", , <<http://www.iana.org/assignments/iana-ipv6-special-registry/iana-ipv6-special-registry.xml>>.
- [35] , "IANA IPv6 Special Purpose Address Registry", , <<http://www.iana.org/assignments/iana-ipv6-special-registry/iana-ipv6-special-registry.xml>>.

- [36] , "IANA IPv4 Address Space Registry", , <<http://www.iana.org/assignments/ipv4-address-space/ipv4-address-space.xml>>.
- [37] , "IANA IPv6 Address Space Registry", , <<http://www.iana.org/assignments/ipv6-unicast-address-assignments/ipv6-unicast-address-assignments.xml>>.
- [38] , "Routing Assets Database", , <<http://www.radb.net>>.
- [39] , "Making Route Flap Damping Usable", , <<http://tools.ietf.org/id/draft-ietf-idr-rfd-usable-02.txt>>.
- [40] , "Security Requirements for BGP Path Validation", , <<http://datatracker.ietf.org/doc/draft-ietf-sidr-bgpsec-reqs/>>.
- [41] , "Threat Model for BGP Path Security", , <<http://datatracker.ietf.org/doc/draft-ietf-sidr-bgpsec-threats/>>.
- [42] , "Autonomous System (AS) Migration Features and Their Effects on the BGP AS_PATH Attribute", , <<http://datatracker.ietf.org/doc/draft-ga-idr-as-migration/>>.

Authors' Addresses

Jerome Durand
CISCO Systems, Inc.
11 rue Camille Desmoulins
Issy-les-Moulineaux 92782 CEDEX
FR

Email: jerduran@cisco.com

Ivan Pepelnjak
NIL Data Communications
Tivolska 48
Ljubljana 1000
Slovenia

Email: ip@nil.com

Gert Doering
SpaceNet AG
Joseph-Dollinger-Bogen 14
Muenchen D-80807
Germany

Email: gert@space.net

Internet Engineering Task Force
Internet-Draft
Intended status: Best Current Practice
Expires: June 2, 2015

J. Durand
CISCO Systems, Inc.
I. Pepelnjak
NIL
G. Doering
SpaceNet
December 2, 2014

BGP operations and security
draft-ietf-opsec-bgp-security-07.txt

Abstract

BGP (Border Gateway Protocol) is the protocol almost exclusively used in the Internet to exchange routing information between network domains. Due to this central nature, it is important to understand the security measures that can and should be deployed to prevent accidental or intentional routing disturbances.

This document describes measures to protect the BGP sessions itself (like TTL, TCP-AO, control plane filtering) and to better control the flow of routing information, using prefix filtering and automatization of prefix filters, max-prefix filtering, AS path filtering, route flap dampening and BGP community scrubbing.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [1].

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on May 29, 2015.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Scope of the document	3
3. Definitions and Acronyms	4
4. Protection of the BGP speaker	4
5. Protection of BGP sessions	5
5.1. Protection of TCP sessions used by BGP	5
5.2. BGP TTL security (GTSM)	6
6. Prefix filtering	6
6.1. Definition of prefix filters	6
6.1.1. Special purpose prefixes	6
6.1.2. Prefixes not allocated	7
6.1.3. Prefixes too specific	11
6.1.4. Filtering prefixes belonging to the local AS and downstreams	11
6.1.5. IXP LAN prefixes	11
6.1.6. The default route	12
6.2. Prefix filtering recommendations in full routing networks	13
6.2.1. Filters with Internet peers	13
6.2.2. Filters with customers	15
6.2.3. Filters with upstream providers	15
6.3. Prefix filtering recommendations for leaf networks . . .	16
6.3.1. Inbound filtering	16
6.3.2. Outbound filtering	16
7. BGP route flap dampening	17
8. Maximum prefixes on a peering	17
9. AS-path filtering	17
10. Next-Hop Filtering	19
11. BGP community scrubbing	20
12. Change logs	20
12.1. Diffs between draft-jdurand-bgp-security-01 and draft- jdurand-bgp-security-00	20

12.2.	Diffs between draft-jdurand-bgp-security-02 and draft-jdurand-bgp-security-01	21
12.3.	Diffs between draft-ietf-opsec-bgp-security-00 and draft-jdurand-bgp-security-02	22
12.4.	Diffs between draft-ietf-opsec-bgp-security-01 and draft-ietf-opsec-bgp-security-00	22
12.5.	Diffs between draft-ietf-opsec-bgp-security-02 and draft-ietf-opsec-bgp-security-01	23
12.6.	Diffs between draft-ietf-opsec-bgp-security-03 and draft-ietf-opsec-bgp-security-02	24
12.7.	Diffs between draft-ietf-opsec-bgp-security-04 and draft-ietf-opsec-bgp-security-03	25
12.8.	Diffs between draft-ietf-opsec-bgp-security-05 and draft-ietf-opsec-bgp-security-04	25
12.9.	Diffs between draft-ietf-opsec-bgp-security-06 and draft-ietf-opsec-bgp-security-05	25
13.	Acknowledgements	26
14.	IANA Considerations	26
15.	Security Considerations	26
16.	References	27
16.1.	Normative References	27
16.2.	Informative References	27
Appendix A.	IXP LAN prefix filtering - example	29
Authors' Addresses	30

1. Introduction

BGP (Border Gateway Protocol - RFC 4271 [2]) is the protocol used in the Internet to exchange routing information between network domains. BGP does not directly include mechanisms that control that routes exchanged conform to the various guidelines defined by the Internet community. This document intends to both summarize common existing guidelines and help network administrators apply coherent BGP policies.

2. Scope of the document

The guidelines defined in this document are intended for generic Internet BGP peerings. The nature of the Internet is such that Autonomous Systems can always agree on exceptions to a common framework for relevant local needs, and therefore configure a BGP session in a manner that may differ from the recommendations provided in this document. While this is perfectly acceptable, every configured exception might have an impact on the entire inter-domain routing environment and network administrators SHOULD carefully appraise this impact before implementation.

3. Definitions and Accronyms

- o ACL: Access Control List
- o ASN: Autonomous System Number
- o IRR: Internet Routing Registry
- o IXP: Internet eXchange Point
- o LIR: Local Internet Registry
- o pMTUd: Path MTU Discovery
- o RIR: Regional Internet Registry
- o Tier 1 transit provider: an IP transit provider which can reach any network on the Internet without purchasing transit services
- o uRPF: Unicast Reverse Path Forwarding

4. Protection of the BGP speaker

The BGP speaker needs to be protected from attempts to subvert the BGP session. This protection SHOULD be achieved by an Access Control List (ACL) which would discard all packets directed to TCP port 179 on the local device and sourced from an address not known or permitted to become a BGP neighbor. Experience has shown that natural protection TCP should offer is not always sufficient as it is sometimes run in control-plane software: in the absence of ACLs it is possible to attack a BGP speaker by simply sending a high volume of connection requests to it.

If supported, an ACL specific to the control-plane of the router SHOULD be used (receive-ACL, control-plane policing, etc.), to avoid configuration of data-plane filters for packets transiting through the router (and therefore not reaching the control plane). If the hardware can not do that, interface ACLs can be used to block packets addressed to the local router.

Some routers automatically program such an ACL upon BGP configuration. On other devices this ACL should be configured and maintained manually or using scripts.

In addition to strict filtering, rate-limiting MAY be configured for accepted BGP traffic. Rate-limiting BGP traffic consists in permitting only a certain quantity of bits per second (or packets per second) of BGP traffic to the control plane. This protects the BGP

router control plane in case the amount of BGP traffic overcomes platform capabilities.

Filtering and rate-limiting of control-plane traffic is a wider topic than "just for BGP" (if network administrator brings down a router by overloading one of the other protocols from remote, BGP is harmed as well). For a more detailed recommendation on how to protect the router's control plane, see RFC 6192 [11].

5. Protection of BGP sessions

Current security issues of TCP-based protocols (therefore including BGP) have been documented in RFC 6952 [14]. The following subsections list the major points raised in this RFC and give best practices related to TCP session protection for BGP operation.

5.1. Protection of TCP sessions used by BGP

Attacks on TCP sessions used by BGP (aka BGP sessions), for example sending spoofed TCP RST packets, could bring down a BGP peering. Following a successful ARP spoofing attack (or other similar Man-in-the-Middle attack), the attacker might even be able to inject packets into the TCP stream (routing attacks).

BGP sessions can be secured with a variety of mechanisms. MD5 protection of TCP session header, described in RFC 2385 [7], was the first such mechanism. It is now deprecated by TCP Authentication Option (TCP-AO, RFC 5925 [4]) which offers stronger protection. While MD5 is still the most used mechanism due to its availability in vendor's equipment, TCP-AO SHOULD be preferred when implemented.

IPsec could also be used for session protection. At the time this document is published, there is not enough experience on impacts of the use of IPsec for BGP peerings and further analysis is required to define guidelines.

The drawback of TCP session protection is additional configuration and management overhead for authentication information (ex: MD5 password) maintenance. Protection of TCP sessions used by BGP is thus NOT REQUIRED even when peerings are established over shared networks where spoofing can be done (like IXPs), but operators are RECOMMENDED to consider the trade-offs and to apply TCP session protection where appropriate.

Network administrators SHOULD block spoofed packets (packets with a source IP address belonging to their IP address space) at all edges of their network (see RFC 2827 [8] and RFC 3704 [9]). This protects

the TCP session used by iBGP from attackers outside the Autonomous System.

5.2. BGP TTL security (GTSM)

BGP sessions can be made harder to spoof with the Generalized TTL Security Mechanisms (GTSM, aka TTL security), defined in RFC 5082 [3]. Instead of sending TCP packets with TTL value of 1, the BGP speakers send the TCP packets with TTL value of 255 and the receiver checks that the TTL value equals 255. Since it's impossible to send an IP packet with TTL of 255 to a non-directly-connected IP host, BGP TTL security effectively prevents all spoofing attacks coming from third parties not directly connected to the same subnet as the BGP-speaking routers. Network administrators SHOULD implement TTL security on directly connected BGP peerings.

GTSM could also be applied to multi-hop BGP peering as well. To achieve this TTL needs to be configured with proper value depending on the distance between BGP speakers (using principle described above). Nevertheless it is not as effective as anyone inside the TTL diameter could spoof the TTL.

Like MD5 protection, TTL security has to be configured on both ends of a BGP session.

6. Prefix filtering

The main aspect of securing BGP resides in controlling the prefixes that are received/advertised on the BGP peerings. Prefixes exchanged between BGP peers are controlled with inbound and outbound filters that can match on IP prefixes (prefix filters, Section 6), AS paths (as-path filters, Section 9) or any other attributes of a BGP prefix (for example, BGP communities, Section 11).

6.1. Definition of prefix filters

This section list the most commonly used prefix filters. Following sections will clarify where these filters should be applied.

6.1.1. Special purpose prefixes

6.1.1.1. IPv4 special purpose prefixes

IANA IPv4 Special-Purpose Address Registry [22] maintains the list of IPv4 special purpose prefixes and their routing scope, and SHOULD be used for prefix filters configuration. Prefixes with value "False" in column "Global" SHOULD be discarded on Internet BGP peerings.

6.1.1.2. IPv6 special purpose prefixes

IANA IPv6 Special-Purpose Address Registry [23] maintains the list of IPv6 special purpose prefixes and their routing scope, and SHOULD be used for prefix filters configuration. Only prefixes with value "False" in column "Global" SHOULD be discarded on Internet BGP peerings.

6.1.2. Prefixes not allocated

IANA allocates prefixes to RIRs which in turn allocate prefixes to LIRs (Local Internet Registries). It is wise not to accept routing table prefixes that are not allocated by IANA and/or RIRs. This section details the options for building a list of allocated prefixes at every level. It is important to understand that filtering prefixes not allocated requires constant updates as prefixes are continually allocated. Therefore automation of such prefix filters is key for the success of this approach. Network administrators SHOULD NOT consider solutions described in this section if they are not capable of maintaining updated prefix filters: the damage would probably be worse than the intended security policy.

6.1.2.1. IANA allocated prefix filters

IANA has allocated all the IPv4 available space. Therefore there is no reason why network administrators would keep checking that prefixes they receive from BGP peers are in the IANA allocated IPv4 address space [24]. No specific filters need to be put in place by administrators who want to make sure that IPv4 prefixes they receive in BGP updates have been allocated by IANA.

For IPv6, given the size of the address space, it can be seen as wise accepting only prefixes derived from those allocated by IANA. Administrators can dynamically build this list from the IANA allocated IPv6 space [25]. As IANA keeps allocating prefixes to RIRs, the aforementioned list should be checked regularly against changes and if they occur, prefix filters should be computed and pushed on network devices. The list could also be pulled directly by routers when they implement such mechanisms. As there is delay between the time a RIR receives a new prefix and the moment it starts allocating portions of it to its LIRs, there is no need for doing this step quickly and frequently. However, network administrators SHOULD ensure that all IPv6 prefix filters are updated within maximum one month after any change in the list of IPv6 prefix allocated by IANA.

If process in place (manual or automatic) cannot guarantee that the list is updated regularly then it's better not to configure any

filters based on allocated networks. The IPv4 experience has shown that many network operators implemented filters for prefixes not allocated by IANA but did not update them on a regular basis. This created problems for latest allocations and required an extra work for RIRs that had to "de-bogonize" the newly allocated prefixes.

6.1.2.2. RIR allocated prefix filters

A more precise check can be performed when one would like to make sure that prefixes they receive are being originated or transited by autonomous systems entitled to do so. It has been observed in the past that an AS (Autonomous System) could easily advertise someone else's prefix (or more specific prefixes) and create black holes or security threats. To partially mitigate this risk, administrators would need to make sure BGP advertisements correspond to information located in the existing registries. At this stage 2 options can be considered (short and long term options). They are described in the following subsections.

6.1.2.2.1. Prefix filters creation from Internet Routing Registries (IRR)

An Internet Routing Registry (IRR) is a database containing Internet routing information, described using Routing Policy Specification Language objects - RFC 4012 [10]. Network administrators are given privileges to describe routing policies of their own networks in the IRR and information is published, usually publicly. A majority of Regional Internet Registries do also operate an IRR and can control that registered routes conform to prefixes allocated or directly assigned. However, it should be noted that the list of such prefixes is not necessarily a complete list, and as such the list of routes in an IRR is not the same as the set of RIR allocated prefixes.

It is possible to use the IRR information to build, for a given neighbor autonomous system, a list of prefixes originated or transited which one may accept. This can be done relatively easily using scripts and existing tools capable of retrieving this information in the registries. This approach is exactly the same for both IPv4 and IPv6.

The macro-algorithm for the script is described as follows. For the peer that is considered, the distant network administrator has provided the autonomous system and may be able to provide an AS-SET object (aka AS-MACRO). An AS-SET is an object which contains AS numbers or other AS-SETs. An operator may create an AS-SET defining all the AS numbers of its customers. A tier 1 transit provider might create an AS-SET describing the AS-SET of connected operators, which in turn describe the AS numbers of their customers. Using recursion,

it is possible to retrieve from an AS-SET the complete list of AS numbers that the peer is likely to announce. For each of these AS numbers, it is also easy to check in the corresponding IRR for all associated prefixes. With these two mechanisms a script can build for a given peer the list of allowed prefixes and the AS number from which they should be originated. One could decide not use the origin information and only build monolithic prefix filters from fetched data.

As prefixes, AS numbers and AS-SETs may not all be under the same RIR authority, a difficulty resides choosing for each object the appropriate IRR to poll. Some IRRs have been created and are not restricted to a given region or authoritative RIR. They allow RIRs to publish information contained in their IRR in a common place. They also make it possible for any subscriber (probably under contract) to publish information too. When doing requests inside such an IRR, it is possible to specify the source of information in order to have the most reliable data. One could check a popular IRR containing many sources (such as RADB [26], the Routing Assets Database) and only select as sources some desired RIRs and trusted major ISPs (Internet Service Providers).

As objects in IRRs may frequently vary over time, it is important that prefix filters computed using this mechanism are refreshed regularly. A daily basis could even be considered as some routing changes must be done sometimes in a certain emergency and registries may be updated at the very last moment. It has to be noted that this approach significantly increases the complexity of the router configurations as it can quickly add tens of thousands configuration lines for some important peers. To manage this complexity, network administrators could for example use IRRToolSet [29], a set of tools making it possible to simplify the creation of automated filters configuration from policies stored in IRR.

Last but not least, network administrators SHOULD publish and maintain their resources properly in IRR database maintained by their RIR, when available.

6.1.2.2.2. SIDR - Secure Inter Domain Routing

An infrastructure called SIDR (Secure Inter-Domain Routing), described in RFC 6480 [12] has been designed to secure Internet advertisements. At the time this document is written, many documents have been published and a framework with a complete set of protocols is proposed so that advertisements can be checked against signed routing objects in RIR routing registries. There are basically two services that SIDR offers:

- o Origin validation, described in RFC 6811 [5], seeks at making sure that attributes associated with a routes are correct (the major point being the validation of the AS number originating this route). Origin validation is now operational (Internet registries, protocols, implementations on some routers...) and in theory it can be implemented knowing that the proportion of signed resources is still low at the time this document is written.
- o Path validation provided by BGPsec [27] seeks at making sure that no ones announce fake/wrong BGP paths that would attract traffic for a given destination, see RFC 7132 [16]. BGPsec is still an on-going work item at the time this document is written and therefore cannot be implemented.

Implementing SIDR mechanisms is expected to solve many of BGP routing security problems in the long term but it may take time for deployments to be made and objects to become signed. It also has to be pointed that SIDR infrastructure is complementing (not replacing) the security best practices listed in this document. Network administrators SHOULD therefore implement any SIDR proposed mechanism (example: route origin validation) on top of the other existing mechanisms even if they could sometimes appear targeting the same goal.

If route origin validation is implemented, reader SHOULD refer to rules described in RFC 7115 [15]. In short, each external route received on a router SHOULD be checked against the RPKI data set:

- o If a corresponding ROA (Route Origin Authorization) is found and is valid then the prefix SHOULD be accepted.
- o If the ROA is found and is INVALID then the prefix SHOULD be discarded.
- o If an ROA is not found then the prefix SHOULD be accepted but corresponding route SHOULD be given a low preference.

In addition to this, network administrators SHOULD sign their routing objects so their routes can be validated by other networks running origin validation.

One should understand that the RPKI model brings new interesting challenges. The paper On the Risk of Misbehaving RPKI Authorities [30] explains how RPKI model can impact the Internet if authorities don't behave as they are supposed to do. Further analysis is certainly required on RPKI, which carries part of BGP security.

6.1.3. Prefixes too specific

Most ISPs will not accept advertisements beyond a certain level of specificity (and in return do not announce prefixes they consider as too specific). That acceptable specificity is decided for each peering between the 2 BGP peers. Some ISP communities have tried to document acceptable specificity. This document does not make any judgement on what the best approach is, it just recalls that there are existing practices on the Internet and recommends the reader to refer to what those are. As an example the RIPE community has documented that as of the time of writing of this document, IPv4 prefixes longer than /24 and IPv6 prefixes longer than /48 are generally not announced/accepted in the Internet [19] [20]. These values may change in the future.

6.1.4. Filtering prefixes belonging to the local AS and downstreams

A network SHOULD filter its own prefixes on peerings with all its peers (inbound direction). This prevents local traffic (from a local source to a local destination) from leaking over an external peering in case someone else is announcing the prefix over the Internet. This also protects the infrastructure which may directly suffer in case backbone's prefix is suddenly preferred over the Internet.

In some cases, for example in multi-homing scenarios, such filters SHOULD NOT be applied as this would break the desired redundancy.

To an extent, such filters can also be configured on a network for the prefixes of its downstreams in order to protect them too. Such filters must be defined with caution as they can break existing redundancy mechanisms. For example in case an operator has a multihomed customer, it should keep accepting the customer prefix from its peers and upstreams. This will make it possible for the customer to keep accessing its operator network (and other customers) via the Internet in case the BGP peering between the customer and the operator is down.

6.1.5. IXP LAN prefixes

6.1.5.1. Network security

When a network is present on an IXP and peers with other IXP members over a common subnet (IXP LAN prefix), it SHOULD NOT accept more specific prefixes for the IXP LAN prefix from any of its external BGP peers. Accepting these routes may create a black hole for connectivity to the IXP LAN.

If the IXP LAN prefix is accepted as an "exact match", care needs to be taken to avoid other routers in the network sending IXP traffic towards the externally-learned IXP LAN prefix (recursive route lookup pointing into the wrong direction). This can be achieved by preferring IGP routes before eBGP, or by using "BGP next-hop-self" on all routes learned on that IXP.

If the IXP LAN prefix is accepted at all, it SHOULD only be accepted from the ASes that the IXP authorizes to announce it - which will usually be automatically achieved by filtering announcements by IRR DB.

6.1.5.2. pMTUD and the loose uRPF problem

In order to have pMTUD working in the presence of loose uRPF, it is necessary that all the networks that may source traffic that could flow through the IXP (ie. IXP members and their downstreams) have a route for the IXP LAN prefix. This is necessary as "packet too big" ICMP messages sent by IXP members' routers may be sourced using an address of the IXP LAN prefix. In the presence of loose uRPF, this ICMP packet is dropped if there is no route for the IXP LAN prefix or a less specific route covering IXP LAN prefix.

In that case, any IXP member SHOULD make sure it has a route for the IXP LAN prefix or a less specific prefix on all its routers and that it announces the IXP LAN prefix or less specific (up to a default route) to its downstreams. The announcements done for this purpose SHOULD pass IRR-generated filters described in Section 6.1.2.2.1 as well as "prefixes too specific" filters described in Section 6.1.3. The easiest way to implement this is that the IXP itself takes care of the origination of its prefix and advertises it to all IXP members through a BGP peering. Most likely the BGP route servers would be used for this. The IXP would most likely send its entire prefix which would be equal or less specific than the IXP LAN prefix.

Appendix Appendix A gives an example of guidelines regarding IXP LAN prefix.

6.1.6. The default route

6.1.6.1. IPv4

The 0.0.0.0/0 prefix is likely not intended to be accepted nor advertised other than in specific customer / provider configurations, general filtering outside of these is RECOMMENDED.

6.1.6.2. IPv6

The `::/0` prefix is likely not intended to be accepted nor advertised other than in specific customer / provider configurations, general filtering outside of these is RECOMMENDED.

6.2. Prefix filtering recommendations in full routing networks

For networks that have the full Internet BGP table, some policies should be applied on each BGP peer for received and advertised routes. It is RECOMMENDED that each autonomous system configures rules for advertised and received routes at all its borders as this will protect the network and its peer even in case of misconfiguration. The most commonly used filtering policy is proposed in this section and uses prefix filters defined in previous section Section 6.1.

6.2.1. Filters with Internet peers

6.2.1.1. Inbound filtering

There are basically 2 options, the loose one where no check will be done against RIR allocations and the strict one where it will be verified that announcements strictly conform to what is declared in routing registries.

6.2.1.1.1. Inbound filtering loose option

In this case, the following prefixes received from a BGP peer will be filtered:

- o Prefixes not globally routable (Section 6.1.1)
- o Prefixes not allocated by IANA (IPv6 only) (Section 6.1.2.1)
- o Routes too specific (Section 6.1.3)
- o Prefixes belonging to the local AS (Section 6.1.4)
- o IXP LAN prefixes (Section 6.1.5)
- o The default route (Section 6.1.6)

6.2.1.1.2. Inbound filtering strict option

In this case, filters are applied to make sure advertisements strictly conform to what is declared in routing registries (Section 6.1.2.2). Warning is given as registries are not always

accurate (prefixes missing, wrong information...) This varies across the registries and regions of the Internet. Before applying a strict policy the reader SHOULD check the impact on the filter and make sure solution is not worse than the problem.

Also in case of script failure each administrator may decide if all routes are accepted or rejected depending on routing policy. While accepting the routes during that time frame could break the BGP routing security, rejecting them might re-route too much traffic on transit peers, and could cause more harm than what a loose policy would have done.

In addition to this, network administrators could apply the following filters beforehand in case the routing registry used as source of information by the script is not fully trusted:

- o Prefixes not globally routable (Section 6.1.1)
- o Routes too specific (Section 6.1.3)
- o Prefixes belonging to the local AS (Section 6.1.4)
- o IXP LAN prefixes (Section 6.1.5)
- o The default route (Section 6.1.6)

6.2.1.2. Outbound filtering

Configuration should be put in place to make sure that only appropriate prefixes are sent. These can be, for example, prefixes belonging to both the network in question and its downstreams. This can be achieved by using a combination of BGP communities, AS-paths or both. It can also be desirable that following filters are positioned before to avoid unwanted route announcement due to bad configuration:

- o Prefixes not globally routable (Section 6.1.1)
- o Routes too specific (Section 6.1.3)
- o IXP LAN prefixes (Section 6.1.5)
- o The default route (Section 6.1.6)

In case it is possible to list the prefixes to be advertised, then just configuring the list of allowed prefixes and denying the rest is sufficient.

6.2.2. Filters with customers

6.2.2.1. Inbound filtering

The inbound policy with end customers is pretty straightforward: only customers prefixes SHOULD be accepted, all others SHOULD be discarded. The list of accepted prefixes can be manually specified, after having verified that they are valid. This validation can be done with the appropriate IP address management authorities.

The same rules apply in case the customer is also a network connecting other customers (for example a tier 1 transit provider connecting service providers). An exception can be envisaged in case it is known that the customer network applies strict inbound/outbound prefix filtering, and the number of prefixes announced by that network is too large to list them in the router configuration. In that case filters as in Section 6.2.1.1 can be applied.

6.2.2.2. Outbound filtering

The outbound policy with customers may vary according to the routes customer wants to receive. In the simplest possible scenario, the customer may only want to receive only the default route, which can be done easily by applying a filter with the default route only.

In case the customer wants to receive the full routing (in case it is multihomed or if wants to have a view of the Internet table), the following filters can be simply applied on the BGP peering:

- o Prefixes not globally routable (Section 6.1.1)
- o Routes too specific (Section 6.1.3)
- o The default route (Section 6.1.6)

There can be a difference for the default route that can be announced to the customer in addition to the full BGP table. This can be done simply by removing the filter for the default route. As the default route may not be present in the routing table, network administrators may decide to originate it only for peerings where it has to be advertised.

6.2.3. Filters with upstream providers

6.2.3.1. Inbound filtering

In case the full routing table is desired from the upstream, the prefix filtering to apply is the same as the one for peers Section 6.2.1.1 with the exception of the default route. The default route can be desired from an upstream provider in addition to the full BGP table. In case the upstream provider is supposed to announce only the default route, a simple filter will be applied to accept only the default prefix and nothing else.

6.2.3.2. Outbound filtering

The filters to be applied would most likely not differ much from the ones applied for Internet peers (Section 6.2.1.2). But different policies could be applied in case it is desired that a particular upstream does not provide transit to all the prefixes.

6.3. Prefix filtering recommendations for leaf networks

6.3.1. Inbound filtering

The leaf network will deploy the filters corresponding to the routes it is requesting from its upstream. In case a default route is requested, a simple inbound filter can be applied to accept only the default route (Section 6.1.6). In case the leaf network is not capable of listing the prefixes because the amount is too large (for example if it requires the full Internet routing table) then it should configure filters to avoid receiving bad announcements from its upstream:

- o Prefixes not routable (Section 6.1.1)
- o Routes too specific (Section 6.1.3)
- o Prefixes belonging to local AS (Section 6.1.4)
- o The default route (Section 6.1.6) depending if the route is requested or not

6.3.2. Outbound filtering

A leaf network will most likely have a very straightforward policy: it will only announce its local routes. It can also configure the following prefixes filters described in Section 6.2.1.2 to avoid announcing invalid routes to its upstream provider.

7. BGP route flap dampening

The BGP route flap dampening mechanism makes it possible to give penalties to routes each time they change in the BGP routing table. Initially this mechanism was created to protect the entire Internet from multiple events impacting a single network. Studies have shown that implementations of BGP route flap dampening could cause more harm than they solve problems and therefore RIPE community has in the past recommended not using BGP route flap dampening [18]. Studies have then been conducted to propose new route flap dampening thresholds in order to make the solution "usable", see RFC 7196 [6] and RIPE has reviewed its recommendations in [21]. This document RECOMMENDS following IETF and RIPE recommendations and only use BGP route flap dampening with the adjusted configured thresholds.

8. Maximum prefixes on a peering

It is RECOMMENDED to configure a limit on the number of routes to be accepted from a peer. Following rules are generally RECOMMENDED:

- o From peers, it is RECOMMENDED to have a limit lower than the number of routes in the Internet. This will shut down the BGP peering if the peer suddenly advertises the full table. Network administrators can also configure different limits for each peer, according to the number of routes they are supposed to advertise plus some headroom to permit growth.
- o From upstreams which provide full routing, it is RECOMMENDED to have a limit higher than the number of routes in the Internet. A limit is still useful in order to protect the network (and in particular the routers' memory) if too many routes are sent by the upstream. The limit should be chosen according to the number of routes that can actually be handled by routers.

It is important to regularly review the limits that are configured as the Internet can quickly change over time. Some vendors propose mechanisms to have two thresholds: while the higher number specified will shutdown the peering, the first threshold will only trigger a log and can be used to passively adjust limits based on observations made on the network.

9. AS-path filtering

This section lists the RECOMMENDED practices when processing BGP AS-paths:

- o Network administrators SHOULD accept from customers only AS(4)-Paths containing ASNs belonging to (or authorized to transit

through) the customer. If network administrators can not build and generate filtering expressions to implement this, they SHOULD consider accepting only path lengths relevant to the type of customer they have (as in, if these customers are a leaf or have customers of their own), and try to discourage excessive prepending in such paths. This loose policy could be combined with filters for specific AS(4)-Paths that must not be accepted if advertised by the customer, such as upstream transit provider or peer ASNs.

- o Network administrators SHOULD NOT accept prefixes with private AS numbers in the AS-path except from customers. Exception: an upstream offering some particular service like black-hole origination based on a private AS number. Customers should be informed by their upstream in order to put in place ad-hoc policy to use such services.
- o Network administrators SHOULD NOT accept prefixes when the first AS number in the AS-path is not the one of the peer unless the peering is done toward a BGP route-server [17] (for example on an IXP) with transparent AS path handling. In that case this verification needs to be de-activated as the first AS number will be the one of an IXP member whereas the peer AS number will be the one of the BGP route-server.
- o Network administrators SHOULD NOT advertise prefixes with non-empty AS-path unless they intend to be transit for these prefixes.
- o Network administrators SHOULD NOT advertise prefixes with upstream AS numbers in the AS-path to their peering AS unless they intend to be transit for these prefixes.
- o Private AS numbers are conventionally used in contexts that are "private" and SHOULD NOT be used in advertisements to BGP peers that are not party to such private arrangements, and should be stripped when received from BGP peers that are not party to such private arrangements.
- o Network administrators SHOULD NOT override BGP's default behavior accepting their own AS number in the AS-path. In case an exception to this is required, impacts should be studied carefully as this can create severe impact on routing.

AS-path filtering should be further analyzed when ASN renumbering is done. Such operation is common and mechanisms exist to allow smooth ASN migration [28]. The usual migration technique, local to a router, consists in modifying the AS-path so it is presented to a peer with the previous ASN, as if no renumbering was done. This

makes it possible to change ASN of a router without reconfiguring all eBGP peers at the same time (as this operation would require synchronization with all peers attached to that router). During this renumbering operation, rules described above may be adjusted.

10. Next-Hop Filtering

If peering on a shared network, like an IXP, BGP can advertise prefixes with a 3rd-party next-hop, thus directing packets not to the peer announcing the prefix but somewhere else.

This is a desirable property for BGP route-server setups [17], where the route-server will relay routing information, but has neither capacity nor desire to receive the actual data packets. So the BGP route-server will announce prefixes with a next-hop setting pointing to the router that originally announced the prefix to the route-server.

In direct peerings between ISPs, this is undesirable, as one of the peers could trick the other one to send packets into a black hole (unreachable next-hop) or to an unsuspecting 3rd party who would then have to carry the traffic. Especially for black-holing, the root cause of the problem is hard to see without inspecting BGP prefixes at the receiving router at the IXP.

Therefore, an inbound route policy SHOULD be applied on IXP peerings in order to set the next-hop for accepted prefixes to the BGP peer IP address (belonging to the IXP LAN) that sent the prefix (which is what "next-hop-self" would enforce on the sending side).

This policy SHOULD NOT be used on route-server peerings, or on peerings where network administrators intentionally permit the other side to send 3rd-party next-hops.

This policy also SHOULD be adjusted if Remote Triggered Black Holing best practice (aka RTBH - RFC 6666 [13]) is implemented. In that case network administrators would apply a well-known BGP next-hop for routes they want to filter (if an Internet threat is observed from/to this route for example). This well known next-hop will be statically routed to a null interface. In combination with unicast RPF check, this will discard traffic from and toward this prefix. Peers can exchange information about black-holes using for example particular BGP communities. Network administrators could propagate black-holes information to their peers using agreed BGP community: when receiving a route with that community a configured policy could change the next-hop in order to create the black hole.

11. BGP community scrubbing

Optionally we can consider the following rules on BGP AS-paths:

- o Network administrators SHOULD scrub inbound communities with their number in the high-order bits, and allow only those communities that customers/peers can use as a signaling mechanism
- o Networks administrators SHOULD NOT remove other communities applied on received routes (communities not removed after application of previous statement). In particular they SHOULD keep original communities when they apply a community. Customers might need them to communicate with upstream providers. In particular network administrators SHOULD NOT (generally) remove the no-export community as it is usually announced by their peer for a certain purpose.

12. Change logs

!!! NOTE TO THE RFC EDITOR: THIS SECTION WAS ADDED TO TRACK CHANGES AND FACILITATE WORKING GROUP COLLABORATION. IT MUST BE DELETED BEFORE PUBLICATION !!!

12.1. Diffs between draft-jdurand-bgp-security-01 and draft-jdurand-bgp-security-00

Following changes have been made since previous document draft-jdurand-bgp-security-00:

- o "This documents" typo corrected in the former abstract
- o Add normative reference for RFC5082 in former section 3.2
- o "Non routable" changed in title of former section 4.1.1
- o Correction of typo for IPv4 loopback prefix in former section 4.1.1.1
- o Added shared transition space 100.64.0.0/10 in former section 4.1.1.1
- o Clarification that 2002::/16 6to4 prefix can cross network boundaries in former section 4.1.1.2
- o Rationale of 2000::/3 explained in former section 4.1.1.2

- o Added 3FFE::/16 prefix forgotten initially in the simplified list of prefixes that must not be routed by definition in former section 4.1.1.2
 - o Warn that filters for prefixes not allocated by IANA MUST only be done if regular refresh is guaranteed, with some words about the IPv4 experience, in former section 4.1.2.1
 - o Replace RIR database with IRR. A definition of IRR is added in former section 4.1.2.2
 - o Remove any reference to anti-spoofing in former section 4.1.4
 - o Clarification for IXP LAN prefix and pMTUd problem in former section 4.1.5
 - o "Autonomous filters" typo (instead of Autonomous systems) corrected in the former section 4.2
 - o Removal of an example for manual address validation in former section 4.2.2.1
 - o RFC5735 obsoletes RFC3300
 - o Ingress/Egress replaced by Inbound/Outbound in all the document
- 12.2. Diffs between draft-jdurand-bgp-security-02 and draft-jdurand-bgp-security-01

Following changes have been made since previous document draft-jdurand-bgp-security-01:

- o 2 documentation prefixes were forgotten due to errata in RFC5735. But all prefixes were removed from that document which now point to other references for sake of not creating a new "registry" that would become outdated sooner or later
- o Change MD5 section with global TCP security session and introducing TCP-AO in former section 3.1. Added reference to BCP38
- o Added new section 3 about BGP router protection with forwarding plane ACL
- o Change text about prefix acceptable specificity in former section 4.1.3 to explain this doc does not try to make recommendations

- o Refer as much as possible to existing registries to avoid creating a new one in former section 4.1.1.1 and 4.1.1.2
 - o Abstract reworded
 - o 6to4 exception described (only more specifics MUST be filtered)
 - o More specific -> more specifics
 - o should -> MUST for the prefixes an ISP needs to filter from its customers in former section 4.2.2.1
 - o Added "plus some headroom to permit growth" in former section 7
 - o Added new section on Next-Hop filtering
- 12.3. Diffs between draft-ietf-opsec-bgp-security-00 and draft-jdurand-bgp-security-02

Following changes have been made since previous document draft-jdurand-bgp-security-02:

- o Added a subsection for RTBH in next-hop section with reference to RFC6666
 - o Changed last sentence of introduction
 - o Many edits throughout the document
 - o Added definition of tier 1 transit provider
 - o Removed definition of a BGP peering
 - o Removed description of routing policies for IPv6 prefixes in IANA special registry as this now contains a routing scope field
 - o Added reference to RFC6598 and changed the IPv4 prefixes to be filtered by definition section
 - o IXP added in acronym/definition section and only term used throughout the doc now
- 12.4. Diffs between draft-ietf-opsec-bgp-security-01 and draft-ietf-opsec-bgp-security-00

Following changes have been made since previous document draft-ietf-opsec-bgp-security-00:

- o Obsolete RFC2385 moved from normative to informative reference
- o Clarification of preference of TCP-AO over MD5 in former section 4.1
- o Mentioning KARP efforts in TCP session protection section in former section 4 and adding 3 RFC as informative references: 6518, 6862 and 6952
- o Removing reference to SIDR working-group
- o Better dissociating origin validation and path validation to clarify what's potentially available for deployment
- o Adding that SIDR mechanisms should be implemented in addition to the other ones mentioned throughout this document
- o Added a paragraph in former section 8 about ASN renumbering
- o Change of security considerations section
- o Added the newly created IANA IPv4 Special Purpose Address Registry instead of references to RFCs listing these addresses

12.5. Diffs between draft-ietf-opsec-bgp-security-02 and draft-ietf-opsec-bgp-security-01

Following changes have been made since previous document draft-ietf-opsec-bgp-security-01:

- o Added a reference to draft-ietf-sidr-origin-ops
- o Added a reference to RFC6811 and RFC6907
- o Changes "Most of RIR's" to "A majority of RIR's" on IRR availability
- o Various edits
- o Added NIST BGP security recommendations document
- o Added that it's possible to get info from ISPs from RADB
- o Correction of the url for IPv4 special use prefixes repository
- o Clarification of the fact only prefixes with Global Scope set to False MUST be discarded

- o IANA list could be pulled directly by routers (not just pushed on routers).
 - o Warning added when prefixes are checked against IRR
 - o Recommend network operators to sign their routing objects
 - o Recommend network operators to publish their routing objects in IRR of their IRR when available
 - o Dissociate rules for local AS and downstreams in former section 5.1.4
- 12.6. Diffs between draft-ietf-opsec-bgp-security-03 and draft-ietf-opsec-bgp-security-02

Following changes have been made since previous document draft-ietf-opsec-bgp-security-02:

- o Added a note on TCP-AO to be preferred over MD5
- o Mention that loose AS filtering with customers can be combined with precise filters for important ASNs (example those of transits) that are must not be received on theses peers in former section 8.
- o MD5 removed from abstract
- o recommended -> RECOMMENDED where appropriate
- o Reference to BCP38 and BCP84 in former section 4.1
- o Added a note to RFC Editor to remove change section before publication
- o Removal of "future work" section
- o Added rate-limiting in addition to filtering in former section 3
- o Reference to IRRToolSet in former section 5.1.2.3
- o Removed "foreword" section

12.7. Diffs between draft-ietf-opsec-bgp-security-04 and draft-ietf-opsec-bgp-security-03

Following changes have been made since previous document draft-ietf-opsec-bgp-security-03:

- o RFC6890 updates RFC5735
- o RFC6890 updates RFC5156
- o Removed reference RFC2234 and RFC 4234
- o Moved route-server draft into informative reference section

12.8. Diffs between draft-ietf-opsec-bgp-security-05 and draft-ietf-opsec-bgp-security-04

Following changes have been made since previous document draft-ietf-opsec-bgp-security-04:

- o RFC7196 updates draft-ietf-idr-rfd-usable
- o RFC7115 updates draft-ietf-sidr-origin-ops
- o draft-ietf-idr-ix-bgp-route-server-05 updates ietf-idr-ix-bgp-route-server-00

12.9. Diffs between draft-ietf-opsec-bgp-security-06 and draft-ietf-opsec-bgp-security-05

Following changes have been made since previous document draft-ietf-opsec-bgp-security-05:

- o Wording improvements
- o Introduction improved
- o References are expanded (not just reference numbers are displayed but also the title of the document)
- o First occurrence of accronyms expanded
- o GTSM for multi-hop peerings
- o Remove eBGP as protected by BCP38
- o Add a caveat for IPsec for session protection

- o Changed MUST for SHOULD everywhere
- o Small changes in communities section
- o Removed simplified IPv6 prefix list
- o Removed note in section 9 about 32 bits ASN
- o IXP LAN prefix example in appendix
- o Make sure all references are in the text. Most of them were removed as they were initially here for previous version when IANA registries with routing scopes did not exist

13. Acknowledgements

The authors would like to thank the following people for their comments and support: Marc Blanchet, Ron Bonica, Randy Bush, David Freedman, Wesley George, Daniel Ginsburg, David Groves, Mike Hugues, Joel Jaeggli, Tim Kleefass, Warren Kumari, Jacques Latour, Lionel Morand, Jerome Nicolle, Hagen Paul Pfeifer, Thomas Pinaud, Carlos Pignataro, Jean Rebiffe, Donald Smith, Kotikalapudi Sriram, Matjaz Straus, Tony Tauber, Gunter Van de Velde, Sebastian Wiesinger, Matsuzaki Yoshinobu.

Authors would like to thank once again Gunter Van de Velde for presenting the draft at several IETF meetings in various working groups, indeed helping dissemination of this document and gathering of precious feedback.

14. IANA Considerations

This memo includes no request to IANA.

15. Security Considerations

This document is entirely about BGP operational security. It depicts best practices that one should adopt to secure its BGP infrastructure: protecting BGP router and BGP sessions, adopting consistent BGP prefix and AS-path filters and configure other options to secure the BGP network.

On the other hand this document doesn't aim at depicting existing BGP implementations and their potential vulnerabilities and ways they handle errors. It does not detail how protection could be enforced against attack techniques using crafted packets.

16. References

16.1. Normative References

- [1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, <<http://xml.resource.org/public/rfc/html/rfc2119.html>>.
- [2] Rekhter, Y., Li, T., and S. Hares, "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, January 2006.
- [3] Gill, V., Heasley, J., Meyer, D., Savola, P., and C. Pignataro, "The Generalized TTL Security Mechanism (GTSM)", RFC 5082, October 2007.
- [4] Touch, J., Mankin, A., and R. Bonica, "The TCP Authentication Option", RFC 5925, June 2010.
- [5] Mohapatra, P., Scudder, J., Ward, D., Bush, R., and R. Austein, "BGP Prefix Origin Validation", RFC 6811, January 2013.
- [6] Pelsser, C., Bush, R., Patel, K., Mohapatra, P., and O. Maennel, "Making Route Flap Damping Usable", RFC 7196, May 2014.

16.2. Informative References

- [7] Heffernan, A., "Protection of BGP Sessions via the TCP MD5 Signature Option", RFC 2385, August 1998.
- [8] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, May 2000.
- [9] Baker, F. and P. Savola, "Ingress Filtering for Multihomed Networks", BCP 84, RFC 3704, March 2004.
- [10] Blunk, L., Damas, J., Parent, F., and A. Robachevsky, "Routing Policy Specification Language next generation (RPSLng)", RFC 4012, March 2005.
- [11] Dugal, D., Pignataro, C., and R. Dunn, "Protecting the Router Control Plane", RFC 6192, March 2011.
- [12] Lepinski, M. and S. Kent, "An Infrastructure to Support Secure Internet Routing", RFC 6480, February 2012.

- [13] Hilliard, N. and D. Freedman, "A Discard Prefix for IPv6", RFC 6666, August 2012.
- [14] Jethanandani, M., Patel, K., and L. Zheng, "Analysis of BGP, LDP, PCEP, and MSDP Issues According to the Keying and Authentication for Routing Protocols (KARP) Design Guide", RFC 6952, May 2013.
- [15] Bush, R., "Origin Validation Operation Based on the Resource Public Key Infrastructure (RPKI)", BCP 185, RFC 7115, January 2014.
- [16] Kent, S. and A. Chi, "Threat Model for BGP Path Security", RFC 7132, February 2014.
- [17] "Internet Exchange Route Server",
<<http://tools.ietf.org/id/draft-ietf-idr-ix-bgp-route-server-05.txt>>.
- [18] Smith, P. and C. Panigl, "RIPE-378 - RIPE Routing Working Group Recommendations On Route-flap Damping", May 2006.
- [19] Smith, P., Evans, R., and M. Hughes, "RIPE-399 - RIPE Routing Working Group Recommendations on Route Aggregation", December 2006.
- [20] Smith, P. and R. Evans, "RIPE-532 - RIPE Routing Working Group Recommendations on IPv6 Route Aggregation", November 2011.
- [21] Smith, P., Bush, R., Kuhne, M., Pelsser, C., Maennel, O., Patel, K., Mohapatra, P., and R. Evans, "RIPE-580 - RIPE Routing Working Group Recommendations On Route-flap Damping", January 2013.
- [22] "IANA IPv4 Special Purpose Address Registry",
<<http://www.iana.org/assignments/iana-ipv4-special-registry/iana-ipv4-special-registry.xhtml>>.
- [23] "IANA IPv6 Special Purpose Address Registry",
<<http://www.iana.org/assignments/iana-ipv6-special-registry/iana-ipv6-special-registry.xml>>.
- [24] "IANA IPv4 Address Space Registry",
<<http://www.iana.org/assignments/ipv4-address-space/ipv4-address-space.xml>>.

- [25] "IANA IPv6 Address Space Registry",
<<http://www.iana.org/assignments/ipv6-unicast-address-assignments/ipv6-unicast-address-assignments.xml>>.
- [26] "Routing Assets Database", <<http://www.radb.net>>.
- [27] "Security Requirements for BGP Path Validation",
<<http://datatracker.ietf.org/doc/draft-ietf-sidr-bgpsec-reqs/>>.
- [28] "Autonomous System (AS) Migration Features and Their
Effects on the BGP AS_PATH Attribute",
<<http://datatracker.ietf.org/doc/draft-ga-idr-as-migration/>>.
- [29] "IRRToolSet project page", <<http://irrtoolset.isc.org>>.
- [30] Cooper, D., Heilman, E., Brogle, K., Reyzin, L., and S.
Goldberg, "On the Risk of Misbehaving RPKI Authorities",
<<http://www.cs.bu.edu/~goldbe/papers/hotRPKI.pdf>>.

Appendix A. IXP LAN prefix filtering - example

An IXP in the RIPE region is allocated an IPv4 /22 prefix by RIPE NCC (X.Y.0.0/22 in this example) and uses a /23 of this /22 for the IXP LAN (let say X.Y.0.0/23). This IXP LAN prefix is the one used by IXP members to configure eBGP peerings. The IXP could also be allocated an AS number (AS64496 in our example).

Any IXP member SHOULD make sure it filters prefixes more specific than X.Y.0.0/23 from all its eBGP peers. If it received X.Y.0.0/24 or X.Y.1.0/24 this could seriously impact its routing.

The IXP SHOULD originate X.Y.0.0/22 and advertise it to its members through an eBGP peering (most likely from its BGP route servers, configured with AS64496).

The IXP members SHOULD accept the IXP prefix only if it passes the IRR generated filters (see Section 6.1.2.2.1)

IXP members SHOULD then advertise X.Y.0.0/22 prefix to their downstreams. This announce would pass IRR based filters as it is originated by the IXP.

Authors' Addresses

Jerome Durand
CISCO Systems, Inc.
11 rue Camille Desmoulins
Issy-les-Moulineaux 92782 CEDEX
FR

Email: jerduran@cisco.com

Ivan Pepelnjak
NIL Data Communications
Tivolska 48
Ljubljana 1000
Slovenia

Email: ip@ipspace.net

Gert Doering
SpaceNet AG
Joseph-Dollinger-Bogen 14
Muenchen D-80807
Germany

Email: gert@space.net

Secure Inter-Domain Routing
Internet-Draft
Intended status: Standards Track
Expires: October 2, 2013

M. Reynolds
IPSw
S. Kent
BBN
M. Lepinski
BBN
Apr 5, 2013

Local Trust Anchor Management for the Resource Public Key Infrastructure
<draft-ietf-sidr-ltamgmt-08.txt>

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at
<http://www.ietf.org/lid-abstracts.html>

The list of Internet-Draft Shadow Directories can be accessed at
<http://www.ietf.org/shadow.html>

This Internet-Draft will expire on October 2, 2013.

Copyright and License Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Abstract

This document describes a facility to enable a relying party (RP) to manage trust anchors (TAs) in the context of the Resource Public Key Infrastructure (RPKI). It is common in RP software (not just in the RPKI) to allow an RP to import TA material in the form of self-signed certificates. However, this approach to incorporating TAs is potentially dangerous. (These self-signed certificates rarely incorporate any extensions that impose constraints on the scope of the imported public keys, and the RP is not able to impose such constraints.) The facility described in this document allows an RP to impose constraints on such TAs. Because this mechanism is designed to operate in the RPKI context, the most important constraints are the Internet Number Resources (INRs) expressed via RFC 3779 extensions. These extensions bind address spaces and/or autonomous system (AS) numbers to entities. The primary motivation for the facility described in this document is to enable an RP to ensure that INR information that it has acquired via some trusted channel is not overridden by the information acquired from the RPKI repository system or by the putative TAs that the RP imports. Specifically, the mechanism allows an RP to specify a set of overriding bindings between public key identifiers and INR data. These bindings take precedence over any conflicting bindings acquired by the putative TAs and the certificates downloaded from the RPKI repository system. This mechanism is designed for local use by an RP, but any entity that is accorded administrative control over a set of RPs may use this mechanism to convey its view of the RPKI to RPs within its jurisdiction. The means by which this latter use case is effected is outside the scope of this document.

Table of Contents

1	Introduction	4
1.1	Terminology	5
2	Overview of Certificate Processing	5
2.1	Target Certificate Processing	5
2.2	Perforation	5
2.3	TA Re-parenting	6
2.4	Paracertificates	6
3	Format of the constraints file	8
3.1	Relying party subsection	8
3.2	Flags subsection	8
3.3	Tags subsection	9
3.3.1	Xvalidity_dates tag	10
3.3.2	Xcrl_dp tag	10
3.3.3	Xcp tag	11
3.3.4	Xaia tag	11
3.4	Blocks subsection	12
4	Certificate Processing Algorithm	13
4.1	Proofreading algorithm	14
4.2	TA processing algorithm	15
4.2.1	Preparatory processing (stage 0)	16
4.2.2	Target processing (stage 1)	17
4.2.3	Ancestor processing (stage 2)	18
4.2.4	Tree processing (stage 3)	19
4.2.5	TA re-parenting (stage 4)	20
4.3	Discussion	21
5	Implications for Path Discovery	21
5.1	Two answers	21
5.2	One answer	22
5.3	No answer	22
6	Implications for Revocation	22
6.1	No state bits set	23
6.2	ORIGINAL state bit set	23
6.3	PARA state bit set	23
6.4	Both ORIGINAL and PARA state bits set	24
7	Security Considerations	24
8	IANA Considerations	24
9	Acknowledgements	24
10	References	24
10.1	Normative References	24
10.2	Informative References	25
	Authors' Addresses	25
	Appendix A: Sample Constraints File	26
	Appendix B: Optional Sorting Algorithm for Ancestor Processing	27

1 Introduction

The Resource Public Key Infrastructure (RPKI) [RFC6480] is a PKI in which certificates are issued to facilitate management of Internet Resource Numbers (INRs). Such resources are expressed in the form of X.509v3 "resource" certificates with extensions defined by RFC 3779 [RFC6487]. Validation of a resource certificate is preceded by path discovery. In a PKI path discovery is effected by constructing a certificate path between a target certificate and a trust anchor (TA). No IETF standards define how to construct a certificate path; commonly such paths are based on a bottom-up search using Subject/Issuer name matching, but top-down and meet-in-the-middle approaches may also be employed [RFC4158]. In contrast, path validation is top-down, as defined by [RFC5280].

In the RPKI, certificates can be acquired in various ways, but the default is a top-down tree walk as described in [RFC6481], initialized via a Trust Anchor Locator [RFC6490]. Note that the process described there is not path discovery per se but the collecting of certificates to populate a local cache. Thus, the common, bottom-up path discovery approach is not inconsistent with these RFCs. Moreover, a bottom-up path discovery approach is more general, accommodating certificates that might be acquired by other means, i.e., not from an RPKI repository. There are circumstances under which an RP may wish to override the INR specifications obtained through the RPKI distributed repository system [RFC6481]. This document describes a mechanism by which an RP may override any conflicting information expressed via putative TAs and the certificates downloaded from the RPKI repository system. Thus the algorithms described in this document adopt a bottom-up path discovery approach.

To effect this local control, this document calls for a relying party to specify a set of bindings between public key identifiers and INRs through a text file known as a constraints file. The constraints expressed in this file then take precedence over any competing claims expressed by resource certificates acquired from the distributed repository system. (The means by which a relying party acquires the key identifier and the RFC 3779 extension data used to populate the constraints file is outside the scope of this document.) The relying party also may use a local publication point (the root of a local directory tree that is made available as if it were a remote repository) as a source of certificates and CRLs (and other RPKI signed objects, e.g., ROAs and manifests) that do not appear in the RPKI repository system.

In order to allow reuse of existing, standard path validation mechanisms, the RP-imposed constraints are realized by having the RP itself represented as the only TA known in the local certificate validation context. To ensure that all RPKI certificates can be validated relative to this TA, this RP TA certificate must contain all-encompassing resource allocations, i.e. 0/0 for IPv4, 0::/0 for IPv6 and 0-4294967295 for AS numbers. Thus, a conforming implementation of this mechanism must be able to cause a self-signed certification authority (CA) certificate to be created with a locally generated key pair. It also must be able to issue CA certificates subordinate to this TA. Finally, a conforming implementation of this

mechanism must process the constraints file and modify certificates as needed in order to enforce the constraints asserted in the file.

The remainder of this document describes in detail the types of certificate modification that may occur, the syntax and semantics of the constraints file, and the implications of certificate modification on path discovery and revocation.

1.1 Terminology

It is assumed that the reader is familiar with the terms and concepts described in "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile" [RFC5280] and "X.509 Extensions for IP Addresses and AS Identifiers" [RFC3779].

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119.

2 Overview of Certificate Processing

The fundamental aspect of the facility described in this document is one of certificate modification. The constraints file, described in more detail in the next section, contains assertions about INRs that are to be specially processed. As a result of this processing, certificates in the local copy of the RPKI repository are transformed into new certificates satisfying the INR constraints so specified. This enables the RP to override conflicting assertions about resource holdings as acquired from the RPKI repository system. Three forms of certificate modification can occur. (Every certificate is digitally signed and thus cannot be modified without "breaking" its signature. In the context of this document we assume that certificates that are modified have been validated previously. Thus the content can be modified, locally, without the need to preserve the integrity of the signature. These modified certificates are referred to as paracertificates (see section 2.4 below).)

2.1 Target Certificate Processing

If a certificate is acquired from the RPKI repository system and its Subject key identifier (SKI) is listed in the constraints file, it will be reissued directly under the RP TA certificate, with (possibly) modified RFC 3779 extensions. (The SKI is used as a compact reference to the public key in a target certificate.) The modified extensions will include any RFC 3779 data expressed in the constraints file. Other certificate fields may also be modified to maintain consistency. (These fields are enumerated in Table 1, and discussed in Section 3.3.) In Section 4.2, target certificate processing corresponds to stage one of the algorithm. (When a target certificate is re-parented, all subordinate signed products will still be valid, unless the set of INRs in the targeted certificate is reduced.)

2.2 Perforation

When a target certificate is re-issued directly under the RP's TA, its INRs MUST be removed from all of its parent (CA) certificates. (If these INRs were not removed, then conflicting assertions about INRs could arise and undermine the authority of the RP TA.) Thus, every

certificate acquired from the RPKI repository MUST be examined to determine if it contains an RFC 3779 extension that intersects the resource data in the constraints file. If there is an intersection the certificate will be reissued directly under the RP TA, with modified RFC 3779 extensions. We refer to the process of modifying the RFC 3779 extension in an affected certificate as "perforation" (because the process will create "holes" in these extensions). The

modified extensions will exclude any RFC 3779 data expressed in the constraints file. In the certificate processing algorithm described in Section 4.2, perforation corresponds to stage two of the algorithm ("ancestor processing") and also to stage three of the algorithm ("tree processing").

2.3 TA Re-parenting

All valid, self-signed certificates offered as TAs in the public RPKI certificate hierarchy, e.g., self-signed certificates issued by IANA or RIRs, will be re-issued under the RP TA certificate. This processing is done even though all but one of these certificates might not intersect any resources specified in the constraints file. We refer to this reissuance as "re-parenting" since the issuer (parent) of the certificate has been changed. The issuer name is changed from that of the certificate subject (this is a self-signed certificate) to that of the RP TA. In the certificate processing algorithm described in Section 4.2, TA re-parenting corresponds to stage four of the algorithm. (In a more generic PKI context, re-parenting enables an RP to insert extensions in these certificates to impose constraints on path processing in a fashion consistent with RFC 5280. In this fashion an RP can impose name constraints, policy constraints, etc.)

2.4 Paracertificates

If a certificate is subject to any of the three forms of processing just described, that certificate will be referred to as an "original" certificate and the processed (output) certificate will be referred to as a paracertificate. When an original certificate is transformed into a paracertificate all the fields and extensions from the original certificate will be retained, except as indicated in Table 1, below.

Original Certificate Field	Action
Version	unchanged
Serial number	created per note A
Signature	replaced if needed with RP's signing alg
Issuer	replaced with RP's name
Validity dates	replaced per note B
Subject	unchanged
Subject public key info	unchanged
Extensions	
Subject key identifier	unchanged
Key usage	unchanged
Basic constraints	unchanged
CRL distribution points	replaced per note B
Certificate policy	replaced per note B
Authority info access	replaced per note B
Authority key ident	replaced with RP's
IP address block	modified as described
AS number block	modified as described
Subject info access	unchanged
All other extensions	unchanged
Signature Algorithm	same as above
Signature value	new

Table 1 Certificate Field Modifications

Note A. The serial number will be created by concatenating the current time (the number of seconds since Jan 1, 1970) with a count of the certificates created in the current run. Because all paracertificates are issued directly below the RP TA, this algorithm ensures serial number uniqueness.

Note B. These fields are derived (as described in Section 3.3 below) from parameters in the constraints file (if present); otherwise, they take on values from the certificates from which the paracertificates are derived.

3 Format of the constraints file

This section describes the syntax of the constraints file. (The syntax has been defined to enable creation and distribution of constraint files to a set of RPs, by an authorized third party.) The model described below is nominal; implementations need not match all details of this model as presented, but the external behavior of implementations MUST correspond to the externally observable characteristics of this model in order to be compliant. It is RECOMMENDED that the syntax described herein be supported, to facilitate interoperability between creators and consumers of constraints files.

The constraints file consists of four logical subsections: the relying party subsection, the flags subsection, the tags subsection and the blocks subsection. The relying party subsection and the blocks subsection are REQUIRED and MUST be present; the flags and tags subsections are OPTIONAL. Each subsection is described in more detail below. Note that the semicolon (;) character acts as the comment character, to enable annotating constraints files. All characters from a semicolon to the end of that line are ignored. In addition, lines consisting only of whitespace are ignored. The subsections MUST occur in the order indicated. An example constraints file is given in Appendix A.

3.1 Relying party subsection

The relying party subsection is a REQUIRED subsection of the constraints file. It MUST be the first subsection of the constraints file, and it MUST consist of two lines of the form:
(RECOMMENDED)

```
PRIVATEKEYMETHOD    value [ ... value ]
TACERTIFICATE        value
```

The first line provides a pointer (including an access method) to the RP's private key. This line consists of the string literal PRIVATEKEYMETHOD, followed by one or more whitespace delimited string values. These values are passed to the certificate processing algorithm as described below. Note that this entry, as for all entries in the constraints file, is case sensitive.

The second line of this subsection consists of the string literal TACERTIFICATE, followed by exactly one string value. This value is the name of a file containing the relying party's TA certificate. The file name is passed to the certificate processing algorithm as described below.

3.2 Flags subsection

The flags subsection of the constraints file is an OPTIONAL subsection. If present it MUST immediately follow the relying party

subsection. The flags subsection consists of one or more lines of the form

```
CONTROL  flagname  booleanvalue
```

Each such line is referred to as a control line. Each control line MUST contain exactly three whitespace delimited strings. The first string MUST be the literal CONTROL. The second string MUST be one of the following three literals:

```
resource_nounion
intersection_always
treegrowth
```

The third string denotes a Boolean value, and MUST be one of the literals TRUE or FALSE. Control flags influence the global operation of the certificate processing algorithm; the semantics of the flags is described in Section 4.2. Note that each flag has a default value, so that if the corresponding CONTROL line does not appear in the constraints file, the algorithm flag is considered to take the corresponding default value. The default value for each flag is FALSE. Thus, if any flag is not named in a control line it takes the value FALSE. If the flags subsection is absent, all three flags assume the default value FALSE.

3.3 Tags subsection

The tags subsection is an OPTIONAL subsection in the constraints file. If present it MUST immediately follow the relying party subsection (if the flags subsection is absent) or the flags subsection (if it is present). The tags subsection consists of one or more lines of the form

```
TAG  tagname  tagvalue [ ... tagvalue ]
```

Each such line is referred to as a tag line. Each tag line MUST consist of at least three whitespace delimited string values, the first of which must be the literal TAG. The second string value gives the name of the tag, and subsequent string(s) give the value(s) of the tag. The tag name MUST be one of the following four string literals:

```
Xvalidity_dates
Xcrldp
Xcp
Xaia
```

The purpose of the tag lines is to provide an indication of the means

by which paracertificate fields, specifically those indicated above under "Note B", of Table 1 are constructed. Each tag has a default, so that if the corresponding tag line is not present in the constraints file, the default behavior is used when constructing the paracertificates. The syntax and semantics of each tag line is described next.

Note that the tag lines are considered to be global; the action of each tag line (or the default action, if that tag line is not present) applies to all paracertificates that are created as part of the certificate processing algorithm.

3.3.1 Xvalidity_dates tag

This tag line is used to control the value of the notBefore and notAfter fields in paracertificates. If this tag line is specified and there is a single tagvalue which is the literal string C, the paracertificate validity interval is copied from the original certificate validity interval from which it is derived. If this tag is specified and there is a single tagvalue which is the literal string R, the paracertificate validity interval is copied from the validity interval of the RP's TA certificate. If this tag is specified and the tagvalue is neither of these literals, then exactly two tagvalues MUST be specified. Each must be a Generalized Time string of the form YYYYMMDDHHMMSSZ. The first tagvalue is assigned to the notBefore field and the second tagvalue is assigned to the notAfter field. It MUST be the case that the tagvalues can be parsed as valid Generalized Time strings such that notBefore is less than notAfter, and also such that notAfter represents a time in the future (i.e., the paracertificate has not already expired).

If this tag line is not present in the constraints file the default behavior is to copy the validity interval from the original certificate to the corresponding paracertificate.

3.3.2 Xcrl_dp tag

This tag line is used to control the value of the CRL distribution point extension in paracertificates. If this tag line is specified and there is a single tagvalue that is the string literal C, the CRLDP of the paracertificate is copied from the CRLDP of the original certificate from which it is derived. If this tag line is specified and there is a single tagvalue that is the string literal R, the CRLDP of the paracertificate is copied from the CRLDP of the RP's TA certificate. If this tag line is specified and there is a single tagvalue that is not one of these two reserved literals, or if there is more than one tagvalue, then each tagvalue is interpreted as a URI that will be placed in the CRLDP sequence in the

paracertificate.

If this tag line is not present in the constraints file the default behavior is to copy the CRLDP from the original certificate into the corresponding paracertificate.

3.3.3 Xcp tag

This tag line is used to control the value of the policyQualifierId field in paracertificates. If this tag line is specified there MUST be exactly one tagvalue. If the tagvalue is the string literal C, the paracertificate value is copied from the value in the corresponding original certificate. If the tagvalue is the string literal R, the paracertificate value is copied from the value in the RP's top level TA certificate. If the tagvalue is the string literal D, the paracertificate value is set to the default OID. If the tagvalue is not one of these reserved string literals, then the tagvalue MUST be an OID specified using the standard dotted notation. The value in the paracertificate's policyQualifierId field is set to this OID. Note the RFC 5280 specifies that only a single policy may be specified in a certificate, so only a single tagvalue is permitted in this tag line, even though the CertificatePolicy field is an ASN.1 sequence.

If this tag line is not specified the default behavior is to use the default OID in creating the paracertificate.

This option permits the RP to convert a value of the policyQualifierId field in a certificate (that would not be in conformance with the RPKI CP) to a conforming value in the paracertificate. This conversion enables use of RPKI validation software that checks the policy field against that specified in the RPKI CP [RFC6484].

3.3.4 Xaia tag

This tag line is used to control the value of the Authority Information Access (AIA) extension in the paracertificate. If this tag line is present then it MUST have exactly one tagvalue. If this tagvalue is the string literal C, then the AIA field in the paracertificate is copied from the AIA field in the original certificate from which it is derived. If this tag line is present and the tagvalue is not the reserved string literal, then the tagvalue MUST be a URI. This URI is set as the AIA extension of the paracertificates that are created.

If this tag line is not specified the default behavior is to use copy the AIA field from the original certificate to the AIA field of the paracertificate.

3.4 Blocks subsection

The blocks subsection is a REQUIRED subsection of the constraints file. If the tags subsection is present, the blocks subsection MUST appear immediately after it. This MUST be the last subsection in the constraints file. The blocks subsection consists of one or more blocks, known as target blocks. A target block is used to specify an association between a certificate (identified by an SKI) and a set of resource assertions. Each target block contains four regions, an SKI region, an IPv4 region, an IPv6 region and an AS number region. All regions MUST be present in a target block.

The SKI region contains a single line beginning with the string literal SKI and followed by forty hexadecimal characters giving the subject key identifier of a certificate, known as the target certificate. The hex character string MAY contain embedded whitespace or colon characters (included to improve readability), which are ignored. The IPv4 region consists of a line containing only the string literal IPv4. This line is followed by zero or more lines containing IPv4 prefixes in the format described in RFC 3779. The IPv6 region consists of a line containing only the string literal IPv6, followed by zero or more lines containing IPv6 prefixes using the format described in RFC 3513. (The presence of the IPv4 and IPv6 literals is to simplify parsing of the constraints file.) Finally, the AS number region consists of a line containing only the string literal AS#, followed by zero or more lines containing AS numbers (one per line). The AS numbers are specified in decimal notation as recommended in RFC 5396. A target block is terminated by either the end of the constraints file, or by the beginning of the next target block, as signaled by its opening SKI region line. An example target block is shown below. (The indentation used below is employed to improve readability and is not required.) See also the complete constraints file example in Appendix A. Note that whitespace, as always, is ignored.

```
SKI 00:12:33:44:00:BA:BA:DE:EB:EE:00:99:88:77:66:55:44:33:22:11
IPv4
  10.2.3/24
  10.8/16
IPv6
  1:2:3:4:5:6/112
AS#
  123
  567
```

The blocks subsection MUST contain at least one target block. Note that it is OPTIONAL that the SKI refer to a certificate that is known

or resolvable within the context of the local RPKI repository. Also, there is no REQUIRED or implied ordering of target blocks within the block subsection. Since blocks may occur in any order, the outcome of processing a constraints file may depend on the order in which target blocks occur within the constraints file. The next section of this document contains a detailed description of the certificate processing algorithm.

4 Certificate Processing Algorithm

The section describes the certificate processing algorithm by which paracertificates are created from original certificates in the local RPKI repository. For the purposes of describing this algorithm, it will be assumed that certificates are persistently associated with state (or metadata) information. This state information is nominally represented by an array of named bits associated with each certificate. No specific implementation of this functionality is mandated by this document. Any implementation that provides the indicated functionality is acceptable, and need not actually consist of a bit field associated with each certificate.

The following state bits used in certificate processing are

- NOCHAIN
- ORIGINAL
- PARA
- TARGET

If the NOCHAIN bit is set, this indicates that a full path between the given certificate and a TA has not yet been discovered. If the ORIGINAL bit is set, this indicates that the certificate in question has been processed by some part of the processing algorithm described in Section 4.2. If it was processed as part of stage one processing, as described in section 4.2.2, the TARGET bit also will be set. Finally, every paracertificate will have the PARA bit set.

At the beginning of algorithm processing each certificate in the local RPKI repository has the ORIGINAL, PARA and TARGET bits clear. If a certificate has a complete, validated path to a TA, or is itself a TA, then that certificate will have the NOCHAIN bit clear, otherwise it will have the NOCHAIN bit set. As the certificate processing algorithm proceeds, the metadata state of original certificates may change. In addition, since the certificate processing algorithm may also be creating paracertificates, it is responsible for actively setting or clearing the state of these four bits on those paracertificates.

The certificate processing algorithm consists of two sub-algorithms:

"proofreading" and "TA processing". Conceptually, the proofreading algorithm performs syntactic checks on the constraints file, while the TA processing algorithm performs the actual certificate transformation processing. If the proofreading algorithm does not succeed in parsing the constraints file, the TA processing-algorithm is not executed. Note also that if the constraints file is not present, neither algorithm is executed and the local RPKI repository is not modified. Each of the constituent algorithms will now be described in detail.

4.1 Proofreading algorithm

The proofreading algorithm checks the constraints file for syntactic errors, e.g., missing REQUIRED subsections, or malformed addresses. Implementation of this algorithm is OPTIONAL. If it is implemented, the following text defines correct operation for the algorithm. The proofreading algorithms performs a set of heuristic checks, such as checking for prefixes that are too large (e.g., larger than /8). The proofreading algorithm also SHOULD examine resource regions (IPv4, IPv6 and AS# regions) within the blocks subsection, and reorder such resources within a region in ascending numeric order. On encountering any error the proofreading algorithm SHOULD provide an error message indicating the line on which the error occurred as well as informative text that is sufficiently descriptive as to allow the user to identify and correct the error. An implementation of the proofreading algorithm MUST NOT assume that it has access to the local RPKI repository (even read-only access). An implementation of the proofreading algorithm MUST NOT alter the local RPKI repository in any way; it also MUST NOT change any of the metadata associated with certificates in that repository. (Recall that the processing described here is creating a copy of that local repository.) For simplicity the remainder of this document assumes that the proofreading algorithm produces a transformed output file. This file contains the same syntactic information as the text version of the constraints file.

The proofreading algorithm performs the following syntactic checks on the constraints file:

- verifies the presence of the REQUIRED relying party subsection and the REQUIRED blocks subsection.
- verifies the order of the two, three or four subsections as stated above.
- verifies that the relying party subsection conforms to the specification given in Section 3.1 above.
- verifies that, if present, the tags and flags subsections conform to the specifications in Sections 3.2 and 3.3 above.

After these checks have been performed, the proofreading algorithm then checks the blocks subsection:

- splits the blocks subsection into constituent target blocks, as delimited by the SKI region line(s)
- verifies that at least one target block is present
- verifies that each SKI region line contains exactly forty hexadecimal digits and contains no additional characters other than whitespace or colon characters.

For each target the proofreading algorithm:

- verifies the presence of the IPv4, IPv6 and AS# regions, and verifies that at least one such resource is present.
- verifies that, for each IPv4 prefix, IPv6 prefix and autonomous system number given, that the indicated resource is syntactically valid according to the appropriate RFC definition, as described in Section 3.4.
- verifies that no IPv4 resource has a prefix larger than /8.
- optionally performing reordering within each of the three resource regions so that stated resources occur in ascending numerical order.

(If the proofreading algorithm has performed any reordering of information it MAY overwrite the constraints file. If it does so, however, it MUST preserve all information contained within the file, including information that is not parsed (such as comments). If the proofreading algorithm has performed any reordering of information but has not overwritten the constraints file, it MAY produce a transformed output file, as described above. If the proofreading algorithm has performed any reordering of information, but has neither overwritten the constraints file nor produced a transformed output file, it MUST provide an error message to the user indicating what reordering was performed.)

4.2 TA processing algorithm

The TA processing algorithm acts on the constraints file (as processed by the proofreading algorithm) and the contents of the local RPKI repository to produce paracertificates for the purpose of enforcing the resource allocations as expressed in the constraints file. The TA processing algorithm operates in five stages, a preparatory stage (stage 0), target processing (stage 1), ancestor processing (stage 2), tree processing (stage 3) and TA re-parenting (stage 4). Conceptually, during the preparatory stage the proofreader output file is read and a set of internal RP, tag and flag variables are set based on the contents of that file. (If the constraint file has not specified one or more of the tags and/or flags, those tags and flags are set to default values.) During target processing all certificates specified by a target block are processed, and the resources for those certificates are (potentially) expanded; for each target found a new paracertificate is manufactured with its various fields set, as shown in Table 1, using the values of the internal variables set in the preparatory stage and also, of course, the fields of the original certificate (and, potentially, fields of the RP's TA certificate). In stage 2 (ancestor) processing, all ancestors of the each target certificate are found, and the claimed resources are then removed (perforated). A new paracertificate with these diminished resources is crafted, with its fields generated based on internal variable settings, original certificate field values, and, potentially, the fields of the RP's TA certificate. In tree processing (stage 3), the

entire local RPKI repository is searching for any other certificates that have resources that intersect a target resource, and that were not otherwise processed during a preceding stage. Perforation is again performed for any such intersecting certificates, and paracertificates created as in stage 2. In the fourth (last) stage, TA re-parenting, any TA certificates in the local RPKI repository that have not already been processed are now re-parented under the RP's TA certificate. This transformation creates paracertificates; however, these paracertificates may have RFC 3779 resources that were not altered during algorithm processing. The final output of algorithm processing will be threefold:

- the metadata information on some (original) certificates in the repository MAY be altered.
- paracertificates will be created, with the appropriate metadata, and entered into the repository.
- the TA processing algorithm SHOULD produce a human readable log of its actions, indicating which paracertificates were created and why. The remainder of this section describes the processing stages of the algorithm in detail.

4.2.1 Preparatory processing (stage 0)

During preparatory processing, the output of the proofreader algorithm, is read. Internal variables are set corresponding to each tag and flag, if present, or to their defaults, if absent. Internal variables are set corresponding to the PRIVATEKEYMETHOD value string(s) and the TACERTIFICATE string. The TA processing algorithm is queried to determine if it supports the indicated private key access methodology. This query is performed in an implementation-specific manner. In particular, an implementation is free to vacuously return success to this query. The TA processing algorithm next uses the value string for the TACERTIFICATE to locate this certificate, again in an implementation-specific manner. The certificate in question may already be present in the local RPKI repository, or it may be located elsewhere. The implementation is free to create the top level certificate at this time, and then assign to this newly-created certificate the name indicated. It is necessary only that, at the conclusion of this processing, a valid trust anchor certificate for the relying party has been created or otherwise obtained.

Some form of access to the RP's private key and top level certificate are required for subsequent correct operation of the algorithm. Therefore, stage 0 processing MUST terminate if one or both conditions are not satisfied. In the error case, the implementation SHOULD provide an error message of sufficient detail that the user can correct the error(s). If stage 0 processing does not succeed, no further stages of TA processing are executed.

4.2.2 Target processing (stage 1)

During target processing, the TA processing algorithm reads all target blocks in the proofreader output file. It then processes each target block in the order specified in the file. In the description that follows, except where noted, the operation of the algorithm on a single target block will be described. Note, however, that all stage 1 processing is executed before any processing in subsequent stages is performed.

The algorithm first obtains the SKI region of the target block. It then locates (in an implementation-dependent manner) the certificate identified by the SKI. Note that this search is performed only against (original) certificates, not against paracertificates. If more than one original certificate is found matching this SKI, there are two possible scenarios. If a resource holder has two certificates issued by the same CA, with overlapping validity intervals and the same key, but distinct subject names (typically, by virtue of the SerialNumber parts being different), then these two certificates are both considered to be (distinct) targets, and are both processed. If, however, a resource holder has certificates issued by two different CAs, containing different resources, but using the same key, there is no unambiguous method to decide which of the certificates is intended as the target. In this latter case the algorithm MUST issue a warning to that effect, mark the target block in question as unavailable for processing by subsequent stages and proceed to the next target block. If no certificate is found then the algorithm SHOULD issue a warning to that effect and proceed to process the next target block.

If a single (original) certificate is found matching the indicated SKI, then the algorithm takes the following actions. First, it sets the ORIGINAL state bit for the certificate found. Second, it sets the TARGET state bit for the certificate found. Third, it extracts the INRs from the certificate. If the global resource_nounion flag is TRUE, the algorithm compares the extracted certificate INRs with the INRs specified in the constraints file. If the two resource sets are different, the algorithm SHOULD issue a warning noting the difference. An output resource set is then formed that is identical to the resource set extracted from the certificate. If, however, the resource_nounion flag is FALSE, then the output resource set is calculated by forming the union of the resources extracted from the certificate and the resources specified for this target block in the constraints file. A paracertificate is then constructed according to Table 1, using fields from the original certificate, the tags that had been set during

stage 0, and, if necessary, fields from the RP's TA certificate. The INR resources of the paracertificate are equated to the derived output resource set. The PARA state bit is set for the newly created paracertificate.

4.2.3 Ancestor processing (stage 2)

The goal of ancestor processing is to discover all ancestors of a target certificate and remove from those ancestors the resources specified in the target blocks corresponding to the targets being processed. Note that it is possible that, for a given chain from a target certificate to a trust anchor, another target might be encountered. This is handled by removing all the target resources of all descendants. The set of all targets that are descendants of the given certificate is formed. The union of all the target resources of the corresponding target blocks is computed, and this union is then removed from the shared ancestor.

In detail, the algorithm is as follows. First, all (original) target certificates processed during stage 1 processing are collected. Second, any collected certificates that have the NOCHAIN state bit set are eliminated from the collection. (Note that, as a result of eliminating such certificates, the resulting collection may be empty, in which case this stage of algorithm processing terminates, and processing advances to stage 3.) Next, an implementation MAY sort the collection. The optional sorting algorithm is described in Appendix B. Note that all stage 2 processing is completed before any stage 3 processing.

Two levels of nested iteration are performed. The outer iteration is effected over all certificates in the collection; the inner iteration is over all ancestors of the designated certificate being processed. The first certificate in the collection is chosen, and a resource set R is initialized based on the resources of the target block for that certificate (since the certificate is in the collection, it must be a target certificate, and thus correspond to a target block). The parent of the certificate is then located using ordinary path discovery over original certificates only. The ancestor's certificate resources A are then extracted. These resources are then perforated with respect to R. That is, an output set of resources is created by forming the intersection I of A and R, and then taking the set difference $A - I$ as the output resources. A paracertificate is then created containing resources that are these output resources, and containing other fields and extensions from the original certificate (and possibly the RP's TA certificate) according to the procedure given in Table 1. The PARA state bit is set on this paracertificate and the ORIGINAL state bit is set on A. If A is also a target certificate, as indicated by its TARGET state bit being set, then

there will already have been a paracertificate created for it. This previous paracertificate is destroyed in favor of the newly created paracertificate. In this case also, the set R is augmented by adding into it the set of resources of the target block for A. The algorithm then proceeds to process the parent of A. This inner iteration continues until the self-signed certificate at the root of the path is encountered and processed. The outer iteration then continues by clearing R and proceeding to the next certificate in the target collection.

Note that ancestor processing has the potential for order dependency, as mentioned earlier in this document. If sorting is not implemented, or if the sorting algorithm fails to completely process the collection of target certificates because the allotted maximum number of iterations has been realized, it may be the case that an ancestor of a certificate logically occurs before that certificate in the collection. Whenever an existing paracertificate is replaced by a newly created paracertificate during ancestor processing, the algorithm SHOULD alert the user, and SHOULD log sufficient detail such that the user is able to determine which resources were perforated from the original certificate in order to create the (new) paracertificate.

In addition, implementations MUST provide for conflict detection and notification during ancestor processing. During ancestor processing a certificate may be encountered two or more times and the modifications dictated by the ancestor processing algorithm may be in conflict. If this situation arises the algorithm MUST refrain from processing that certificate. Further, the implementation MUST present the user with an error message that contains enough detail so that the user can locate those directives in the constraints file that are creating the conflict. For example, during one stage of the processing algorithm it may be directed that resources R1 be added to a certificate C, while during a different stage of the processing algorithm it may be directed that resources R2 be removed from certificate C. If the resource sets R1 and R2 have a non-empty intersection, that is a conflict.

4.2.4 Tree processing (stage 3)

The goal of tree processing is to locate other certificates containing INRs that conflict with the resources allocated to a target, by virtue of the INRs specified in the constraints file. The certificates processed are not ancestors of any target. The algorithm used is described below.

First, all target certificates are collected. Second, all target certificates that have the NOCHAIN state bit set are eliminated from this collection. Third, if the intersection_always

global flag is set, target blocks that occur in the constraints file, but that did not correspond to a certificate in the local repository, are added to the collection. In tree processing, unlike ancestor processing, this collection is not sorted. An iteration is now performed over each certificate (or set of target block resources) in the collection. Note that the collection may be empty, in which case this stage of algorithm processing terminates, and processing advances to stage 4. Note also that all stage 3 processing is performed before any stage 4 processing.

Given a certificate or target resource block, each top level original TA certificate is examined. If that TA certificate has an intersection with the target block resources, then the certificate is perforated with respect to those resources. A paracertificate is created based on the contents of the original certificate (and possibly the RP's TA certificate, as indicated in Table 1) using the perforated resources. The ORIGINAL state bit is set on the original certificate processed in this manner, and the PARA state bit is set on the paracertificate just created. An inner iteration then begins on the descendants of the original certificate just processed. There are two ways in which this iteration may proceed. If the treegrowth global flag is clear, then examination of the children proceeds until all children are exhausted, or until one child is found with intersecting resources. If the treegrowth global flag is set, all children are examined. If a transfer of resources is in process, more than one child may possess intersecting resources. In this case, it is RECOMMENDED that the treegrowth flag be set. The inner iteration proceeds until all descendants have been examined and no further intersecting resources are found. The outer iteration then continues with the next certificate or target resource block in the collection. Note that unlike ancestor processing, there is no concept of a potentially cumulating resource collection R; only the resources in the target block are used for perforation.

4.2.5 TA re-parenting (stage 4)

In the final stage of TA algorithm processing, all TA certificates (other than the RP's TA certificate) that have not already been processed are now processed. At this stage all unprocessed TA certificates have no intersection with any target resource blocks. As such, in creating the corresponding paracertificates, the output resource set is identical to the input resource set. Other transformations as described in Table 1 are performed. The original TA certificates have the ORIGINAL state bit set; the newly created paracertificates have the PARA state bit set. Note that once stage four processing is completely, only a single TA certificate will remain in an unprocessed state, namely the relying party's own TA certificate.

4.3 Discussion

The algorithm described in this document effectively creates two coexisting certificate hierarchies: the original certificate hierarchy and the paracertificate hierarchy. Original certificates are not removed during any of the processing described in the previous section. Some original certificates may move from having no state bits set (or only the NOCHAIN state bit set) to having one or both of the ORIGINAL and TARGET state bits set. In addition, the NOCHAIN state bit will still be set if it was set before any processing. The paracertificate hierarchy, however, is intended to supersede the original hierarchy for ROA validation. The presence of two hierarchies has implications for path discovery, and for revocation.

If one thinks of a certificate as being "named" by its SKI, then there can now be two certificates with the same name, an original certificate and a paracertificate. The next two sections discuss the implications of this duality in detail. Before proceeding, it is worth noting that even without the existence of the paracertificate hierarchy, cases may exist in which two or more original certificates have the same SKI. As noted earlier, in Section 4.2.2, these cases may be subdivided into the case in which such certificates are distinguishable by virtue of having different subject names, but identical issuers and resource sets, versus all other cases. In the distinguishable case, the path discovery algorithm treats the original certificates as separate certificates, and processes them separately. In all other cases, the original certificates should be treated as indistinguishable, and path validation should fail.

5 Implications for Path Discovery

Path discovery proceeds from a child certificate C by asking for a parent certificate P such that the AKI of C is equal to the SKI of P. With one hierarchy this question would produce at most one answer. With two hierarchies, the original certificate hierarchy and the paracertificate hierarchy, the question may produce two answers, one answer, or no answer. Each of these cases is considered in turn.

5.1 Two answers

If two paths are discovered, it SHOULD be the case that one of the matches is a certificate with the ORIGINAL state bit set and the PARA state bit clear, while the other match inversely has the ORIGINAL state bit clear and the PARA state bit set. If any other combination of ORIGINAL and PARA state bits obtains, the path discovery algorithm MUST alert the user. In addition, the path discovery algorithm SHOULD refrain from attempting to make a

choice as to which of the two certificates is the putative parent. In the no-error case, with the state bits as indicated, the certificate with the PARA state bit set is chosen as the parent P. Note this means, in effect, that all children of the original certificate have been re-parented under the paracertificate.

5.2 One answer

If the matching certificate has neither the ORIGINAL state bit set nor the PARA state bit set, this certificate is the parent. If the matching certificate has the PARA state bit set but the ORIGINAL state bit not set, this certificate is the parent. (This situation would arise, for example, if the original certificate had been revoked by its issuer but the paracertificate had not been revoked by the RP.) If the matching certificate has the ORIGINAL state bit set but the PARA state bit not set, this is not an error but it is a situation in which path discovery MUST be forced to fail. The parent P MUST be set to NULL, and the NOCHAIN state bit must be set on C and all its descendants; the user SHOULD be warned. Even if the RP has revoked the paracertificate, the original certificate MAY persist. Forcing path discovery to unsuccessfully terminate is a reflection of the RP's preference for path discovery to fail as opposed to using the original hierarchy. Finally, if the matching certificate has both the ORIGINAL and PARA state bits set, this is an error. The parent P MUST be set to NULL, and the user MUST be warned.

5.3 No answer

This situation occurs when C has no parent in either the original hierarchy or the paracertificate hierarchy. In this case the parent P is NULL and path discovery terminates unsuccessfully. The NOCHAIN state bit must be set on C and all its descendants.

6 Implications for Revocation

In a standard implementation of revocation in a PKI, a valid CRL names a (sibling) certificate by serial number. That certificate is revoked and is purged from the local RPKI repository. The original certificate hierarchy and the paracertificate hierarchy created by applying the algorithms described above are closely related. It can thus be asked how revocation is handled in the presence of these two hierarchies. In particular do changes in one of the hierarchies trigger corresponding changes in the other hierarchy. There are four cases based on the state of the ORIGINAL and PARA bits. These are discussed in the subsections below. It should be noted that the existence of two hierarchies presents a particular challenge with respect to revocation. If a CRL arrives and is processed, that

processing can result in the destruction of one of the path chains. In the case of a single hierarchy this would mean that certain objects would fail to validate. In the presence of two hierarchies, however, a CRL revocation may force the preferred path to be destroyed. If the RP later determines that the CRL revocation should not have occurred, he is faced with an undesirable situation: the deprecated path will be discovered. In order to prevent this outcome, an RP MUST be able to configure one or more additional repository URIs in support of local trust anchor management.

6.1 No state bits set

If the CRL names a certificate that has neither the ORIGINAL state bit set nor the PARA state bit set, revocation proceeds normally. All children of the revoked certificate have their state modified so that the NOCHAIN state bit is set.

6.2 ORIGINAL state bit set

If the CRL names a certificate with the ORIGINAL state bit set and the PARA state bit clear, then this certificate is revoked as usual. If this original certificate also has the TARGET state bit set, then the corresponding paracertificate (if it exists) is not revoked; if this original certificate has the TARGET state bit clear, then the corresponding paracertificate is revoked as well. Note that since all the children of the original certificate have been re-parented to be children of the corresponding paracertificate, as described above, the revocation algorithm MUST NOT set the NOCHAIN state bit on these children unless the paracertificate is also revoked. Note also that if the original certificate is revoked but the paracertificate is not revoked, the paracertificate retains its PARA state bit. This is to ensure that path discovery proceeds preferentially through the paracertificate hierarchy, as described above.

6.3 PARA state bit set

If the CRL names a certificate with the PARA state bit set and the ORIGINAL state bit clear, this CRL must have been issued, perforce, by the RP itself. This is because all the paracertificates are children of the RP's TA certificate. (Recall that a TA is not revoked via a CRL; it is merely removed from the repository.) The paracertificate is revoked and all children of the paracertificate have the NOCHAIN state bit set. No action is taken on the corresponding original certificate; in particular, its ORIGINAL state bit is not cleared.

Note that the serial numbers of paracertificates are synthesized according to the procedure given in Table 1, rather than being assigned by an algorithm under the control of the (original) issuer.

6.4 Both ORIGINAL and PARA state bits set

This is an error. The revocation algorithm MUST alert the user and take no further action.

,

7 Security Considerations

The goal of the algorithm described in this document is to enable an RP to impose its own view of the RPKI, which is intrinsically a security function. An RP using a constraints file is trusting the assertions made in that file. Errors in the constraints file used by an RP can undermine the security offered by the RPKI, to that RP. In particular, since the paracertificate hierarchy is intended to trump the original certificate hierarchy for the purposes of path discovery, an improperly constructed paracertificate hierarchy could validate ROAs that would otherwise be invalid. It could also declare as invalid ROAs that would otherwise be valid. As a result, an RP must carefully consider the security implications of the constraints file being used, especially if the file is provided by a third party.

8 IANA Considerations

[Note to IANA, to be removed prior to publication: there are no IANA considerations stated in this version of the document.]

9 Acknowledgements

The authors would like to acknowledge the significant contributions of Charles Gardiner, who was the original author of an internal version of this document, and who contributed significantly to its evolution into the current version.

10 References

10.1 Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3513] Hinden, R., and S. Deering, "Internet Protocol Version 6 (IPv6) Addressing Architecture", RFC 3513, April 2003.
- [RFC3779] Lynn, C., Kent, S., and K. Seo, "X.509 Extensions for IP Addresses and AS Identifiers", RFC 3779, June 2004.
- [RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key

Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, May 2008.

- [RFC5396] Huston, G., and G. Michaelson, "Textual Representation of Autonomous System (AS) Numbers", RFC 5396, December 2008.
- [RFC6480] Lepinski, M. and S. Kent, "An Infrastructure to Support Secure Internet Routing", RFC 6480, February 2012.
- [RFC6481] Huston, G., Loomans, R., and G. Michaelson, "A Profile for Resource Certificate Policy Structure", RFC 6481, February 2012.
- [RFC6487] Huston, G., Michaelson, G., and R. Loomans, "A Profile for X.509 PKIX Resource Certificates", RFC 6487, February 2012.

10.2 Informative References

None.

Authors' Addresses

Stephen Kent
Raytheon BBN Technologies
10 Moulton St.
Cambridge, MA 02138

Email: kent@bbn.com

Matthew Lepinski
Raytheon BBN Technologies
10 Moulton St.
Cambridge, MA 02138

Email: mlepinsk@bbn.com

Mark C. Reynolds
Island Peak Software
328 Virginia Road
Concord, MA 01742

Email: mcr@islandpeaksoftware.com

Appendix A: Sample Constraints File

```
;
; Sample constraints file for TBO LTA Test Corporation.
;
; TBO manages its own local (10.x.x.x) address space
; via the target blocks in this file.
;

;
; Relying party subsection. TBO uses ssh-agent as
; a software cryptographic agent.
;

PRIVATEKEYMETHOD      OBO(ssh-agent)
TACERTIFICATE          tbomaster.cer

;
; Flags subsection
;
; Always use the resources in this file to augment
; certificate resources.
; Always process resource conflicts in the tree, even
; if the target certificate is missing.
; Always search the entire tree.
;

CONTROL  resource_nounion      FALSE
CONTROL  intersection_always   TRUE
CONTROL  treegrowth            TRUE

;
; Tags subsection
;
; Copy the original cert's validity dates.
; Use the default policy OID.
; Use our own CRLDP.
; Use our own AIA.
;

TAG      Xvalidity_dates      C
TAG      Xcp                  D
TAG      Xcrl dp              rsync://tbo_lta_test.com/pub/CRLs
TAG      Xaia                  rsync://tbo_lta_test.com/pub/repos

;
; Block subsection
;
```

```
;
; First block: TBO Corporate
;

; Resource Holder: TBO Corporation

SKI 00112233445566778899998877665544332211
  IPv4
    10.2.3/24
    10.8/16
  IPv6
    2001:db8::/32
  AS#
    60123
    5507

;

; Second block: TBO LTA Test Enforcement Division
;

; Resource Holder: TBO Corporation

SKI 653420AF758421CF600029FF857422AA6833299F
  IPv4
    10.2.8/24
    10.47/16
  IPv6
  AS#
    60124

;

; Third block: TBO LTA Test Acceptance Corporation
; Quality financial services since sometime
; late yesterday.
;

; Resource Holder: TBO Acceptance Corporation

SKI 19:82:34:90:8b:a0:9c:ef:00:af:a0:98:23:09:82:4b:ef:ab:98:09
  IPv4
    10.3.3/24
  IPv6
  AS#
    60125

; End of TBO constraints file
```

Appendix B: Optional Sorting Algorithm for Ancestor Processing

Sorting is performed in an effort to eliminate any order dependencies in ancestor processing, as described in section 4.2.3 of this

document. The sorting algorithm does this by rearranging the processing of certificates such that if A is an ancestor of B, B is processed before A. The sorting algorithm is an OPTIONAL part of ancestor processing. Sorting proceeds as follows. The collection created at the beginning of ancestor processing is traversed and any certificate in the collection that is visited as a result of path discovery is temporarily marked. After the traversal, all unmarked certificates are moved to the beginning of the collection. The remaining marked certificates are unmarked, and a traversal again performed through this sub-collection of previously marked certificates. The sorting algorithm proceeds iteratively until all certificates have been sorted or until a predetermined fixed number of iterations has been performed. (Eight is suggested as a munificent value for the upper bound, since the number of sorting steps need not be any greater than the maximum depth of the tree.) Finally, the ancestor processing algorithm is applied in turn to each certificate in the remaining sorted collection. If the sorting algorithm fails to converge, that is if the maximum number of iterations has been reached and unsorted certificates remain, the implementation SHOULD warn the user.

Network Working Group
Internet-Draft
Updates: RFC 6490 (if approved)
Intended status: Standards Track
Expires: November 23, 2013

R. Gagliano
Cisco Systems
T. Manderson
ICANN
C. Martinez Cagnazzo
LACNIC
May 22, 2013

Multiple Repository Publication Points support in the Resource Public
Key Infrastructure (RPKI)
draft-ietf-sidr-multiple-publication-points-00

Abstract

The Resource Public Key Infrastructure (RPKI) depends on Relying Parties (RP) ability to access its Trust Anchors' certificate specified in the different "Trust Anchor Locator (TAL)" files and the Repository Objects located at the Certificate Authorities (CA) repositories hosted in its respective publication point. This document updates [RFC6490] by allowing multiple URI associated to a single public key in a TAL file and introduces the concept of multiple repository publication point operators for every CA in the RPKI. This document provides also recommendation for the RP behavior when analyzing signed objects that include multiple publications points.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on November 23, 2013.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Requirements notation	3
2. Introduction	4
3. Multiple Operators support in TAL files	6
3.1. Update to RFC 6490 Section 2.1	6
3.2. Rules for Relying Parties (RP)	7
4. Multiple Operators support in Certificates	8
4.1. Rules for Relying Parties (RP)	8
5. IANA Considerations	9
6. Security Considerations	10
7. Acknowledgements	11
8. Normative References	12
Authors' Addresses	13

1. Requirements notation

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

2. Introduction

The RPKI repository system described in [RFC6481] requires scalability and diversity in order to address challenges such as Distributed Denial of Service (DDoS) attacks, to secure the availability of the system when performing maintenance activities and against possible security incidents in one particular implementation. Additionally, when a single operator manages a RPKI Repository Publication Point, it is more probable to introduce circular dependencies when the Route Origin Authorization (ROA) signed objects for the Repository Publication Point IP addresses are hosted in servers that uses those same addresses.

The current toolset for a CA to diversify its repository system is limited for both TA distribution and CA publication point management. In the case of trust anchors, [RFC6490] requires a unique URI per key on each TAL file. Conversely, in the case of the different publication points and although supported by [RFC6487], there is no current guidance on how RPs should support multiple publication points for the same object.

When using a single URI, the options for diversity and scalability are reduced to:

1. Give the content to a Content Delivery Network (CDN) to have the content distributed (as long as the CDN supports the CA's access method, which is not currently the case for rsync). The implementation will typically require the configuration of a CNAME resource record in the authoritative server pointing to a server farm inside the CDN who will handle load-balancing by using a set of internally defined metrics. If, for the sake of diversity, a CA administrator would like to use two different CDNs for the same URI it will need to modify the authoritative name server behavior to break RFC1034 standard behavior and allow multiple CNAME records for the same alias. This modification is not available by default on most of the more widely deployed DNS servers.
2. Copy the content to different Repository Publication Points around the globe (i.e. using [I-D.ietf-sidr-publication]) and load balance the content using different Domain Name System (DNS) techniques. The load balancing implementation will need to verify the availability of the target server before providing a DNS response to avoid blackholes caused by unavailable servers or clusters. This "feature" needs also be added to the authoritative name server or the full DNS resolution or outsourced to a third party (which would introduce another non-diversified element).

This document addresses this problem by enabling multiple operators for trust anchor material, and, while not making it mandatory, recommends the use of multiple publication points in signed objects.

The main idea is that the a CA will host its RPKI signed objects in different locations, using diverse routing paths and diverse DNS resolution. The RP will have more processing to perform to fetch the different objects when dealing with exceptions.

The first thing that is needed is to add multiple URIs support for each Trust Anchor. [RFC6490] requires that each TAL file includes a unique URI. This document removes this requirement by allowing one or more URI for each public key in a TAL file. In steady state, an RP should receive the same material from each of the different URI for the same root certificate. An exception could happen when the certificate is been updated or rolled-over, a process which should not have operational consequences.

For the root certificate trust anchor, this proposal has an additional consequence: it would create the idea of root-CA repository operators. This concept has worked well in the case of DNS, where one organization is responsible for creating the root zone material and a number of different organizations are responsible in running the root servers.

A CA can add support for multiple Repository Publication Points operators by adding more than one respective object for the Authority Information Access (AIA), the Subject Information Access (SIA) and the CRL Distribution Points (CRLDP) and which is supported by [RFC5280] and [RFC6487] . This document provides guidance on the RP expected behavior when analyzing signed objects with multiple Repository Publication Points in Section 4.

3. Multiple Operators support in TAL files

The idea of multiples operators support for a TA certificate expressed on its TAL file is similar to the support for several Root Server operators in a DNS hints file.

An example of such a TAL file with 3 operators would be:

```
rsync://rpki.operator1.org/rpki/hedgehog/root.cer
rsync://rpki.operator2.net/rpki/hedgehog/root.cer
rsync://rpki.operator3.biz/rpki/hedgehog/root.cer
```

```
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAOvWQL2lh6knDx
GUG5hbtCXvvh4AOzjhDkSHlj22gn/loiM9IeDATIwP44vhQ6L/xvuk7W6
Kfa5ygmqQ+xOZOWTWPcrUbqaQyPNxokuivzyvqVZVDecOEqs78q58mSp9
nbtxmLRW7B67SJCBSzfa5XpVyXYEgYAjk3fpmefU+AcxtxvvHB5OVPIa
BfPcs80ICMgHQX+fphvute9XLxjfJKJWkhZqZ0v7pZm2uhkcPx1PMGcrG
ee0WSDC3fr3erLueagpiLsFjwwpX6F+Ms8vqz45H+DKmYKvPSstZjCCq9
aJ0qANT9OtnfSDOS+aLRPjZryCNyvvBHxZXqj5YCGKtwIDAQAB
```

As we can see in this example, a RP would have different URI where to fetch the self-signed certificate for the trust anchor. In each location, the same result should be expected as all the URI share the same public key.

In order to increase diversity, It is RECOMMENDED that the different FQDN could be resolved to IP addresses included in ROA objects from different CAs and hosted in diverse repository publication points.

3.1. Update to RFC 6490 Section 2.1

The following text will replace the last paragraph on Section 2.1 of RFC 6490:

The TAL is an ordered sequence of:

- 1) One or more rsync URI [RFC5781],
- 2) A <CRLF> or <LF> line break after each URI,
- 3) A line containing a single <CRLF> or <LF> line break, and
- 4) A subjectPublicKeyInfo [RFC5280] in DER format [X.509], encoded in Base64 (see Section 4 of [RFC4648]).A

3.2. Rules for Relying Parties (RP)

A RP can use different rules to select the URI from where fetch the Trust Anchor certificate. Some examples are:

- o Using the order provided in the TAL file
- o Selecting the URI randomly from the available list
- o Creating a prioritized list of URIs based on RP specific parameters such as connection establishment delay

If the connection to the preferred URI fails or the fetched certificate public key does not match the TAL public key, the RP SHOULD fetch the TA certificate from the next URI of preference.

4. Multiple Operators support in Certificates

The support for multiple operators in the RPKI Certificate Authority (CA) and End Entity (EE) certificates is supported as the RFC 5082 allows multiple repository publication point operators as the SIA, AIA and CRLDP are implemented as sequences. Consequently, no changes are needed on the existing RPKI standard and this section could be considered informative.

In the case of the SIA extension, for each operator, the `accessMethods` for both the CA repository publication point and for the correspondent manifest needs to be added.

4.1. Rules for Relying Parties (RP)

A RP can use different rules to select the URI to fetch the different repository objects and when performing the validation.

When a RP needs to fetch one or more object from a list of possible URIs, it can chose the URI by adopting a locally defined rule that could be:

- o Using the order provided in the correspondent certificate
- o Selecting the URI randomly from the available list
- o Creating a prioritized list of URIs based on RP specific parameters such as connection establishment delay

If the connection to the preferred URI fails , the RP SHOULD fetch the repository objects from the next URI of preference.

5. IANA Considerations

No IANA requirements

6. Security Considerations

TBA

7. Acknowledgements

TBA.

8. Normative References

- [I-D.ietf-sidr-publication]
"A Publication Protocol for the Resource Public Key Infrastructure (RPKI)", <<http://www.ietf.org/id/draft-ietf-sidr-publication-02.txt>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, May 2008.
- [RFC6481] Huston, G., Loomans, R., and G. Michaelson, "A Profile for Resource Certificate Repository Structure", RFC 6481, February 2012.
- [RFC6484] Kent, S., Kong, D., Seo, K., and R. Watro, "Certificate Policy (CP) for the Resource Public Key Infrastructure (RPKI)", BCP 173, RFC 6484, February 2012.
- [RFC6485] Huston, G., "The Profile for Algorithms and Key Sizes for Use in the Resource Public Key Infrastructure (RPKI)", RFC 6485, February 2012.
- [RFC6487] Huston, G., Michaelson, G., and R. Loomans, "A Profile for X.509 PKIX Resource Certificates", RFC 6487, February 2012.
- [RFC6490] Huston, G., Weiler, S., Michaelson, G., and S. Kent, "Resource Public Key Infrastructure (RPKI) Trust Anchor Locator", RFC 6490, February 2012.
- [RFC6492] Huston, G., Loomans, R., Ellacott, B., and R. Austein, "A Protocol for Provisioning Resource Certificates", RFC 6492, February 2012.

Authors' Addresses

Roque Gagliano
Cisco Systems
Avenue des Uttins 5
Rolle, 1180
Switzerland

Email: rogaglia@cisco.com

Terry Manderson
ICANN

Email: terry.manderson@icann.org

Carlos Martinez Cagnazzo
LACNIC

Email: carlos@lacnic.net

Network Working Group
Internet-Draft
Updates: RFC 6490 (if approved)
Intended status: Standards Track
Expires: August 18, 2014

R. Gagliano
Cisco Systems
T. Manderson
ICANN
C. Martinez Cagnazzo
LACNIC
February 14, 2014

Multiple Repository Publication Points support in the Resource Public
Key Infrastructure (RPKI)
draft-ietf-sidr-multiple-publication-points-01

Abstract

The Resource Public Key Infrastructure (RPKI) depends on Relying Parties (RP) ability to access its Trust Anchors' certificate specified in the different "Trust Anchor Locator (TAL)" files and the Repository Objects located at the Certificate Authorities (CA) repositories hosted in its respective publication point. This document updates [RFC6490] by allowing multiple URI associated to a single public key in a TAL file and introduces the concept of multiple repository publication point operators for every CA in the RPKI. This document provides also recommendation for the RP behavior when analyzing signed objects that include multiple publications points.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 18, 2014.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Requirements notation	3
2. Introduction	4
3. Multiple Operators support in TAL files	6
3.1. Update to RFC 6490 Section 2.1	6
3.2. Rules for Relying Parties (RP)	7
4. Multiple Operators support in Certificates	8
4.1. Rules for Relying Parties (RP)	8
5. IANA Considerations	9
6. Security Considerations	10
7. Acknowledgements	11
8. Normative References	12
Authors' Addresses	13

1. Requirements notation

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

2. Introduction

The RPKI repository system described in [RFC6481] requires scalability and diversity in order to address challenges such as Distributed Denial of Service (DDoS) attacks, to secure the availability of the system when performing maintenance activities and against possible security incidents in one particular implementation. Additionally, when a single operator manages a RPKI Repository Publication Point, it is more probable to introduce circular dependencies when the Route Origin Authorization (ROA) signed objects for the Repository Publication Point IP addresses are hosted in servers that uses those same addresses.

The current toolset for a CA to diversify its repository system is limited for both TA distribution and CA publication point management. In the case of trust anchors, [RFC6490] requires a unique URI per key on each TAL file. Conversely, in the case of the different publication points and although supported by [RFC6487], there is no current guidance on how RPs should support multiple publication points for the same object.

When using a single URI, the options for diversity and scalability are reduced to:

1. Give the content to a Content Delivery Network (CDN) to have the content distributed (as long as the CDN supports the CA's access method, which is not currently the case for rsync). The implementation will typically require the configuration of a CNAME resource record in the authoritative server pointing to a server farm inside the CDN who will handle load-balancing by using a set of internally defined metrics. If, for the sake of diversity, a CA administrator would like to use two different CDNs for the same URI it will need to modify the authoritative name server behavior to break RFC1034 standard behavior and allow multiple CNAME records for the same alias. This modification is not available by default on most of the more widely deployed DNS servers.
2. Copy the content to different Repository Publication Points around the globe (i.e. using [I-D.ietf-sidr-publication]) and load balance the content using different Domain Name System (DNS) techniques. The load balancing implementation will need to verify the availability of the target server before providing a DNS response to avoid blackholes caused by unavailable servers or clusters. This "feature" needs also be added to the authoritative name server or the full DNS resolution or outsourced to a third party (which would introduce another non-diversified element).

This document addresses this problem by enabling multiple operators for trust anchor material, and, while not making it mandatory, recommends the use of multiple publication points in signed objects.

The main idea is that the a CA will host its RPKI signed objects in different locations, using diverse routing paths and diverse DNS resolution. The RP will have more processing to perform to fetch the different objects when dealing with exceptions.

The first thing that is needed is to add multiple URIs support for each Trust Anchor. [RFC6490] requires that each TAL file includes a unique URI. This document removes this requirement by allowing one or more URI for each public key in a TAL file. In steady state, an RP should receive the same material from each of the different URI for the same root certificate. An exception could happen when the certificate is been updated or rolled-over, a process which should not have operational consequences.

For the root certificate trust anchor, this proposal has an additional consequence: it would create the idea of root-CA repository operators. This concept has worked well in the case of DNS, where one organization is responsible for creating the root zone material and a number of different organizations are responsible in running the root servers.

A CA can add support for multiple Repository Publication Points operators by adding more than one respective object for the Authority Information Access (AIA), the Subject Information Access (SIA) and the CRL Distribution Points (CRLDP) and which is supported by [RFC5280] and [RFC6487] . This document provides guidance on the RP expected behavior when analyzing signed objects with multiple Repository Publication Points in Section 4.

3. Multiple Operators support in TAL files

The idea of multiples operators support for a TA certificate expressed on its TAL file is similar to the support for several Root Server operators in a DNS hints file.

An example of such a TAL file with 3 operators would be:

```
rsync://rpki.operator1.org/rpki/hedgehog/root.cer
rsync://rpki.operator2.net/rpki/hedgehog/root.cer
rsync://rpki.operator3.biz/rpki/hedgehog/root.cer
```

```
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAovWQL2lh6knDx
GUG5hbtCXvvh4AOzjhDkSHlj22gn/loiM9IeDATIwP44vhQ6L/xvuk7W6
Kfa5ygmqQ+xOZOwTWPcrUbqaQyPNxokuivzyvqVZVDecOEqs78q58mSp9
nbtxmLRW7B67SJCBSzfa5XpVyXYEgYAJkk3fpmefU+AcctxvvHB5OVPIa
BfPcs80ICMgHQX+fphvute9XLxjfJKJWkhZqZ0v7pZm2uhkcPx1PMGcrG
ee0WSDC3fr3erLueagpiLsFjwwpX6F+Ms8vqz45H+DKmYKvPSstZjCCq9
aJ0qANT90tnfSDOS+aLRPjZryCNyvvBHxZXqj5YCGKtwIDAQAB
```

As we can see in this example, a RP would have different URI where to fetch the self-signed certificate for the trust anchor. In each location, the same result should be expected as all the URI share the same public key.

In order to increase diversity, It is RECOMMENDED that the different FQDN could be resolved to IP addresses included in ROA objects from different CAs and hosted in diverse repository publication points.

3.1. Update to RFC 6490 Section 2.1

The following text will replace the last paragraph on Section 2.1 of RFC 6490:

The TAL is an ordered sequence of:

- 1) One or more rsync URI [RFC5781],
- 2) A <CRLF> or <LF> line break after each URI,
- 3) A line containing a single <CRLF> or <LF> line break, and
- 4) A subjectPublicKeyInfo [RFC5280] in DER format [X.509], encoded in Base64 (see Section 4 of [RFC4648]).A

3.2. Rules for Relying Parties (RP)

A RP can use different rules to select the URI from where fetch the Trust Anchor certificate. Some examples are:

- o Using the order provided in the TAL file
- o Selecting the URI randomly from the available list
- o Creating a prioritized list of URIs based on RP specific parameters such as connection establishment delay

If the connection to the preferred URI fails or the fetched certificate public key does not match the TAL public key, the RP SHOULD fetch the TA certificate from the next URI of preference.

4. Multiple Operators support in Certificates

The support for multiple operators in the RPKI Certificate Authority (CA) and End Entity (EE) certificates is supported as the RFC 5082 allows multiple repository publication point operators as the SIA, AIA and CRLDP are implemented as sequences. Consequently, no changes are needed on the existing RPKI standard and this section could be considered informative.

In the case of the SIA extension, for each operator, the accessMethods for both the CA repository publication point and for the correspondent manifest needs to be added.

4.1. Rules for Relying Parties (RP)

A RP can use different rules to select the URI to fetch the different repository objects and when performing the validation.

When a RP needs to fetch one or more object from a list of possible URIs, it can chose the URI by adopting a locally defined rule that could be:

- o Using the order provided in the correspondent certificate
- o Selecting the URI randomly from the available list
- o Creating a prioritized list of URIs based on RP specific parameters such as connection establishment delay

If the connection to the preferred URI fails , the RP SHOULD fetch the repository objects from the next URI of preference.

5. IANA Considerations

No IANA requirements

6. Security Considerations

TBA

7. Acknowledgements

TBA.

8. Normative References

- [I-D.ietf-sidr-publication]
"A Publication Protocol for the Resource Public Key Infrastructure (RPKI)", <<http://www.ietf.org/id/draft-ietf-sidr-publication-02.txt>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, May 2008.
- [RFC6481] Huston, G., Loomans, R., and G. Michaelson, "A Profile for Resource Certificate Repository Structure", RFC 6481, February 2012.
- [RFC6484] Kent, S., Kong, D., Seo, K., and R. Watro, "Certificate Policy (CP) for the Resource Public Key Infrastructure (RPKI)", BCP 173, RFC 6484, February 2012.
- [RFC6485] Huston, G., "The Profile for Algorithms and Key Sizes for Use in the Resource Public Key Infrastructure (RPKI)", RFC 6485, February 2012.
- [RFC6487] Huston, G., Michaelson, G., and R. Loomans, "A Profile for X.509 PKIX Resource Certificates", RFC 6487, February 2012.
- [RFC6490] Huston, G., Weiler, S., Michaelson, G., and S. Kent, "Resource Public Key Infrastructure (RPKI) Trust Anchor Locator", RFC 6490, February 2012.
- [RFC6492] Huston, G., Loomans, R., Ellacott, B., and R. Austein, "A Protocol for Provisioning Resource Certificates", RFC 6492, February 2012.

Authors' Addresses

Roque Gagliano
Cisco Systems
Avenue des Uttins 5
Rolle, 1180
Switzerland

Email: rogaglia@cisco.com

Terry Manderson
ICANN

Email: terry.manderson@icann.org

Carlos Martinez Cagnazzo
LACNIC

Email: carlos@lacnic.net

