SIDR S. Kent
Internet-Draft BBN Technologies
Intended status: Informational D. Ma
Expires: January 26, 2017 ZDNS
July 25, 2016

Adverse Actions by a Certification Authority (CA) or Repository Manager in the Resource Public Key Infrastructure (RPKI)

draft-ietf-sidr-adverse-actions-01

Abstract

This document analyzes actions by or against a CA or independent repository manager in the RPKI that can adversely affect the Internet Number Resources (INRs) associated with that CA or its subordinate CAs. The analysis is based on examination of the data items in the RPKI repository, as controlled by a CA (or independent repository manager) and fetched by Relying Parties (RPs). The analysis is performed from the perspective of an affected INR holder. The analysis does not purport to be comprehensive; it does represent an orderly way to analyze a number of ways that errors by or attacks against a CA or repository manager can affect the RPKI and routing decisions based on RPKI data.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

In the context of this document, any change to the Resource Public Key Infrastructure (RPKI) [RFC6480] that results in a diminution of the set of Internet Numeric Resources (INRs) associated with an INR holder contrary to the holder's wishes is termed "adverse". Additionally, when a ROA or router certificate is created that "competes" with an existing ROA or router certificate (respectively), the creation of the new ROA or router certificate may be adverse. (A newer ROA competes with an older ROA if the newer ROA points to a different ASN, contains the same or a more specific prefix, and is issued by a different CA. A newer router certificate competes with an older router certificate if the newer one contains the same ASN a different public key, and is issued by a different CA.) Note that

transferring resources, or changing of upstream providers may yield competing ROAs and/or router certificates, under some circumstances. Thus not all instances of competition are adverse actions.

The SIDR WG is exploring proposals to address mistakes by RPKI Certification Authorities (CAs) and to accommodate INR transfers. Mistakes are not the only way that adverse changes to RPKI data can arise. A CA or repository operator might be subject to an attack [RFC7132]. For a CA, if an attack allows an adversary to use the private keys of that CA to sign RPKI objects, then the effect is analogous to the CA making mistakes. There is also the possibility that a CA or repository operator may be subject to legal measures that compel them to generate "bogus" signed objects or remove legitimate repository data. In many cases, such actions may be hard to distinguish from non-malicious mistakes, other than with respect to the time required to remedy the adverse action. Thus this document examines the implications of adverse actions with respect to INRs irrespective of the cause of the actions. Standards for transfer of INRs are being explored in [I-D.ymbk-sidr-transfer]. That work also motivates exploration of the impact of both legitimate transfers and botched transfer attempts on the RPKI.

This document analyzes the various types of actions by a CA (or independent repository manager) that can adversely affect the INRs associated with that CA, as well as the INRs of subordinate CAs. The analysis is based on examination of the data items in the RPKI repository, as controlled by a CA (or independent repository manager) and fetched by Relying Parties (RPs). The analysis is done from the perspective of an affected INR holder.

1.1. Terminology

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. Analysis of RPKI Repository Objects

This section enumerates the RPKI repository system objects and examines how changes to them affect Route Origination Authorizations (ROAs) and router certificate validation. Identifiers are assigned to errors for reference by later sections of this document. Note that not all adverse actions may be addressed by this taxonomy.

The RPKI repository [RFC6481] contains a number of (digitally signed) objects that are fetched and processed by RPs. Until the deployment of BGPsec [I-D.ietf-sidr-bgpsec-overview], the principal goal of the RPKI is to enable an RP to validate ROAs [RFC6482]. A ROA binds

address space to an Autonomous System Number (ASN). A ROA can be used to verify BGP announcements with respect to route origin [RFC6483]. The most important objects in the RPKI for origin validation are ROAs; all of the other RPKI objects exist to enable the validation of ROAs in a fashion consistent with the INR allocation system. Thus errors that result in changes to a ROA, or to RPKI objects needed to validate a ROA, can cause RPs to reach different (from what was intended) conclusions about the validity of the bindings expressed in a ROA.

When BGPsec is deployed, router certificates [I-D.ietf-sidr-bgpsec-pki-profiles] will be added to repository publication points. These are End-Entity (EE) certificates used to verify signatures applied to BGP update data, to enable path validation [I-D.ietf-sidr-bgpsec-protocol]. Router certificates are as important to path validation as ROAs are to origin validation.

The objects contained in the RPKI repository are of two types: conventional PKI objects (certificates and Certificate Revocation Lists (CRLs)) and RPKI-specific signed objects. The latter make use of a common encapsulation format [RFC6488] based on the Cryptographic Message Syntax (CMS) [RFC5652]. A syntax error in this common format will cause an RP to reject the object, e.g., a ROA or Manifest, as invalid.

Adverse actions take several forms:

- Deletion (D) is defined as removing an object from a publication point, without the permission of the INR holder.

- Suppression (S) is defined as not deleting an object, or not publishing an object, as intended by an INR holder. This action also includes retaining a prior version of an object in a publication point when a newer version is available for publication.

- Corruption (C) is defined as modification of a signed object in a fashion not requiring access to the private key used to sign the object. Thus a corrupted object will not carry a valid signature. Implicitly, the corrupted object replaces the legitimate version.

- Modification (M) is defined as publishing a syntactically valid, verifiable version of an object that differs from the (existing) version authorized by the INR holder. Implicitly,

the legitimate version of the affected object is deleted and replaced by the modified object.

- Revocation (R) is defined as revoking a certificate (EE or CA) by placing its serial number on the appropriate CRL, without authorization of the INR holder.

- Injection (I) is defined as introducing an instance of a signed object into a publication point (without authorization of the INR holder). It assumes that the signature on the object will be viewed as valid by RPs.

The first three of these actions (deletion, suppression, and corruption) can be effected by any entity that manages the publication point of the affected INR holder. Also, an entity with the ability to act as a man-in-the-middle between an RP and a repository can effect these actions with respect to the RP in question.

The latter three actions (modification, revocation, and injection) nominally require access to the private key of the INR holder.

All six of these actions also can be effected by a parent CA. A parent CA could reissue the INR holder's CA certificate, but with a different public key, matching a private key to which the parent CA has access. The CA could generate new signed objects using the private key associated with the reissued certificate, and publish these objects at a location of its choosing.

Most of these actions may be performed independently or in combination with one another. For example, a ROA may be revoked and deleted or revoked and replaced with a modified ROA. Where appropriate, the analysis of adverse actions will distinguish between individual actions, or combinations thereof, that yield different outcomes for RPs. Recall that the focus of the analysis is the impact on ROAs and router certificates, with respect to RP processing.

The following sections examine how the actions enumerated above affect objects in the RPKI repository system. Each action is addressed in order (Deletion, Suppression, Corruption, Modification, Revocation, and Injection) for each object, making it easy to see how each action has been considered with regard to each object. (For the GhostBusters record we condensed the discussion of the actions because the impact is the same in each case.)

2.1. CA Certificates

Every INR holder is represented by one or more CA certificates. An INR holder has multiple CA certificates if it holds resources acquired from different sources. Also, every INR holder has more than one CA certificate during key rollover [RFC6489] and algorithm rollover [RFC6916].

If a publication point is not a leaf in the RPKI hierarchy, then the publication point will contain one or more CA certificates, each representing a subordinate CA. Each subordinate CA certificate contains a pointer (SIA) to the publication point where the signed objects associated with that CA can be found [RFC6487].

A CA certificate is a complex data structure and thus errors in that structure may have different implications for RPs depending on the specific data that is in error.

Adverse actions against a CA certificate can cause the following errors:

A-1.1 Deletion

A-1.1.1 Deletion of a CA certificate would cause an RP to not be able to locate signed objects generated by that CA, except those that have been cached by the RP. Thus an RP would be unaware of changed or new (issued after the cached data) INR bindings asserted in subordinate ROAs, and the RP would be unable to validate new or changed router certificates. If the missed objects were intended to replace ROAs or router certificates prior to expiration, then when those objects expire, RPs may cease to view them as valid. As a result, valid routes may be viewed as NotFound or Invalid.

A-1.2 Suppression

A-1.2.1 If publication of a CA certificate is suppressed, the impact depends on what changes appeared in the suppressed certificate. If the SIA value changed, the effect would be the same as in A-1.1 or 1.4.3. If the 3779 extensions in the suppressed certificate changed, the impact would be the same as in 1.4.1. If the AIA extension changed in the

suppressed certificate, the impact would be the same as in 1.4.4.

A-1.3 Corruption

A-1.3.1 Corruption of a CA certificate will cause it to be rejected by RPs. In turn, this may cause subordinate signed objects to become invalid. An RP that has cached the subtree under the affected CA certificate may continue to view it as valid, until objects expire. But changed or new objects might not be retrieved, depending on details of the design of the RP software. Thus this action may be equivalent to suppressing changes to the affected subtree.

A-1.4 Modification

A-1.4.1 If a CA certificate is modified, but still conforms to the RPKI certificate profile [RFC6485], it will be accepted by RPs. If an [RFC3779] extension in this certificate is changed to exclude INRs that were previously present, then subordinate signed objects will become invalid if they rely on the excised INRs. If these objects are CA certificates, their subordinate signed objects will be treated as invalid. If the objects are ROAs, the binding expressed by the affected ROAs will be ignored by RPs. If the objects are router certificates, BGPsec_Path attributes [I-D.ietf-sidr-bgpsec-protocol] verifiable under these certificates will be considered invalid.

A-1.4.2 If the SIA extension of a CA certificate is modified to refer to another publication point, this will cause an RP to look at another location for subordinate objects. This could cause RPs to not acquire the objects that the INR holder intended to be retrieved - manifests, ROAs, router certificates, Ghostbuster records, or any subordinate CA certificates associated with that CA. If the objects at this new location contain invalid signatures or appear to be corrupted, they may be rejected. In this case, cached versions of the objects may be viewed as valid by an RP, until they expire. If the objects at the new location have valid signatures and pass path validation

checks, they will replace the cached objects, effectively replacing the INR holder's objects.

A-1.4.3 If the AIA extension in a CA certificate is modified, it would point to a different CA certificate, not the parent CA certificate. This extension is used only for path discovery, not path validation. Path discovery in the RPKI is usually performed on a top-down basis, starting with TAs and recursively descending the RPKI hierarchy. Thus there may be no impact on the ability of clients to acquire and validate certificates if the AIA is modified.

A-1.4.4 If the Subject Public Key Info (and Subject Key Identifier extension) in a CA certificate is modified to contain a public key corresponding to a private key held by the parent, the parent could sign objects as children of the affected CA certificate. With this capability, the parent could replace the INR holder, issuing new signed objects that would be accepted by RPs (as long as they do not violate the path validation criteria). This would enable the parent to effect modification, revocation, and injection actions against all of the objects under the affected CA certificate, including subordinate CA certificates. (Note that key rollover also yields a new CA certificate. However, the new certificate will co-exist with the old one for a while, which may help distinguish this legitimate activity from an adverse action.)

A-1.5 Revocation

A-1.5.1 If a CA certificate is revoked an RP will treat as invalid all subordinate signed objects, both immediate and transitively. The effects are essentially the same as described in A-3.4.2.

A-1.6 Injection

A-1.6.1 If a CA certificate is injected the impact will depend on the data contained in the injected certificate. Changes will generally be equivalent to modification actions as described in A-1.4.

2.2. Manifest

Each repository publication point contains a manifest [RFC6486]. The RPKI incorporates manifests to enable RPs to detect suppression and/or substitution of (more recent) publication point objects, as the result of a mistake or attack. A manifest enumerates (by filename) all of the other signed objects at the publication point. The manifest also contains a hash of each enumerated file, to enable an RP to determine if the named file content matches what the INR holder identified in the manifest.

A manifest is an RPKI signed object, so it is validated as per [RFC6488]. If a manifest is modified in a way that causes any of these checks to fail, the manifest will be considered invalid. Suppression of a manifest itself (indicated by a stale manifest) also can cause an RP to not detect suppression of other signed objects at the publication point. (Note that if a Manifest's EE certificate expires at the time that the Manifest is scheduled to be replaced, a delay in publication will cause the Manifest to become invalid, not merely stale. This very serious outcome should be avoided, e.g., by making the Manifest EE certificate's notAfter value the same as that of the CA certificate under which it was issued). If a signed object at a publication point can be validated (using the rules applicable for that object type), then an RP MAY accept that object, even if there is no matching entry for it on the manifest. However, it appears that most RP software ignores publication point data that fails to match Manifest entries (at the time this document was written).

Corruption, suppression, modification, or deletion of a manifest might not affect RP processing of other publication point objects, as specified in [RFC6486]. However, as noted above, many RP implementations ignore objects that are present at a publication point but not listed in a valid Manifest. Thus the following actions against a manifest can impact RP processing:

A-2.1 Deletion

A-2.1.1 A Manifest may be deleted from the indicated publication point. In this circumstance an RP may elect to use the previous Manifest (if available), and may ignore any new/changed objects at the publication point. The implications of this action are equivalent to suppression of publication of the objects that are not recognized by RPs because the new objects are not present in

the old Manifest. For example, a new ROA could be ignored (A-1.2). A newly issued CA certificate might be ignored (A-1.1). A subordinate CA certificate that was revoked might still be viewed as valid by RPs (A-4.1). A new or changed router certificate might be ignored (A-6.2) as would a revised Ghostbusters record (A-4.1).

A-2.2 Suppression

A-2.2.1 Publication of a newer Manifest may be suppressed. Suppression of a newer Manifest probably will cause an RP to rely on a cached Manifest (if available). The older Manifest would not enumerate newly added objects, and thus those objects might be ignored by an RP, equivalent to deletion of those objects (A-1.1, A-3.1, A-4.1, A-5.1, A-6.1).

A-2.3 Corruption

A-2.3.1 A Manifest may be corrupted. A corrupted Manifest will be rejected by RPs. This may cause RPs to rely on a previous manifest, with the same impact as A-2.2. If an RP does not revert to using a cached Manifest, the impact of this action is very severe, i.e., all publication point objects probably will be viewed as invalid, including subordinate tree objects. This is equivalent to revoking or deleting an entire subtree (see A-4.4.2).

A-2.4 Modification

A-2.4.1 A Manifest may be modified to remove one or more objects. Because the modified Manifest is viewed as valid by RPs, any objects that were removed may be ignored by RPs. This is equivalent to deleting these objects from the repository. The impact of this action will vary, depending on which objects are (effectively) removed. However, the impact is equivalent to deletion of the object in question, (A-1.1, A-3.1, A-4.1, A-5.1, A-6.1).

A-2.4.2 A Manifest may be modified to add one or more objects. If an added object has a valid signature (and is non-expired), it will be accepted by RPs and processed accordingly. If the added object

was previously deleted by the INR holder, this action is equivalent to suppressing deletion of that object. If the object is newly created, or modified, it is equivalent to a modification or injection action for the type of object in question, and thus is discussed in the relevant section for those actions for the object type.

A-2.4.3 A Manifest may be modified to list an incorrect hash for one or more objects. An object with an incorrect hash may be ignored by an RP. Thus the effect may be equivalent to corrupting the object in question, although the error reported by RP software would differ from that reported for a corrupted object. (The Manifest specifications do not require an RP to ignore an object that has a valid signature and that is not revoked or expired, but for which the hash doesn't match the object. However, an RP may elect to do so.)

A-2.5 Revocation

A-2.5.1 A Manifest may be revoked (by including its EE certificate on the CRL for the publication point). A revoked Manifest will be ignored by an RP, which probably would revert to an older (cached) Manifest. The implications for RPs are equivalent to A-2.1, with regard to new/changed objects.

A-2.6 Injection

A-2.6.1 A Manifest representing different objects may be injected into a publication point. The effects are the same as for a modified Manifest (see above). The impact will depend on the type of the affected object(s), and thus is discussed in the relevant section(s) for each object type.

2.3. Certificate Revocation List

Each publication point contains a CRL that enumerates revoked (not yet expired) certificates issued by the CA associated with the publication point [RFC6481].

Adverse actions against a CRL can cause the following errors:

A-3.1 Deletion

A-3.1.1 If a CRL is deleted, RPs will continue to use an older, previously fetched Certificate Revocation List. As a result, they will not be informed of any changes in revocation status of subordinate CA or router certificates or the EE certificates of signed objects, e.g., ROAs. This action is equivalent to corruption of a CRL, since a corrupted CRL will not be accepted by an RP.

A-3.1.2 Deletion of a CRL could cause an RP to continue to accept a ROA that no longer expresses the intent of an INR holder. As a result, an announcement for the affected prefixes would be viewed as Valid, instead of NotFound or Invalid. In this case, the effect is analogous to A-5.2.

A-3.1.3 If a router certificate were revoked, and the CRL were deleted, RPs would not be aware of the revocation. They might continue to accept the old, revoked, router certificate. If the certificate had been revoked due to a compromise of the router's private key, RPs would be vulnerable to accepting routes signed by an unauthorized entity.

A-3.1.4 If a subordinate CA certificate were revoked on the deleted CRL, the revocation would not take effect. This could interfere with a transfer of address space from the subordinate CA, adversely affecting routing to the new holder of the space.

A-3.2 Suppression

A-3.2.1 If publication of the most recent CRL is suppressed, an RP will not be informed of the most recent revocation status of subordinate CA or router certificates or the EE certificates of signed objects. If an EE certificate has been revoked and the associated signed object is still present in the publication point, an RP might mistakenly treat that object as valid. (This would happen if the object is still in the manifest or the RP is configured to process valid objects that are not on the manifest.) This type of action is of special concern if the affected object is a ROA, a router certificate, or a

subordinate CA certificate. The effects here are equivalent to CRL deletion (A-3.1), but suppression of a new CRL may not even be reported as an error, i.e., if the suppressed CRL were issued before the NextUpdate time (of the previous CRL).

A-3.3 Corruption

A-3.3.1 If a CRL is corrupted, an RP will reject it. If a prior CRL has not yet exceeded its NextUpdate time, an RP will continue to use the prior CRL. Even if the prior CRL has passed the NextUpdate time, an RP may choose to continue to rely on the prior CRL. The effects are essentially equivalent to suppression or deletion of a CRL (A-3.1, A-3.2).

A-3.4 Modification

A-3.4.1 If a CRL is modified to erroneously list a signed object's EE certificate as revoked, the corresponding object will be treated as invalid by RPs, even if it is present in a publication point. If this object is a ROA, the (legitimate) binding expressed by the ROA will be ignored by an RP (see A-5.5). If a CRL is modified to erroneously list a router certificate as revoked, a path signature associated with that certificate will be treated as Not Valid by RPs (see A-6.5).

A-3.4.2 If a CRL is modified to erroneously list a CA certificate as revoked, that CA and all subordinate signed objects will be treated as invalid by RPs. Depending on the location of the affected CA in the hierarchy, these effects could be very substantial, causing routes that should be Valid to be treated as NotFound.

A-3.4.3 If a CRL is modified to omit a revoked EE, router, or CA certificate, RPs likely will continue to accept the revoked, signed object as valid. This contravenes the intent of the INR holder. If an RP continues to accept a revoked ROA, it may make routing decisions on now-invalid data. This could cause valid routes to be de-preferenced and invalid routes to continue to be accepted.

A-3.5 Revocation

A-3.5.1 A CRL cannot be revoked, per se, but it will fail validation if the CA certificate under which it was issued is revoked. See A-1.5 for a discussion of that action.

A-3.6 Injection

A-3.6.1 Insertion of a bogus CRL can have the same effects as listed above for a modified CRL, depending on how the inserted CRL differs from the correct CRL.

2.4. ROA

In addition to the generic RPKI object syntax checks, ROA validation requires that the signature on the ROA can be validated using the public key from the EE certificate embedded in the ROA [RFC6482]. It also requires that the EE certificate be validated consistently with the procedures described in [RFC6482] and [RFC6487]. Adverse actions against a ROA can cause the following errors:

A-4.1 Deletion

A-4.1.1 A ROA may be deleted from the indicated publication point. The result is to void the binding between the prefix(es) and the AS number in the ROA. An RP that previously viewed this binding as authentic will now not have any evidence about its validity. For origin validation, this means that a legitimate route will be treated as NotFound (if there are no other ROAs for the same prefix) or Invalid (if there is another ROA for the same prefix, but with a different AS number).

A-4.2 Suppression

A-4.2.1 Publication of a newer ROA may be suppressed. If the INR holder intended to change the binding between the prefix(es) and the AS number in the ROA, this change will not be effected. As a result, RPs may continue to believe an old prefix/ ASN binding that is no longer what the INR holder intended.

A-4.2.2 If an INR holder intends to issue and publish two (or more) new ROAs for the same address space, one (or more) of the new ROAs may be suppressed while the other is published. In this case, RPs will de-preference the suppressed prefix/ASN binding. Suppression of the new ROA might cause traffic to flow to an ASN other than the one(s) intended by the INR holder.

A-4.2.3 If an INR holder intends to delete all ROAs for the same address space, some of them may be retained while the others are deleted. Preventing the deletion of some ROAs can cause traffic to continue to be delivered to the ASNs that were advertised by these ROAs. Deletion of all ROAs is consistent with a transfer of address space to a different INR holder, in a phased fashion. Thus this sort of attack could interfere with the successful transfer of the affected address space (until such time as the prefixes are removed from the previous INR holder's CA certificate).

A-4.3 Corruption

A-4.3.1 A ROA may be corrupted. A corrupted ROA will be ignored by an RP, so the effect is essentially the same as for A-4.1 and 4.5. A possible difference is that an RP may be alerted to the fact that the ROA was corrupted, which might attract attention to the attack.

A-4.4 Modification

A-4.4.1 A ROA may be modified so that the Autonomous System Number (ASN) or one or more of the address blocks in a ROA is different from the values the INR holder intended for this ROA. (This action assumes that the modified ROA's ASN and address ranges are authorized for use by the INR holder.) This attack will cause RPs to de-preference the legitimate prefix/ASN binding intended by the INR holder.

A-4.5 Revocation

A-4.5.1 A ROA may be revoked (by placing its EE certificate on the CRL for the publication point). This has the same effect as A-4.1.

A-4.6 Injection

A-4.6.1 A ROA expressing different bindings than those published by the INR holder may be injected into a publication point. This action could authorize an additional ASN to advertise the specified prefix, allowing that ASN to originate routes for the prefix, thus enabling route origin spoofing. In this case, the injected ROA is considered to be in competition with any existing authorized ROAs for the specified prefix.

A-4.6.2 An injected ROA might express a different prefix for an ASN already authorized to originate a route, e.g., a longer prefix, which could enable that ASN to override other advertisements using shorter prefixes. If there are other ROAs that authorize different ASNs to advertise routes to the injected ROA's prefix, then the injected ROA is in competition with these ROAs.

2.5. Ghostbusters Record

The Ghostbusters record [RFC6493] is a signed object that MAY be included at a publication point, at the discretion of the INR holder or publication point operator. The record is validated according to [RFC6488]. Additionally, the syntax of the record is verified based on the vCard profile from Section 5 of [RFC6493]. Errors in this record do not affect RP processing. However, if an RP encounters a problem with objects at a publication point, the RP may use information from the record to contact the publication point operator.

Adverse actions against a Ghostbusters record can cause the following error:

A-5.1 Deletion, suppression, corruption, or revocation of a Ghostbusters record could prevent an RP from contacting the appropriate entity when a problem is detected by the RP. Modification or injection of a Ghostbusters record could cause an RP to contact the wrong entity, thus delaying remediation of a detected anomaly. All of these actions are viewed as equivalent from an RP processing perspective; they do not alter RP validation of ROAs or router certificates. However, these actions can interfere with remediation of a problem when detected by an RP.

2.6. Router Certificates

Router certificates are used by RPs to verify signatures on BGPsec_Path attributes carried in Update messages.

Each AS is free to determine the granularity at which router certificates are managed [I-D.ietf-sidr-bgpsec-pki-profiles]. Each participating AS is represented by one or more router certificates. During key or algorithm rollover, multiple router certificates will be present in a publication point, even if the AS is normally represented by just one such certificate.

Adverse actions against router certificates can cause the following errors:

A-6.1 Deletion

A-6.1.1 Deletion of a router certificate would cause an RP to not be able to verify signatures applied to BGPsec_Path attributes on behalf of the AS in question. In turn, this would cause the route to be treated with lower preference than competing routes that have valid BGPsec_Path attribute signatures. (However, if another router certificate for the affected AS is valid and contains the same AS number and public key, and is in use by that AS, there would be no effect on routing. This scenario will arise if a router certificate is renewed, i.e., issued with a new validity interval.)

A-6.2 Suppression

A-6.2.1 Suppression of a router certificate could have the same impact as deletion of a certificate of this type, i.e., if no router certificate was available, BGPsec attributes that should be verified using the certificate would fail validation. If an older certificate existed, and had not expired, it would be used by RPs. If the older certificate contained a different ASN, the impact would be the same as in A-6.4.

A-6.3 Corruption

A-6.3.1 Corruption of a router certificate will result in the certificate being rejected by RPs. Absent a valid router certificate, BGPsec_Path attributes associated with that certificate will be unverifiable. In turn, this would cause the route to be treated with lower preference than competing routes that have valid BGPsec_Path attribute signatures.

A-6.4 Modification

A-6.4.1 If a router certificate is modified to represent a different ASN, but it still passes syntax checks, then this action could cause signatures on BGPsec_Path attributes to be associated with the wrong AS. This could cause signed routes to be inconsistent with the intent of the INR holder, e.g., traffic might be routed via a different AS than intended.

A-6.5 Revocation

A-6.5.1 If a router certificate were revoked, BGPsec_Path attributes verifiable using that certificate would not longer be considered valid. The impact would be the same as for a deleted certificate, as described in A-6.1.

A-6.6 Injection

A-6.6.1 Insertion of a router certificate could authorize additional routers to sign BGPsec traffic for the targeted ASN, and thus undermine fundamental BGPsec security guarantees. If there are existing, authorized router certificates for the same ASN, then the injected router certificate is in competition with these existing certificates.

3. Analysis of Actions Relative to Scenarios

This section examines the types of problems that can arise in four scenarios described below. We consider mistakes, (successful) attacks against a CA or a publication point, and situations in which a CA or publication point manager is compelled to take action by a law enforcement authority.

We explore the following four scenarios:

A. An INR holder operates its own CA and manages its own repository publication point.

B. An INR holder operates its own CA, but outsources management of its repository publication point to its parent or another entity.

C. An INR holder outsources management of its CA to its parent, but manages its own repository publication point.

D. An INR holder outsources management of its CA and its publication point to its parent.

Note that these scenarios focus on the affected INR holder as the party directly affected by an adverse action. The most serious cases arise when the INR holder appears as a high-tier CA in the RPKI hierarchy; in such situations subordinate INR holders may be affected as a result of an action. A mistake by or an attack against a "leaf" has more limited impact because all of the affected INRs belong to the INR holder itself.

In Scenario A, actions by the INR holder can adversely affect all of its resources and, transitively, resources of any subordinate CAs. (If the CA is a "leaf" in the RPKI, then it has no subordinate CAs and the damage is limited to its own INRs.)

In Scenario B, actions by the (outsourced) repository operator also can adversely affect the resources of the INR holder, and those of any subordinates CAs. (If the CA is a "leaf" in the RPKI, then it has no subordinate CAs and the damage is limited, as in Scenario A.) The range of adverse effects here includes those in Scenario A, and adds a new potential source of adverse actions, i.e., the outsourced repository operator.

In Scenario C, all signed objects associated with the INR holder are generated by the parent CA but are self-hosted. (We expect this scenario to be rare, because an INR holder that elects to outsource CA operation seems unlikely to manage its own repository publication point.) Because that CA has the private key used to sign them, it can generate alternative signed objects---ones not authorized by the INR holder. However, erroneous objects created by the parent CA will not be published by the INR holder IF the holder checks them first. Because the parent CA is acting on behalf of the INR holder, mistakes by or attacks against that entity are equivalent to ones effected by the INR holder in Scenario A.

The INR holder is most vulnerable in Scenario D. Actions by the parent CA, acting on behalf of the INR holder, can adversely affect

all signed objects associated with that INR holder, including any subordinate CA certificates. These actions will presumably translate directly into publication point changes, because the parent CA is managing the publication point for the INR holder. The range of adverse effects here includes those in Scenarios A, B, and C.

3.1. Scenario A

In this scenario, the INR holder acts as its own CA and it manages its own publication point. Actions by the INR holder can adversely affect all of its resources and, transitively, resources of any subordinate CAs. (If the CA is a "leaf" in the RPKI, then it has no subordinate CAs and the damage is limited to its own INRs.) Mistakes by the INR holder can cause any of the actions noted in Section 2. A successful attack against this CA can effect all of the modification, revocation, or injection actions noted in that section. (We assume that objects generated by the CA are automatically published). An attack against the publication point can effect all of the deletion, suppression, or corruption actions noted in that section.

3.2. Scenario B

In this scenario, the INR holder acts as its own CA and but it delegates management of it own publication point to a third party. Mistakes by the INR holder can cause any of the modification, revocation, or injection actions described in Section 2. Actions by the repository operator can adversely affect the resources of the INR holder, and those of any subordinate CAs. (If the CA is a "leaf" in the RPKI, then it has no subordinate CAs and the damage is limited, as in Scenario A.) The range of adverse effects here includes those in Scenario A, and adds a new potential source of adverse actions, i.e., the third party repository operator. A successful attack against the CA can effect all of the modification, revocation, or injection actions noted in that section (assuming that objects generated by the CA are automatically published). Here, actions by the publication point manager (or attacks against that entity) can effect all of the deletion, suppression, or corruption actions noted in Section 2.

3.3. Scenario C

In this scenario, the INR holder outsources management of its CA to its parent, but manages its own repository publication point. All signed objects associated with the INR holder are generated by the parent CA but are self-hosted. (We expect this scenario to be rare, because an INR holder that elects to outsource CA operation seems unlikely to manage its own repository publication point.) Because that CA has the private key used to sign them, it can generate

alternative signed objects -- ones not authorized by the INR holder. However, erroneous objects created by the parent CA will not be published by the INR holder IF the holder checks them first. Because the parent CA is acting on behalf of the INR holder, mistakes by or attacks against that entity are equivalent to ones effected by the INR holder in Scenario A. Mistakes by the INR holder, acted upon by the parent CA, can cause any of the actions noted in Section 2. Actions unilaterally undertaken by the parent CA also can have the same effect, unless the INR holder checks the signed objects before publishing them. A successful attack against the parent CA can effect all of the modification, revocation, or injection actions noted in Section 2, unless the INR holder checks the signed objects before publishing them. An attack against the INR holder (in its role as repository operator) can effect all of the deletion, suppression, or corruption actions noted in Section 2 (because the INR holder is managing its publication point), unless the INR holder checks the signed objects before publishing them. (An attack against the INR holder implies that the path it uses to direct the parent CA to issue and publish objects has been compromised.)

3.4. Scenario D

In this scenario an INR holder outsources management of both its CA and its publication point to its parent. The INR holder is most vulnerable in this scenario. Actions by the parent CA, acting on behalf of the INR holder, can adversely affect all signed objects associated with that INR holder, including any subordinate CA certificates. These actions will presumably translate directly into publication point changes, because the parent CA is managing the publication point for the INR holder. The range of adverse effects here includes those in Scenarios A, B, and C. Mistakes by the INR holder, acted upon by the parent CA, can cause any of the actions noted in Section 2. Actions unilaterally undertaken by the parent CA also can have the same effect. A successful attack against the parent CA can effect all of the modification, revocation, or injection actions noted in Section 2. An attack against the parent CA can also effect all of the deletion, suppression, or corruption actions noted in Section 2 (because the parent CA is managing the INR holder's publication point).

4. Security considerations

This informational document describes a threat model for the RPKI, focusing on mistakes by or attacks against CAs and independent repository managers. It is intended to provide a basis for the design of future RPKI security mechanisms that seek to address the concerns associated with such actions.

The analysis in this document identifies a number of circumstances in which attacks or errors can have significant impacts on routing. One ought not interpret this as a condemnation of the RPKI. It is only an attempt to document the implications of a wide range of attacks and errors, in the context of the RPKI. The primary alternative mechanism for disseminating routing information is Internet Routing Registry (IRR) technology ([RFC2650], [RFC2725]), which uses the Routing Policy Specification Language (RPSL) [RFC2622]. IRR technology exhibits its own set of security problems, which are discussed in [RFC7682].

5. IANA Considerations

This document has no actions for IANA.

6. Acknowledgements

The authors thank Richard Hansen and David Mandelberg for their extensive review, feedback and editorial assistance.

7. References

7.1. Normative References

[I-D.ietf-sidr-bgpsec-overview]
Lepinski, M. and S. Turner, "An Overview of BGPsec", draft-ietf-sidr-bgpsec-overview-08 (work in progress), June 2016.

[I-D.ietf-sidr-bgpsec-pki-profiles]
Reynolds, M., Turner, S., and S. Kent, "A Profile for BGPsec Router Certificates, Certificate Revocation Lists, and Certification Requests", draft-ietf-sidr-bgpsec-pki-profiles-18 (work in progress), July 2016.

[I-D.ietf-sidr-bgpsec-protocol]
Lepinski, M. and K. Sriram, "BGPsec Protocol Specification", draft-ietf-sidr-bgpsec-protocol-17 (work in progress), June 2016.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC3779] Lynn, C., Kent, S., and K. Seo, "X.509 Extensions for IP Addresses and AS Identifiers", RFC 3779, DOI 10.17487/RFC3779, June 2004, <http://www.rfc-editor.org/info/rfc3779>.

[RFC5652] Housley, R., "Cryptographic Message Syntax (CMS)", STD 70, RFC 5652, DOI 10.17487/RFC5652, September 2009, <http://www.rfc-editor.org/info/rfc5652>.

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

[RFC6481] Huston, G., Loomans, R., and G. Michaelson, "A Profile for Resource Certificate Repository Structure", RFC 6481, DOI 10.17487/RFC6481, February 2012, <http://www.rfc-editor.org/info/rfc6481>.

[RFC6482] Lepinski, M., Kent, S., and D. Kong, "A Profile for Route Origin Authorizations (ROAs)", RFC 6482, DOI 10.17487/RFC6482, February 2012, <http://www.rfc-editor.org/info/rfc6482>.

[RFC6483] Huston, G. and G. Michaelson, "Validation of Route Origination Using the Resource Certificate Public Key Infrastructure (PKI) and Route Origin Authorizations (ROAs)", RFC 6483, DOI 10.17487/RFC6483, February 2012, <http://www.rfc-editor.org/info/rfc6483>.

[RFC6485] Huston, G., "The Profile for Algorithms and Key Sizes for Use in the Resource Public Key Infrastructure (RPKI)", RFC 6485, DOI 10.17487/RFC6485, February 2012, <http://www.rfc-editor.org/info/rfc6485>.

[RFC6486] Austein, R., Huston, G., Kent, S., and M. Lepinski, "Manifests for the Resource Public Key Infrastructure (RPKI)", RFC 6486, DOI 10.17487/RFC6486, February 2012, <http://www.rfc-editor.org/info/rfc6486>.

[RFC6487] Huston, G., Michaelson, G., and R. Loomans, "A Profile for X.509 PKIX Resource Certificates", RFC 6487, DOI 10.17487/RFC6487, February 2012, <http://www.rfc-editor.org/info/rfc6487>.

[RFC6488] Lepinski, M., Chi, A., and S. Kent, "Signed Object Template for the Resource Public Key Infrastructure (RPKI)", RFC 6488, DOI 10.17487/RFC6488, February 2012, <http://www.rfc-editor.org/info/rfc6488>.

[RFC6489] Huston, G., Michaelson, G., and S. Kent, "Certification Authority (CA) Key Rollover in the Resource Public Key Infrastructure (RPKI)", BCP 174, RFC 6489, DOI 10.17487/RFC6489, February 2012, <http://www.rfc-editor.org/info/rfc6489>.

[RFC6493] Bush, R., "The Resource Public Key Infrastructure (RPKI) Ghostbusters Record", RFC 6493, DOI 10.17487/RFC6493, February 2012, <http://www.rfc-editor.org/info/rfc6493>.

[RFC6916] Gagliano, R., Kent, S., and S. Turner, "Algorithm Agility Procedure for the Resource Public Key Infrastructure (RPKI)", BCP 182, RFC 6916, DOI 10.17487/RFC6916, April 2013, <http://www.rfc-editor.org/info/rfc6916>.

[RFC7132] Kent, S. and A. Chi, "Threat Model for BGP Path Security", RFC 7132, DOI 10.17487/RFC7132, February 2014, <http://www.rfc-editor.org/info/rfc7132>.

7.2. Informative References

[I-D.ymbk-sidr-transfer]
Austein, R., Bush, R., Huston, G., and G. Michaelson, "Resource Transfer in the Resource Public Key Infrastructure", draft-ymbk-sidr-transfer-01 (work in progress), July 2015.

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

[RFC2650] Meyer, D., Schmitz, J., Orange, C., Prior, M., and C. Alaettinoglu, "Using RPSL in Practice", RFC 2650, DOI 10.17487/RFC2650, August 1999, <http://www.rfc-editor.org/info/rfc2650>.

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

[RFC7682] McPherson, D., Amante, S., Osterweil, E., Blunk, L., and D. Mitchell, "Considerations for Internet Routing Registries (IRRs) and Routing Policy Configuration", RFC 7682, DOI 10.17487/RFC7682, December 2015, <http://www.rfc-editor.org/info/rfc7682>.

Authors' Addresses

Stephen Kent
BBN Technologies
10 Moulton St
Cambridge, MA 02138-1119
USA

Email: kent@bbn.com

Di Ma
ZDNS
4 South 4th St. Zhongguancun
Haidian, Beijing 100190
China

Email: madi@zdns.cn