A perspective on the role of the IETF's CRISP working group

Leslie Daigle & Andrew Newton

June 2003

1 Introduction

This document describes the work that is being done in the IETF's CRISP ("Cross Registry Information Service Protocol) working group, as well as giving a clear indication of the work that is not being undertaken there. This should highlight the "technical" versus "policy" split – the IETF work is on a technical specification that are designed to allow services to be operated in compliance with whatever policies are set by the appropriate (non-IETF) bodies.

The purpose of this document is to give those policy-makers sufficient insight into the technical capabilities of the CRISP working group output to understand the broad range of policy possibilities it will support. The hope is that output represents a sufficiently broad toolset that policy-makers will be able to focus on policy development, without having to keep coming back to the question of technical feasibility for each potential policy choice.

2 CRISP – an overview

As are all IETF working groups, CRISP is an open group, working to achieve and support an open standard.

The goal of the IETF's CRISP working group is to develop the requirements for, and identify a protocol that could be used to provide a service for accessing Internet registry information, such as domain registry information. Although this is commonly referred to as the "WHOIS" service today, the working group started from the premise that the existing WHOIS protocol specification is too limited, technically, to be modified to meet today's needs. A clean break – from the WHOIS name, and the underlying protocol – was made.

The top 3 challenges in establishing a registry information service protocol are:

1. Many different user constituencies need to be able to access and use the information.
2. There needs to be some level of standard to facilitate finding and accessing the authoritative information.
3. The information will be offered from a wide variety of sources (service operators) that exist under diverse legal and policy systems and have different service requirements.

Note that these challenges exist for creating the next generation of WHOIS service for domain registrations, as well as for any other registry (e.g., IP address allocation registries, ENUM telephony registrations, etc).

The objectives of the IETF CRISP working group include determining how, technically, to provide a standard protocol and service that:

- have a standard set of queries to support well-known, legitimate uses of the domain registration WHOIS service;
- work in a properly internationalized world of service;
- meet the collected requirements on the service (and protocol) responding to those queries;
- allow users to find the authoritative source of information for their queries; and
- allow providers of the information to support whatever local policies they may have (within their organization; within their geographic region; within their service contracts).

In other words, the IETF is building the tool that can be used to offer a global registry information service, which is a technical activity. The IETF is not attempting to determine or mandate what the global (or local) policies should be.

3 CRISP Requirements

The CRISP working group has documented a set of requirements for the intended service and protocol. These exist in draft form, and are available from the CRISP working group web page, http://www.ietf.org/html.charters/crisp-charter.html.

Given that the CRISP working group is not attempting to mandate global (or local) policies, it must be understood that these are requirements for the *toolset*; they describe how to structure a protocol that will be able to support the policies that may need to be

applied in order to offer the service. For example, there are requirements that say the protocol "must" support a certain feature; these requirements are *not* asserting that individual service operators "must" enable that feature.

As currently defined, the document consists of 4 major sections:

1. Introduction & scoping
2. Definition of communities of applicability
3. Functional requirements
4. Feature requirements

An overview of the scope and communities of applicability are given here (above, and Section 4.2, respectively).

3.1 Functional Requirements

The CRISP functional requirements distinguish between requirements on a protocol that would support the eventual service, and functional requirements for a service to be able to participate in a network of services (i.e., interoperate at a service level).

The general "registry information service" requirements include, but are not limited to:

- Mining Prevention: providing some technical means to guard against massive mining of the information base.
- Minimal Technical Reinvention: to promote the ease of standing up a service, and create client software that will use it.
- Standard and Extensible Schemas: facilitating interoperation
- Level of Access: unlike today's WHOIS, not all data need be equally accessible by all users of the service
- Client Processing: facilitating the creation of client software that can automatically extract relevant details from the service's responses.
- Decentralization: the protocol must not require that all data be aggregated in some central repository in order to provide the service.
- Authentication Distribution: the protocol itself must not require individual service operators to participate in a single, standard authentication system.

The domain registry-specific functional requirements include (but are not limited to):

- Lookups: e.g., the protocol must allow a nameserver lookup given a fully-qualified host name or IP address of a nameserver.
- Searches: e.g., the protocol must allow domain registrant searches by either exact name or partial name match with the ability to narrow the search to registrants of a particular TLD. Note that this is a *protocol* requirement. A *service* may elect to restrict the accessibility of this search.
- Result Set Limits: the protocol must include a provision for allowing a server operator to express a client search limit.

- Internationalization

The reader is referred to the CRISP requirements document itself for a full listing and specification of the functional requirements.

3.2 Feature Requirements

In addition to the functional requirements, the CRISP working group has also identified some feature requirements, including:

- Client Authentication: there needs to be a mechanism for presenting credentials information to the servers.
- Referrals: in order to be able to knit disparate servers together into a cohesive global service, the protocol must provide a mechanism for referring client software to other servers.
- Common Referral Mechanism: the referral mechanism must be commonly understood, in order to allow interoperability.

Again, the reader is referred to the CRISP requirements document itself for a full listing and specification of the feature requirements.

The remaining sections provide more detail on specific experiences from the RFC954 WHOIS reality, and illustrations of potential authentication and privacy management tools for this next generation of registry information services.

4 Applicable whois Experience

4.1 Uses of the RFC 954 Protocol

The name of the title of RFC 954 is "NICNAME/WHOIS". Its port number is registered with the IANA as "nicname", but the protocol is commonly known today as "whois". It was first described in RFC 812 in 1982, a year before the first description of DNS in RFC 882 (the current base RFC's for DNS are 1034/1035).

Because of this history, the whois protocol extends beyond the scope of domain registries and registrars. Therefore it is impractical to try to determine policy for the entire scope of whois as it is used today.

At the least, there are three major types of information held in whois servers: domain registration, IP address allocation, and IP routing specification. Some whois servers contain all three types of information. And there are servers that contain other types of data that do not fall into any of these categories (e.g. whois.abuse.net).

4.2 Users of the RFC 954 Protocol

With each type of server, there are various types of users. And because the situations for which RFC 954 have been put to use cannot be enumerated, it is impossible to also enumerate all the users.

However, it is possible to list some of the major categories of users with regards to domain registration data. Such categories are:

- Registrants
- Registries
- Registrars
- Service providers and network operators
- Intellectual property holders
- Law enforcement
- Certificate authorities
- DNS users
- Abusive users

While this is probably not a complete list of users seeking domain registration information, it likely encompasses the majority of them.

Defining the major users of the system and the type of data in the system allows for the definition of possible use cases. From use cases it is possible to define policy requirements.

5 Authentication Mechanisms and Authorization Scenarios

The policies in place for conducting authentication via nicname/whois are but a small subset of what is possible with a more robust and modern protocol. Because nicname/whois does not define any authentication mechanism, authorization policy based on the nicname/whois authentication methods is restricted to:

1) Anonymous access
2) Access based on the client's IP address

The second authentication mechanism is an artifact of the Internet Protocol and was never intended to be an authentication mechanism. Hence, it is not a very good authentication mechanism.

However, the CRISP proposed protocols IRIS and FIRS will allow for a much broader range of authorization policies because both protocols support many more authentication mechanisms. The following scenarios discuss various authentication mechanisms and the various authorization possibilities available because of them.

5.1 Passwords

User passwords are a simple form of user authentication with which most people have a familiarity. They are easy to use, and various authentication mechanisms exist to allow them to be sent over the Internet without exposing them to attackers.

Passwords allow for user-based authorization, where policy is dictated based on information about the user assigned to the password. This user-based authorization can then be compared in the larger operating context. The following gives an example:

- If user has not logged in
  - Only show nameservers
- If user is Alice
  - Allowed to see email addresses of all contacts
- If user is Alice and is viewing Alice's data
  - Allowed to see everything

In the authorization of users across multiple domain registries and registrars, there is a non-trivial operational aspect with regard to the distribution of user passwords.

5.2 One-Time Passwords

One-time password systems are cryptographic mechanisms designed to keep passphrases from being sent in the clear over unencrypted sessions. However, their design limits their use to a finite number of authentications with both parties keeping track of the number of uses. While this sounds complex, to the user these systems seem very much like a standard password system.

However, they allow for a sequence-based authorization policy. That is, the type of authorization the user may obtain may be altered based on the number of times the user authenticates. Therefore, such a policy might be that Alice is given a passphrase good for 10 one-time passwords. After Alice has authenticated 10 times, she can no longer access the system.

5.3 Digital Certificates

Digital certificates use a branch of mathematics called public key cryptography to conduct authentication. Used in conjunction with TLS (i.e. SSL), they also allow for server authentication and session encryption.

Digital certificates offer the following authorization models:

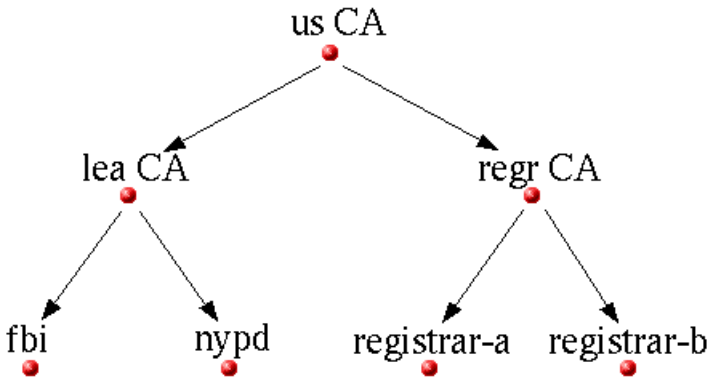
- user-based: This is the same as with passwords. Authorization is based on the user associated with the digital certificate.
- chain-based: Authorization is based on the chain of digital certificates that have signed the authenticated digital certificate.
- attribute-based: Authorization is based on information found in the digital certificate.
- time-based: Every certificate is given a shelf life. Once they expire, they are no longer valid for authentication.

Chain-based authorization is based on the tree structure inherent in digital certificates used by TLS (these are called X.509 certificates). Each certificate is signed by the private key of an upstream certificate until the chain of certificates ends at a root certificate. A single digital certificate may be used to sign multiple downstream certificates, but a certificate may only be signed by one upstream certificate. Therefore,

the certificates are organized in a tree model, and the path from a certificate back to the root is known as the chain. An organization that use the private key corresponding to their certificate to sign other certificates is called a Certificate Authority, or CA.

The chain-based authority differs from the user-based authority because the authorization is based on the identity of one of the certificates in the chain instead of the identity of the user's certificate. Therefore, a server may only need to know the identity of a CA to authorize a user. The advantage is that there will likely be far less CA's than users.

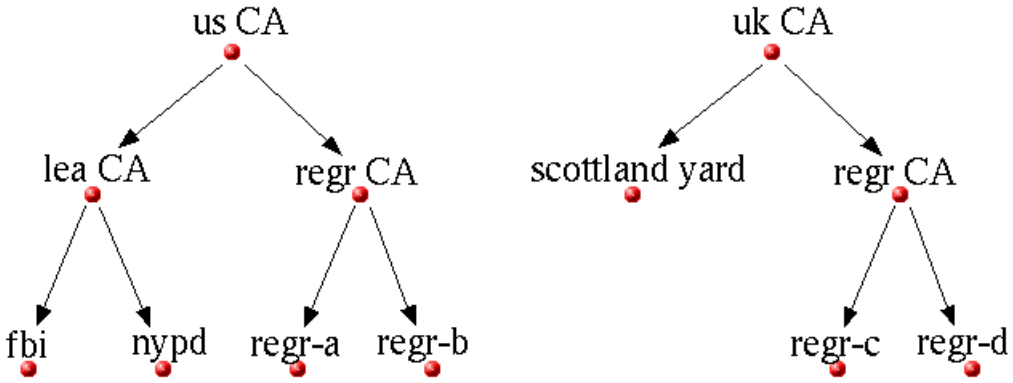
The following figure shows an example tree of digital certificates.



Based on this tree, the following example policy may be put into place:

- If the certificate is signed by the "lea CA"
  - Allow access to all contact data
- If the certificate is signed by the "regr CA"
  - Allow access only to all domain and registrant data

Servers may also trust multiple root certificates, and therefore use multiple certificate trees, as shown in the following figure.



An attribute-based authorization policy takes advantage of the fact that a certificate is digitally signed by the private key of another certificate. If its contents are altered, the certificate will not validate. Therefore, authorization may be based on information carried within the certificate.

The figure above is a graphical representation of an example digital certificate. Based on the information within the certificate, the following example policy may be used:

- If the "Type" attribute in the certificate equals "LEA"

- Allow access to all contact data

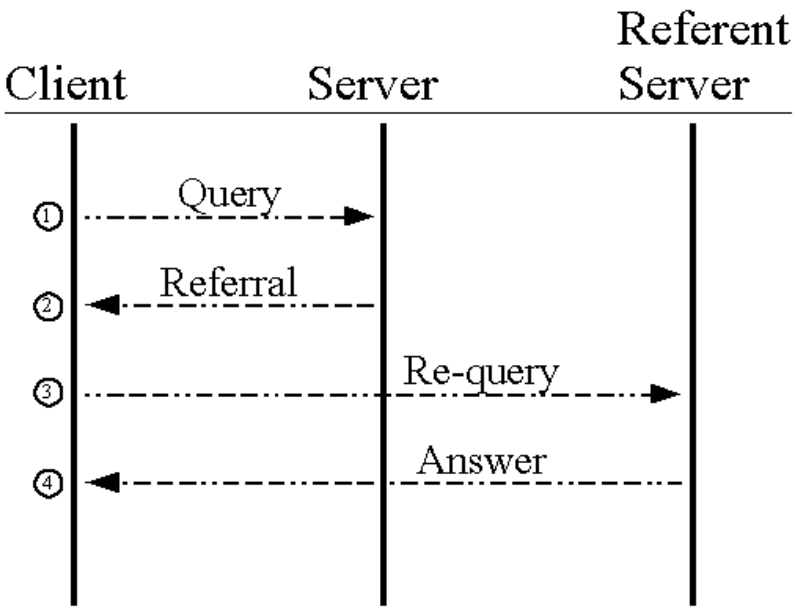
- If the "Type" attribute in the certificate equals "Registrar"
  - Allow access only to all domain and registrant data

Finally, digital certificates offer the benefit of non-repudiation that password systems do not poses. This is helpful with audit trails. With password systems, both parties know the shared secret. If trust breaks down between them, one party may accuse the other party of using the shared secret without their knowledge and/or approval.

However, digital certificates rely on public key cryptography. Public key cryptography has two basic components: the public key and the private key. The public key is what is put into the certificate and may be given to all parties. The private key is always kept only by the person to whom the certificate is assigned. Not even the CA's know the private key. In order to authenticate, a user needs both their certificate and the corresponding private key. Therefore, only the party owning the certificate may use it.

5.4 Referrals

The CRISP protocols allow a server to pass extra information via a client to a referent server. If this information were to contain some sort of authentication data, then this allows for a referee-based authorization policy.



A referee-based authorization policy could focus on many aspects of the information given by the referee. For example, such information might be the identity of the referee, or it might be flags indicating the level of access. The following is an example referee-based authorization policy:

- If the referee is the Department of Justice

    - Allow access to all contact data
- If the referee has set the "Registrar" flag
    - Allow access only to all domain and registrant data

When combined with known cryptographically secure functions, the referent server can have a high-degree of confidence that the client has not altered the referee information. If that cryptographically secure function uses public keys, then non-repudiation may also be attached to this type of policy.

6 Conclusion

The IETF CRISP working group has canvassed a wide range of input and applied IETF community technical expertise to set down a technical description of requirements for developing a registry information service. In the judgment of the IETF community, it is technically feasible to fulfill the protocol requirements as described.

The CRISP working group remains agnostic about the definition and application of policy governing content of and access to registry information. Therefore, moving forward, it is critical that the appropriate communities continue to discuss and come to consensus on what those policies should be for domain name registries and registrars.

If this document has been successful, it has communicated that the technical requirements and selection of protocols is readily and appropriately handled within the IETF community, in the CRISP working group.

Leslie Daigle (leslie@thinkingcat.com)

Andrew Newton (anewton@ecotroph.net)