Abuse Handling design for the current RIPE Database model

Abuse Handling design for the current RIPE Database model

Introduction

Currently the RIPE Database has a combination of several implementations for handling abuse contact details. With a mixture of methods, data maintainers do not understand how to document abuse handlers. The public and many tool writers do not understand how to find abuse handlers. Most abuse details are still written into remarks. For automated tools it is very hard to parse remarks. They can say anything, in any language and be included in any object.

For a tool to always find the abuse contact for any resource we need some re-structuring of the data. With one simple, effective and well documented method for recording abuse contact details and a well defined process for finding them, the RIPE NCC can provide an interface to always return this data. Third party tool writers can program against this interface and know they will always get the right address(es). If anything changes in the underlying logic, this can be reflected in the interface and all the tools will continue to work correctly.

The current methods were all additional functionality built on top of each other, but without re-using, what already existed within the database. When the code is adapted and re-adapted several times the quality gets worse - sometimes it is better to take a step back and re-design a solution. One approach now is to go back to the database basics and implement a full abuse handling process that fully integrates into the current RIPE Database design and user experience. This article aims to propose one such new design.

Proposal to re-structure the data

An abuse handler is a contact related to an Internet resource who has a defined responsibility or role. In terms of the current RIPE Database structure, this is the same as an administrative contact, technical contact or DNS zone contact. So lets start with an "abuse-c:" attribute referencing an abuse contact. This fits the existing model of "admin-c:", "tech-c:" and "zone-c:" attributes that is well known and understood. To make this new abuse contact reference work effectively as an abuse handler it needs three things defining:

- what it points at
- where to put it
- how to find it

What it points at

An abuse handler has a defined role. So it makes sense to use the existing **role** object to hold the contact details. No need for new or complex data objects. To make abuse well defined a number of additional features can be built in. Introduce a new NIC hdl suffix '-abuse'. Only allow "abuse-c:" to reference a '-abuse' NIC hdl and not allow any other contact to reference these.

The database software already supports two ways to enforce an attribute. Syntax makes attributes mandatory or optional. Business rules can make optional attributes required or not in some situations. We can adjust the syntax to make both "e-mail:" and "abuse-mailbox:" optional attributes in a **role** object. Business rules can then make "abuse-mailbox:" required in any **role** object with a '-abuse' NIC hdl and not allowed in any other **role** object. Business rules can also make "e-mail:" required in any **role** without a '-abuse' NIC hdl (maintaining current syntax behaviour).

Where to put it

To keep it simple we want to be able to put it in the least number of places with the most flexible options to give maximum coverage. All LIRs and new independent resource holders 1 already have an organisation defined. All their Internet resources allocated by the RIPE NCC are linked to their organisation. So the simplest location is to add a (mandatory) reference to their abuse role to their organisation details. This one, single reference covers all the Internet resources managed by this organisation, including all their more specific customer's networks. Any abuse

complaints for any network derived from any of the LIR's resources can be directed to this default abuse handler for the LIR's organisation. (It does not handle downstream customers by default as this type of
hierarchy is not currently defined in the RIPE Database.)

Of course one size never fits all. By adding optional references to the abuse role in address space and Autonomous System Number resources, there is the flexibility to fine tune the abuse handling to any level. This can be applied in a hierarchical way so again the minimum amount of data needs to hold these references for maximum effect.

1 Defining organisations for independent resource holders is a recent policy. There are still many independent resources that do not yet have an organisation defined. This is being addressed as part of the 2007-01 policy implementation.

To show this simply as a table:

role type	NIC suffix	e-mail	abuse-mailbox	referenced by	referenced from (mandatory)	referenced from (optional)
abuse	only -abuse	optional	required	abuse-c	organisation	inet(6)num, aut-num
other	anything except -abuse	required	not allowed	admin-c, tech-c, zone-c	none	any object with contacts

How to find it

Finding the abuse contact for a resource with this structure is also simple. The basic philosophy is to start with an Internet resource. Then ask questions of the database:

- does it have a direct reference to an abuse role? If so that is the abuse handler.
- does it have a direct reference to an organisation? If so the organisation will reference a default abuse handler.
- if neither, search up the hierarchy looking for an object that has one of these references. When one is found that is the abuse handler. If both are found, the direct abuse role reference takes precedence.
- If none are found, then this resource does not have a dedicated abuse handler defined.

These searches will be a combination of query logic based on syntax and additional business rule logic. Any other searches may be easily applied. For example, at some point a rule may be to search for the originating AS Number of the routes for address space and check for an "abuse-c:" reference in the **aut-num** object. Also consider that not all network hierarchies are perfectly formed and don't always follow the allocation/assignment hierarchy (think ERX and possible future transfers of bits of ranges). By keeping the database structure simple and defining easy rules for where to put the references, we can build any complexity and exceptions into the search logic hidden behind the RIPE NCC's Abuse Finder interface. This will provide more accurate information than directly querying the RIPE Database yourself and possibly making the wrong assumptions. This service can offer a web interface for the public to use directly. It can also provide a programmable interface using HTTP get for third party tool writers.

What to do if no dedicated abuse handler found

There will need to be a transition time for users to re-structure existing data. During this period the RIPE NCC's Abuse Finder can apply the logic of both old and new structures. If contacts are found for both, the new structure may be given a higher priority. After a defined transition period all other "abuse-mailbox:" attributes should be converted into "remarks:" and the syntax can be adjusted for those objects.

Options could be made available to drive the changeover or introduce abuse contacts where previously there were none. The update software can use the abuse finder logic to check if a resource has abuse contacts when the resource is created or modified. Warnings could be added to the update acknowledgment if no abuse contact is found. After a further transition period these warnings could be changed to errors. Then abuse contacts will have to be set up before a resource can be created or modified.

It should be the responsibility of the registrant of the resource to ensure that dedicated abuse contact details are properly registered in the RIPE Database. It may be a long time before all data is covered. In the mean time should the abuse finder tool provide other email addresses, like "notify:" or "upd-to:", which were not intended for handling abuse or simply state no abuse contact is available for that resource?

How to set up abuse contacts

The above sections define the logic of how to define the abuse contact details and how to find it. The user needs to know how to set it up.

Start by creating a **role** object with a nic-hdl ending in '-abuse'. Set the "abuse-mailbox:" attribute in this **role** object to the email address used for receiving abuse reports.

If existing users have an **organisation** object, add an "abuse-c:" attribute to the **organisation** object referencing the new **role** object. This becomes the default abuse contact for this organisation. If this addition is mandatory, next time the **organisation** object is modified it will have to be added. The default abuse handling for your organisation is now set up for all your Internet resources, including all your more specific customers. Without doing anything else you are covered.

You may want to use different abuse handlers for different Internet resources or parts of your network. You can optionally add references to different abuse contact roles to any of your resources or any part of your networks. If one of your assigned customers is willing to handle his own

abuse complaints for his network, you can add a reference to his organisation into the resources you assign to him.

Adding these additional references has the effect of delegating the abuse handling for this part of a network and any more specifics. There can be multiple levels of these references. So an LIR with an address space allocation may make a sub-allocation to a customer, who then makes an assignment to an end user. All of this address space is covered by the LIR's organisation default abuse contact. But additional references can delegate responsibility to the sub-allocation customer and the end user for their respective parts of the network.

Now optionally add "abuse-c:" attributes at any point in the network of **inet(6)num** objects to delegate the abuse handling for this part of a network and more specifics down to the next **inet(6)num** object that contains an "abuse-c:" or "org:" reference.

If the **organisation** is referenced at any point in the network of **inet(6)num** objects, they are already covered by this as a default from the point in the network where this **organisation** object is referenced, down through all the more specific objects. They do not need to add an "abuse-c:" attribute to any of the **inet(6)num** objects in this part of their network.

If the user has any AS numbers they can add "abuse-c:" attributes to the **aut-num** objects directly, or reference an **organisation** object. For recently
assigned AS numbers, reference to an organisation was required. So it is already covered.

All address space and AS numbers **must** be covered by abuse handlers. The user can choose if they do it by referencing their **organisation** object or putting the "abuse-c:" attributes into their **inet(6)num** and **aut-num** objects.

Users should **not** use "remarks:" attributes to define abuse handling.