A RESTful Web Service for Internet Name and Address Directory Services

ANDREW NEWTON, DAVE PISCITELLO, BENEDETTO FIORELLI, AND STEVE SHENG

Andy Newton is the Chief Engineer at the American Registry for Internet Numbers, where he oversees system and data architecture and leads multiple teams of software developers.
andy@arin.net

Dave Piscitello is a Senior Security Technologist for ICANN. A 35-year Internet veteran, Dave currently serves on ICANN's Security and Stability Advisory Committee and the Internet Policy Committee of the Anti-Phishing Working Group (APWG).
dave.piscitello@icann.org

Benedetto Fiorelli is a Senior Software Engineer at RIPE NCC. He has contributed to vision, design, and programming of production-grade software for different types of organizations across very different domains. His main areas of expertise are software design, service-oriented architecture, data modeling, and quality software management.
bfiorell@ripe.net

Steve Sheng joined ICANN in 2009 as a Senior Technical Analyst. In this position, he supports projects of ICANN's Security and Stability Advisory Committee (SSAC) and provides research support for projects in the policy department.
steve.sheng@icann.org

Domain name and Internet Protocol address registry operators collect point of contact and other information associated with the delegated administration and use of domain names or IP addresses. Registries and other operators make this information available via the Whois protocol (RFC 3912 [1]) and Web-based interfaces. In this article, we describe how directories based on representational state transfer (REST [2]) Web services could support features that have been identified as desirable or beneficial for domain name and IP address registration data.

By registration data, we refer to the information that registrants provide in order to obtain the right to use a domain name or to have an IP address space allocated for use. For domains in the generic top-level domain space, these data elements are specified in the Registrar Accreditation Agreement and individual registry agreements with ICANN. For IP allocations, each Regional Internet Registry specifies its own set of data elements. In this article, we provide summaries of prototype and early production deployments of RESTful Web services, the current status of standardization efforts, and future plans for this work.

Background

Created in the 1980s, NICNAME/WHOIS began as a service used by Internet operators to identify network administrators. Today, we access registration data via Whois services to identify and contact the registered users of Internet (IP) address allocations and domain names for business matters and on matters related to trademark protection, criminal activities (phishing, spam, botnets), to verify online merchants, and more.

Whois services have evolved in a largely ad hoc manner for over two decades, and not without considerable scrutiny and criticism. Many of the deficiencies and desired additional features are mentioned in reports from ICANN's Security and Stability Advisory Committee [3, 4, 5, 6], in reports of ICANN supporting organizations [7], and by members of the Regional Internet Registry community [8, 9]. Among these are:

Need to support internationalized registration data. The Whois protocol has no standard mechanism for clients and servers to signal a character set. Monolingual users whose language cannot be represented using characters from the US-ASCII7 character set are greatly disadvantaged by this limitation. Unicode and multilingual support is widely available from Web applications. This internationalization is becoming increasingly necessary for registration data. The

proliferation of non-standard signaling conventions by registry operators does not scale.

Need to standardize and enhance service. The Whois protocol describes exchanges of queries and responses between a client and a server over a specific TCP port (43). The only constraint the specification imposes on query and message formats is that they must be terminated with ASCII line feed and carriage return characters. The specification does not define standard formats or encodings. It does not have a schema for replies or error messages. The resulting variability across client and server implementations detracts from the quality and usability of Whois "port 43" client as well as Web-based applications. In particular, the lack of uniformity inhibits or adds complexity to machine parsing (automation).

Need for security services. Users or applications access Whois services anonymously, requiring no identity assertion, credentialing, or authentication. Few methods are used to restrict access to Whois servers other than rate limiting based on IP address-level access controls. The lack of authentication mechanisms inhibits adoption of effective user- or group-level access controls, auditing, or privacy measures, features that are typical for directory services [10].

Of these, we believe that developing support for internationalized registration data and seeking uniformity of service are matters that require urgent attention. The security services mentioned, while important, are largely absent from Whois services today, and their inclusion is a matter for policy development.

We posit that the community should consider universal adoption of Unicode and should adopt markup languages and widely accepted data interchange formats. Modern applications benefit from adoption of these instead of free-form text. We further posit that Whois users would benefit from efforts to standardize on signaling and error messages. We recognize that some Whois users view security services as desirable and beneficial, and believe that any framework and protocol considered for a registration data directory service must be sufficiently extensible to support security services should they become requirements.

Why RESTful?

In choosing a RESTful approach, we began by considering first principles for protocols. All protocols have a control part, which defines message formats, security features, signaling, encodings, and errors, and a data part, which is the information that users see and applications process. The control part of the existing Whois protocol specifies little about the data part and would require a virtual rewrite to satisfy the needs we identified earlier. Development of the control part would be encumbered by the need to provide backwards compatibility for the numerous ad hoc extensions various providers of Whois services have adopted in the absence of a formally specified control part in RFC 3912. Today, non-Latin characters are supported differently by several domain name registries. In the extreme, users or applications must know these for each registry they query.

IRIS [11], envisioned by some as a successor to Whois services, has a control part that is specific to the data part (i.e., registry-specific). IRIS also requires its own application-specific transport to operate correctly over TCP or UDP. Whereas a Whois service has very little control part, experience with IRIS has proved its control part is complex and requires sufficient investments in application development for both clients and servers that it was an impractical choice for a widely offered (and, in some circumstances, mandatory) service that typically does not generate revenue.

By juxtaposing Whois against IRIS, we were able to derive a list of desirable criteria for a registration data directory service. The control part should be sufficiently flexible to satisfy domain name and Internet address registry needs. It should support structured, typed data and international encodings. If possible, it should leverage or integrate well with existing (Web) application infrastructures. It should readily accommodate the inclusion of security as mandatory or optional features.

The American Registry for Internet Numbers (ARIN) had been experimenting with Representational State Transfer-based Web services, and their results encouraged the authors and their colleagues to experiment as well. Prototypes developed by ICANN and RIPE further reinforce our belief that a RESTful approach might satisfy the needs we have identified.

Benefits of Representational State Transfer-based Web Services

Representational State Transfer-based Web services define a pattern of usage with HTTP to create, read, update, and delete (CRUD) resources. Resources are addressable as Universal Resource Locators (URLs). The RESTful framework leverages the HyperText Transfer Protocol (HTTP) infrastructure, including caching, referrals, authentication, version control, and secure transport (HTTPS). The programming API accommodates Unicode and numerous markup languages, supports signaling and standard error messages, and runs on top of standard Internet transport protocols.

A RESTful approach allows us to identify structured data types and incorporate these in URL patterns to refer unambiguously to individual resource objects. The existing Whois Web interfaces demonstrate our ability to support a wide client base, including ordinary Web browsers; command-line utilities like curl, wget, and xmllint; and embedded client implementations such as libcurl and various libraries for Perl, PHP, and Java.

RESTful Web services implemented on platforms that support SSL/TLS [12] for other purposes (such as registration services) may be able to leverage this implementation to provide authenticity of origin, transport confidentiality, (sub) authentication, and other security services that have already been implemented around the HTTPS framework.

Benefits of Structured and Typed Registration Data

Several benefits can be derived from establishing data conventions or standards for Web-based queries and responses to registration databases or repositories. First, they allow the signaling or delivery of metadata about the registration data, such as the language of the registration data. Second, they facilitate search mechanisms.

Some consumers of domain name registration data want to be able to search registration data using attributes or object types in much the same way that they can currently perform such searches using, for example, the APNIC Whois service [13]. Today, submission forms typically only accommodate rudimentary query arguments such as <domain name> or <IP address>. Certain communities can benefit from queries for such information as sponsoring registrar or name server information for domain name registrations, or autonomous system numbers, networks, or reverse DNS delegations for Internet address registrations.

Structured, typed data facilitates such queries. Such searches are performed now, often by legal obligation, through ad hoc requests. Providing a standard mechanism for performing such searches could lower the costs to registries and searchers in undertaking this duty. Having a standard, machine-parsable format is especially valuable for performing large-scale data analysis across the registration databases. With appropriate controls to balance privacy interests, such a facility would provide a means of greatly enhancing our understanding of some types of activity on the network.

Experience with RESTful Registration Data Directory Services

Using the prototypes and early production experiences we describe in this section, we demonstrate that adopting a RESTful framework for registration data directory services offers ease of implementation and ease and rapid integration with existing deployments, and, further, that registration data defined using XML offers the opportunity to create flexible submission queries and easily parsable responses using a "universally understood" meta-language that accommodates internationalized registration data requirements for domain name and Internet address registries.

ARIN Whois-RWS

In 2009, ARIN completely rewrote its traditional Whois service as part of a major software re-engineering effort, enhancing this service with a RESTful Web interface, Whois-RWS.

Our Whois-RWS mirrors the major structured data types used by ARIN in URL patterns:

`/rest/poc/XXXX` for points of contact
`/rest/net/XXXX` for networks (IP prefixes)
`/rest/org/XXXX` for organizations
`/rest/asn/XXXX` for autonomous system numbers
`/rest/rdns/XXXX` for reverse DNS delegations

The relationship between these structured types was easy to express using HTTP URLs.

Searches make use of HTTP URL matrix query parameters. Such queries do not utilize a specific identifier to narrow the result set to a single intended item. Multiple query parameters allow the querying party to specify multiple, distinct search inputs. For example, "/rest/pocs;first=John&last=Doe" is a request for points of contacts where the contact's given name is "John" and surname is "Doe".

While XML is the primary output format, the user can direct Whois-RWS to render results in XHTML [14], JSON [15], or plain text by using the HTTP Accept header or by appending a file extension type to the query URL (e.g., .xml, .txt, etc.). The HTTP Accept header is a standards-based method for requesting a specific MIME type. This allows Web browsers to request XHTML or XML styled with CSS automatically so that end users can directly view data in Whois-RWS. When rendered in XML styled with CSS, the output is both easily machine parsable and user friendly.

The ARIN team uses the Relax NG formal schema language [16] to define the XML so that programmers can easily determine what to expect in the response. Exten-

sion points are clearly defined in the schemas, which can thus be extended while preserving backwards compatibility. ARIN has already utilized this extensibility when it rolled out new reverse DNS delegation infrastructure to better support DNSSEC. We were able to create Whois-RWS quickly and easily by reusing large software components hitherto developed for ARIN's Web portal system, ARIN Online. The reuse of standard Web software components gave us time to add additional features, such as CIDR query support and a new near-real-time data replication system. Initial prototypes of Whois-RWS were greenlighted in the summer of 2009, and Whois-RWS was released as a production system in July 2010. By this time, customers were using Whois-RWS to augment their own IP address management systems, and one Flash-based application had been written to take advantage of the machine-readable information. As of March 2011, over 40% of Whois queries served come through the RESTful Web interface.

ICANN

The ICANN team began to experiment with a RESTful service for domain name registration data directory service in 2010. Our goals for this pilot were to understand whether a RESTful Web service could support requirements for the submission and display of internationalized domain names and registration data, as the team understood the requirements at that time [17]. Having performed several studies that involved machine parsing and normalization of tens of thousands of registration records collected from a random and large set of registrars [18, 19], we also sought to improve usability of domain name registration data for such purposes by defining standard, extensible formats for the data that would also accommodate future changes. Lastly, we sought to understand the design tradeoffs associated with a RESTful service versus an enhanced Whois for domain registration data, and the approximate cost and complexity of implementation.

Similar to ARIN's implementation, ICANN's prototype service uses the HTTP protocol and conforms to the REST architecture. The client sends its request with the following URL structure:

`/rest/domain/XXXX` for domain name request
`/rest/contact/XXXX` for contact request (by contact ID only)
`/rest/host/XXXX` for host request

The client signals the preferred format using the standard HTTP Accept header. The client can also signal the preferred format by adding a DOS-file-style extension to the resource. The server provides responses in XML, HTML, and plain text format.

The ICANN prototype uses a formal XML schema language so that programmers can easily determine what to expect in the response. The data schema largely reuses the data schema defined in the Extension Provisioning Protocol (EPP) [20, 21, 22, 23]. The ICANN team chose the EPP schema to leverage the existing standards associated with registry-registrar information transfer and to minimize reinvention: ICANN accredited registries and registrars use EPP for their operations, so they are familiar with the schema and may be able to reuse existing software.

The prototype demonstrates that the REST architecture with EPP supports internationalized domain names and registration data in the following way. Queries can be expressed using Internationalized Resource Identifiers (IRIs) [24] and thus

accommodate non-ASCII input. EPP data schema accommodate internationalized contact name, organization, and address information represented in unrestricted UTF-8 using the attribute (type="loc") from RFC 5733.

The ICANN team continues to experiment with the RESTful service to develop and propose standardized error handling and to find better ways to signal encodings other than UTF-8. We will investigate whether the code base is suitable for further work as an open source project. While we currently use "canned data," we are considering hosting an experimental service for the IANA registries or certain TLDs that ICANN manages (such as the .INT and .ARPA).

RIPE Database REST API

A large number of organizations and individuals use the RIPE Database. Historical and technical aspects of the Whois protocol, and in particular the Routing Policy Specification Language (RPSL [25]), influence how these clients interact with the RIPE Database.

RPSL specifies much more than routing policies, describing, in practice, more than 20 different types of RPSL objects. It is more a formalization of the way policy records have been stored and exchanged in the existing Whois systems since the late 1980s than a high-level domain language specification. As a result, the policy specification language and its extensions are tightly coupled to the way policies are stored in the existing Whois systems and vice versa; thus any new system that implements the language essentially reproduces a Whois service.

These processes are modeled on a human-centered workflow. They are not optimized for building new services, extending existing ones, or building tools on top of them. Thus, client applications can quickly become overly complex when dealing with RPSL. Many become too expensive to be extended or maintained. The RIPE Database Group observed needs for simpler interfaces and machine-parsable data formats that would simplify the development of services and tools and increase the value of registries by exposing new domain-specific interfaces.

The RIPE Database Group developed a layer of Service Provider Interfaces (SPI) and a data schema simple enough to be agnostic of the underlying registry implementation. The SPI allows composition of domain-specific services and also makes possible real-time interoperation between registries. The RIPE team chose REST because HTTP is the most accessible protocol and the HTTP methods, resource locator protocol, HTTPS, and other features provide a flexible and proven framework for stateless services. For the representation schema we have designed a relaxed attribute-oriented XML Schema. We only apply a structural validation via XML Schema Definition [26]. After some testing we decided to remove any form of attribute or type validation in order to reuse the same schema on different RPSL flavors.

The services support JSON, HTML, and plain text, all derived via XSL [27]. HTML and text transformation demonstrate the transformation powers of XML and how resource navigation can be accomplished using any HTML browser. As is the case in the ARIN implementation, content negotiation is done using HTTP headers or by appending a file extension to the request URL.

The query services can be used on any RPSL-based Whois server or mirror. It is possible to execute the same Lookup or Search request on all the Regional Internet

Registries and return all the responses as a unique set of resources. The query services feature:

- Single-resource lookup service: Given a primary key, a type, and a registry, it always returns one and only one object. It can also be used to identify resources by URL bookmark.
- Resolution of referenced resources: Given a resource, all the attribute values that represent references to other resources contain an xlink anchor that can be followed to navigate and browse networks of resources.
- The client can navigate through any network of resources via xlinks without requiring any stateful information to be stored on the servers. This comes as a benefit of the two previous features.
- Normalization of continuation lines, end-of-line comments, and other RPSL intricacies, and normalization of comma-separated values when they represent references to multiple resources.

CRUD interfaces also can be used on any RPSL-based Whois server or mirror by building adapter modules for the different update mechanisms provided by different registries (given that they adopt the same set of resource types). The CRUD Services split the single overloaded generic update interface provided by Mail Updates and Syncupdates into separate low-level interfaces, each defining a simpler contract, designed for programmatic use rather than for human interaction. The new CRUD Services provide a Delete interface that only requires a primary key, a type, a registry identifier, and one or more passwords. It is the equivalent of a lookup but is executed with the HTTP Delete method. This is exposed only on HTTPS, through a request of the form HTTP DELETE: https://lab.db.ripe.net/whois/delete/test/person/pp16-test?password=123. The server responds with an HTTP Status code indicating success or failure. Failure conditions return unique status codes.

We also prototyped some attribute modification services on top of the CRUD methods. With only one request, we can implement complex update workflow. For example, using one HTTP request it will be possible to execute commands such as:

- Replace all the attributes of a given type with a new set of attributes.
- Remove all the attributes that have a value matching the given regular expression.
- Add this set of attributes after the Nth attribute of type X.

Examples of the lookup and search services can be performed using the following URLs:

http://apps.db.ripe.net/whois/lookup/ripe/organisation/ORG-BME1-RIPE.xml
http://apps.db.ripe.net/whois/lookup/ripe/route/193.6.23.0/24/AS2547.xml
http://apps.db.ripe.net/whois/search.xml?flags=r&source=ripe&source=apnic&source=afrinic&query-string=AS2547

The technical documentation of the RIPE Database REST API can be found at [28].

Findings and Conclusions

The prototyping and early production experiences support our claim that a RESTful approach is a simple yet elegant solution to the problem set we have identified in this paper. We are able to support internationalized registration data (and, generally, structured and typed data), provide unambiguous signaling, and improve

error reporting. We are able to leverage existing client and server infrastructures and provide security services, including transport confidentiality and integrity checking, authentication, and data filtering, in an extensible manner, again with the prospect of being able to leverage implementations and Web infrastructure that makes use of security services today.

Future Work

We intend to continue collaborative experimentation and further development of prototypes. ARIN's production Whois-RWS will provide valuable insight into the features most commonly used. Users may identify additional features or may assist in identifying areas for improvement.

We have requested and received approval from the Internet Engineering Task Force (IETF) Applications area director to present this work to the technical community. We have submitted a draft requirements specification to introduce a series of documents that define the overall problem and some available solutions. The series includes Requirements for Internet Registry Services, descriptions of the ARIN, ICANN, and RIPE Internet Registry Service APIs, and a description of RESTful Whois. Our intent is to publish one of the API specifications as a standards track document and the remainder (including this memo) as informational documents. To participate in discussions about work on this next-generation Whois technology at the IETF, join the Whois-based Extensible Internet Registration Data Service (WEIRDS [29]).

Acknowledgments

The authors recognize the following colleagues for their cooperation or contributions to this joint activity: Joe Abley (ICANN), Francisco Arias (ICANN), Kim Davies (ICANN), Mark Kosters (ARIN), Paul Palse (RIPE), Kaveh Ranjbar (RIPE), Andrew Sullivan, Leo Visgoda (ICANN).

References

[1] L. Daigle, Whois Protocol Specification, RFC 3912, 2004: http://www.rfc-editor.org/rfc/rfc3912.txt.

[2] Roy Fielding, "Architectural Styles and the Design of Network-based Software Architectures," PhD dissertation, University of California, Irvine, 2000: http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm.

[3] ICANN Security and Stability Advisory Committee (SSAC), *Whois Recommendation of the Security and Stability Advisory Committee* (SSAC publication No. 003), 2003: retrieved from http://www.icann.org/en/committees/security/sac003.pdf.

[4] ICANN Security and Stability Advisory Committee (SSAC), *Is the Whois Service a Source for Email Addresses for Spammers?* (SSAC publication No. 023), 2007: retrieved from http://www.icann.org/en/committees/security/sac023.pdf.

[5] ICANN Security and Stability Advisory Committee (SSAC), *SSAC Comment to GNSO regarding Whois Studies* (SSAC publication No. 027), 2008: retrieved from http://www.icann.org/en/committees/security/sac027.pdf.

[6] ICANN Security and Stability Advisory Committee (SSAC), *Domain Name Registration Information and Directory Services* (SSAC publication No. 033), 2008: retrieved from http://www.icann.org/en/committees/security/sac033.pdf.

[7] ICANN Generic Names Supporting Organization (GNSO), *Inventory of Whois Service Requirement Final Report* (Marina Del Rey, CA: ICANN), 2010: retrieved October 21, 2010 from http://gnso.icann.org/issues/whois/whois-service-requirements-draft-final-report-31may10-en.pdf.

[8] A. Newton, "Replacing the Whois Protocol: IRIS and the IETF's CRISP Working Group," *IEEE Internet Computing*, vol. 10, no. 4, July–Aug. 2006, pp. 79-84.

[9] Cathy Murphy, "Some RIR Input regarding Whois Deficiencies," presented at IETF 53, 2002: retrieved from http://www.ietf.org/proceedings/53/slides/crisp-1/sld001.htm.

[10] ICANN Security and Stability Advisory Committee (SSAC), *Domain Name Registration Records and Directory Services* (SSAC publication No. 33), 2008: retrieved from http://www.icann.org/en/committees/security/sac033.pdf.

[11] A. Newton and M. Sanz, "IRIS: The Internet Registry Information Service Core Protocol," 2005: http://www.rfc-editor.org/rfc/rfc3981.txt.

[12] T. Dierks and E. Rescorla, "The Transport Layer Security (TLS) Protocol, Version 1.2," 2008: http://www.rfc-editor.org/rfc/rfc5246.txt.

[13] APNIC Whois Search: http://www.apnic.net/apnic-info/whois_search2.

[14] World Wide Web Consortium, Extensible HyperText Markup Language (2nd ed.), 2002: retrieved from http://www.w3.org/TR/xhtml1/.

[15] JavaScript Object Notation (JSON), 2011: retrieved from http://www.json.org/.

[16] Relax NG, 2011: retrieved from http://relaxng.org/.

[17] ICANN Security and Stability Advisory Committee (SSAC), *Display and Usage of Internationalized Registration Data: Support for Characters from Local Languages or Scripts* (SSAC Publication No. 37), 2003: retrieved from http://www.icann.org/en/committees/security/sac037.pdf.

[18] D. Piscitello and S. Sheng, "Abuse of Domain Name Privacy Protection Services," 2010: retrieved from http://securityskeptic.typepad.com/the-security-skeptic/2010/04/domain-name-privacy-misuse-studies.html.

[19] D. Piscitello and S. Sheng, "Abuse of Domain Name Privacy Protection Services: Act Deux," 2010: retrieved from http://securityskeptic.typepad.com/the-security-skeptic/2010/10/misuse-of-domain-privacy-protection-services-act-deux.html.

[20] S. Hollenbeck, RFC 5730, Extensible Provisioning Protocol (EPP), 2009: http://www.rfc-editor.org/rfc/rfc5730.txt.

[21] S. Hollenbeck, RFC 5731, Extensible Provisioning Protocol (EPP) Domain Name Mapping, 2009: http://www.rfc-editor.org/rfc/rfc5731.txt.

[22] S. Hollenbeck, RFC 5732, Extensible Provisioning Protocol (EPP), Host Mapping, 2009: http://www.rfc-editor.org/rfc/rfc5732.txt.

[23] S. Hollenbeck, RFC 5733, Extensible Provisioning Protocol (EPP) Contact Mapping, 2009: http://www.rfc-editor.org/rfc/rfc5733.txt.

[24] M. Duerst and M. Suigard, RFC 3987, Internationalized Resource Identifiers (IRIs), 2005: http://www.rfc-editor.org/rfc/rfc3987.txt.

[25] RPSL Reference Guide: http://www.irr.net/docs/rpsl.html.

[26] XML Schema Definition: http://www.w3.org/XML/Schema.

[27] XML Stylesheet Language: http://www.w3.org/Style/XSL/.

[28] Benedetto Fiorelli, RIPE Database REST API: http://labs.ripe.net/ripe-database/database-api/api-documentation.

[29] WEIRDS Info Page: https://www.ietf.org/mailman/listinfo/weirds.