RIPE Database Software

Recent Changes

Shane Kerr, RIPE NCC
shane@ripe.net

X.509 Support Added

As part of the Improved Secure Communication System for RIPE NCC Members, X.509 support was added to the RIPE Database.

More information about X.509 authentication can be found here:

http://www.ripe.net/db/x509.html

Database Schema Changes

The KEY-CERT class was changed to allow users to represent their X.509 certificates, like this:

```
key-cert:   X509-42
method:     X509
owner:      /C=NL/O=RIPE NCC/OU=Members/CN=eu.ripe-ncc.shane/Email=shane@ripe.net
fingerpr:   D5:92:29:08:F8:AB:75:5F:42:F5:A8:5F:A3:8D:08:2E
certif:     -----BEGIN CERTIFICATE-----
certif:     MIID/DCCA2WgAwIBAgICAIQwDQYJKoZIhvcNAQEEBQAwcTELMAkGA1UEBhMCRVUx
certif:     EDAOBgNVBAgTB0hvbGxhbmQxEDAOBgNVBAoTB25jY0RFTU8xHTAbBgNVBAMTFFNv
  .
  .
  .
certif:     hZNmF5c/d0gauqvL+egb+3V+Zg+sJTzHMVKQLF1ybWgJjU75Pi+mO7BG0zsQ13pT
certif:     YxuZCR2W15nwt7zLiHtmfw==
certif:     -----END CERTIFICATE-----
remarks:    Sample Key Certificate
notify:     ripe-dbm@ripe.net
mnt-by:     RIPE-DBM-MNT
changed:    ripe-dbm@ripe.net 20040101
source:     RIPE
```

The “auth:” attribute was updated so that the KEY-CERT objects that contain X.509 certificates can be referenced.

Interface Changes

E-mail support has been extended to allow S/MIME messages, which is the standard for using X.509 authentication in e-mail.

Client-side certificates are recognised by both the webupdates and syncupdates interfaces when SSL is used.

Organisation Object Added

A new type of object, the ORGANISATION object, is now available in the RIPE Database. Full information can be found at:

http://www.ripe.net/db/organisation.html

The RIPE Whois Database stores three main types of contact information: PERSON, ROLE, and ORGANISATION objects. The PERSON and ROLE objects provide a way to find people responsible for operations or usage of the resources represented in the RIPE Whois Database (IP blocks, Autonomous Systems, and domain names). However, these do not provide an easy way of mapping resources to a particular organisation. The ORGANISATION object fulfils this need.

Object Details

A sample organisation object:

```
organisation: ORG-RBI1-RIPE
org-name:     Ruritania Banking Interchange
org-type:     NON-REGISTRY
address:      1 High Street
address:      Polarcity
address:      Northern Nowhere
phone:        +31 20 5354444
e-mail:       bit-bucket@ripe.net
admin-c:      HOHO15-RIPE
tech-c:       HOHO15-RIPE
ref-nfy:      bit-bucket@ripe.net
mnt-ref:      RURITANIA-MNT
mnt-by:       RIPE-NCC-HM-MNT
changed:      ripe-dbm@ripe.net 20040419
source:       RIPE
```

All object types may refer to an object, by adding the "org:" attribute. The maintainer specified in the "mnt-ref:" of the organisation object must authorise the reference (`RURITANIA-MNT` in the above object). This is to prevent people from referencing organisation objects that they have no relationship with.

Queries

The organisations can be looked up by handle, such as `ORG-RBI1-RIPE`, or by name, such as "Ruritania Banking Interchange". Also, you can look up all of the objects that reference a given organisation via an inverse query. For example:

```
whois -r -i org ORG-SANT1-RIPE
```

Will return all objects that have `ORG-SANT1-RIPE` in their "org:" attributes.

By default, any organisation objects referenced by another object will be returned with that object, the same as person and role objects are.

IANA, RIR, and LIR Objects

The IANA and RIRs have organisation objects in the RIPE Whois Database that are maintained by the RIPE NCC. These are used to mark appropriate resources, such as INETNUM objects for /8 allocations from the IANA to the RIPE NCC.

LIRs have organisation objects that are created and maintained for them by the RIPE NCC. The ORGANISATION object for an LIR is created in the RIPE Whois Database when the organisation becomes an LIR.

INETNUM, INET6NUM, and AS-BLOCK objects were updated to include references to the appropriate organisation.

LIRs can update parts of their organisation objects through the LIR Portal:

https://lirportal.ripe.net/

Reverse DNS Changes

Some changes have been made to the way that reverse DNS domains (`in-addr.arpa` and `ip6.arpa`) are handled, including the database support for reverse domain objects. The best place for current reverse DNS information is:

http://www.ripe.net/reverse/

Old Way

The reverse data was kept in the RIPE Database as DOMAIN objects, and also in DNS zone files, maintained separately. When users wanted to update their reverse DNS information, they would send an e-mail to <auto-inaddr@ripe.net>. A program would then verify the request was valid, and then update both the RIPE Database and the zone files.

There were several problems with this set-up:

- LIRs had to use a separate interface for the maintenance of DNS.
- LIRs would update the RIPE Database directly, causing inconsistencies between the view from Whois and DNS.
- The database update software developed more rapidly than the DNS update, making features like web updates and X.509 authentication unavailable to DNS administrators.
- Full automation was impractical, requiring human intervention and the related delays.
- The policy required significant additional work from LIRs.

New Way

The reverse data is kept in the RIPE Database as DOMAIN objects. DNS zone files are periodically built from the contents of the RIPE Database.

When users want to update their reverse DNS information, they send an e-mail to <auto-dbm@ripe.net>. The database update program will verify the DNS information and update the RIPE Database. The information will appear in the DNS after a short delay.

The policy constraints have been reduced. Previously only space assigned to end users could be reverse delegated in DNS. This caused administrative burden to LIRs, as every time space was assigned, reverse DNS had to be set up for that space. Now reverse DNS can be set up for an entire allocation.

There are other advantages, in addition to solving the problems above:

- The introduction of the "mnt-domains:" attribute allows the DNS to be administered by a different set of people to those that maintain INETNUM or INET6NUM objects.
- The rules for DOMAIN object names have been made more strict, preventing accidents (for example, `666.193.in-addr.arpa`).
- Deployment of DNS Security Extensions (DNSSEC) requires a method for the exchange of public keys. Using the Whois Database as the authoritative source for zone file creation enables the use of the Whois Database authorisation mechanisms including the LIR Portal, PGP keys and X.509 certificates, for DNSSSEC public key exchanges.

NONE Authentication Scheme Deprecated

The RIPE Database will no longer accept updates using the NONE authentication scheme.

NONE was intended to be used consciously, as a notification facility or as a means to tag objects. This authentication scheme was deprecated because it is likely that in many cases NONE is used simply because it is easy.

An announcement with full details was sent to the Database Working Group, but not to a larger audience, because of the potential security concerns. Full details about the process, as well as instructions for users affected by the change, can be found here:

http://www.ripe.net/db/none-deprecation-042004.html

Maintainer Modification

The procedure used by the RIPE NCC to update maintainers was straightforward. Any “auth: NONE” attributes were removed. If that was the only authentication scheme, then a password was generated, and an MD5-PW “auth:” attribute was added. Any such password generated was e-mailed to the contact(s) of the maintainers.

RIPE-NCC-NONE-MNT

A maintainer with NONE authentication, `RIPE-NCC-NONE-MNT`, was added to objects without any maintainer when the database was converted from RIPE-181 format to RPSL format in April 2001. The main use of this maintainer was for INETNUM objects. There were approximately 60000 such objects, making it impractical to create new maintainers for all of them.

Instead, a special maintainer was used to replace these, `RIPE-NCC-LOCKED-MNT`, which has an authentication only available to the RIPE NCC. A URL was sent to the contacts on the INETNUM or other objects referencing this maintainer, which will let them generate a new maintainer or assign another existing maintainer to the object.

Routing Policy

Another use of the `RIPE-NCC-NONE-MNT` has been to allow the creation of objects representing routing policy for resources not allocated or assigned by the RIPE NCC. This was done by using "mnt-routes: RIPE-NCC-NONE-MNT" or "mnt-lower: RIPE-NCC-NONE-MNT" as appropriate. A new maintainer object, `RIPE-NCC-RPSL-MNT`, was created for these cases, with a well-known password, published in the object.

Other Database Changes

There were a number of relatively minor changes to the database that were also made. Details may be found in various announcements:

http://www.ripe.net/ripe/mail-archives/db-wg/2004/msg00282.html
http://www.ripe.net/ripe/mail-archives/db-wg/2003/msg00035.html

CIDR Notation for INETNUM Creation Supported

We now allow the use of CIDR notation when creating INETNUM objects. The CIDR string is replaced by the expanded range notation before the object is processed. This feature is *only* permitted when creating an object. For example, if the following object is submitted to the RIPE Database:

```
inetnum: 1.2.0.0/16
```

Before any further processing is done it will be converted to:

```
inetnum: 1.2.0.0 - 1.2.255.255
```

Prefix Range Lists for "mnt-routes:" Implemented

The "mnt-routes:" attribute syntax has been extended to allow prefix range lists. This will enable people to specify which maintainer has to authorise the creation of the specific routes. For example:

```
mnt-routes: MY-MNT { 20.34.0.0/1617-18, 20.34.0.0/16^- }
mnt-routes: NOT-MY-MNT { ANY }
```

Support for the previous syntax is unchanged. The extended syntax complies with RFC2725 which can be found at:

ftp://ftp.ripe.net/rfc/rfc2725.txt

Creation of Overlapping INETNUM Objects Prevented

It used to be possible to create INETNUM objects with overlapping begin and end IP addresses. For example:

```
inetnum:  10.0.2.0 - 10.0.3.255
inetnum:  10.0.3.0 - 10.0.4.255
```

The database will no longer allow this sort of behaviour. Nested INETNUM objects may still be created.