Registration Data Access Protocol (RDAP) for Digital Forensic Investigators

by Bruce Nikkel

nikkel@digitalforensics.ch

Elsevier Digital Investigation Journal
https://doi.org/10.1016/j.diin.2017.07.002

November 28, 2017

Abstract

This paper describes the Registration Data Access Protocol (RDAP) with a focus on relevance to digital forensic investigators. RDAP was developed as the successor to the aging WHOIS system and is intended to eventually replace WHOIS as the authoritative source for registration information on IP addresses, Domain Names, Autonomous Systems, and more. RDAP uses a RESTful interface over HTTP and introduces a number of new features related to security, internationalization, and standardized query/response definitions. It is important for digital forensic investigators to become familiar with RDAP as it will play an increasingly important role in Internet investigations requiring the search and collection of registration data as evidence.

Keywords: RDAP, WHOIS, registrar, registry, internet investigation, cyber investigation, network forensics, cyber forensics

Contents

1 Historic overview of WHOIS

The original specification for WHOIS was written in 1982 as Request For Comments (RFC) document RFC-812. It was created to provide a directory of contact information for users of systems and networks on the ARPANET. The WHOIS specification was updated several times and in 2004 the RFC-3912 became the final WHOIS RFC published[1].

The WHOIS protocol is simply a TCP connection to port 43 of a WHOIS server, and a query submitted with a trailing newline. To illustrate the simplicity of this interaction, a regular telnet client is used here to fetch WHOIS information for the domain example.com. No special WHOIS client software is necessary:

```
$ telnet whois.iana.org 43
Trying 192.0.47.59...
Connected to ianawhois.vip.icann.org.
Escape character is '^]'.
example.com
% IANA WHOIS server
% for more information on IANA, visit http://www.iana.org
% This query returned 1 object

domain:       EXAMPLE.COM

organisation: Internet Assigned Numbers Authority

created:      1992-01-01
source:       IANA

Connection closed by foreign host.
$
```

Every WHOIS server operator has their own command syntax for queries and searches, and the output format of commands will vary for each server. An overview of the use of WHOIS search queries for digital forensic investigations and an introduction to other concepts discussed here can be found in a previous paper by the author[2].

The original WHOIS database was centralized and operated by a single entity (SRI-NIC). As the Internet grew, management of registration information was delegated to various regional registries who operated their own WHOIS services. There is no standardization for WHOIS query commands and output formats.

A number of attempts were made to address this decentralized and fragmented WHOIS landscape. The Referral WHOIS[3] Protocol - RWHOIS (RFC-1714 and RFC-2167) was designed to facilitate referring queries with a distributed and hierarchical WHOIS architecture. Other proposed (and in some cases, implemented) solutions included more complex X.500 directory services[4] with synchronized databases between registries (the Shared Whois Project - SWIP),

the Cross Registry Internet Service Protocol (CRISP)[5] and Internet Registry Information Service (IRIS) protocol[6]. In 2012 the IETF proposed a working group to develop the Web Extensible Internet Registration Data Service (WEIRDS). WEIRDS was eventually renamed to the Registration Data Access Protocol (RDAP) and further developed as an IETF standards track intended to replace WHOIS. The new RDAP standard is sometimes called "Restful Whois" due to the use of the REST (REpresentational State Transfer) interface for the query and response protocol.

2 Introduction to RDAP

The need for a successor to the WHOIS protocol was clear to the IETF, and the Registration Data Access Protocol, or RDAP, was put forward as the replacement. The WHOIS databases at different registries have been a valuable source of intelligence, evidence, and investigative information for many years. RDAP will continue to provide this information to digital forensic investigators, but with a number of improvements.

2.1 RDAP overview

The RDAP standard is defined in a suite of five RFCs:

- RFC-7480 HTTP Usage in the Registration Data Access Protocol (RDAP)[7]
- RFC-7481 Security Services for the Registration Data Access Protocol (RDAP)[8]
- RFC-7482 Registration Data Access Protocol (RDAP) Query Format[9]
- RFC-7483 JSON Responses for the Registration Data Access Protocol (RDAP)[10]
- RFC-7484 Finding the Authoritative Registration Data (RDAP) Service[11]

An additional study was performed to analyze the output objects from the WHOIS servers of various registries to help with the design of RDAP. The results of this study were documented in RFC-7485, entitled Inventory and Analysis of WHOIS Registration Objects[12]. This analysis also helps RDAP providers ensure that data from unstructured WHOIS services can be mapped to RDAP services without loss of information.

The high level goal of RDAP was to address deficiencies in the WHOIS standard. Four specific issues were listed in RFC-4782 that are addressed by RDAP:

- *lack of standardized command structures*
- *lack of standardized output and error structures*
- *lack of support for internationalization and localization*
- *lack of support for user identification, authentication, and access control*

RDAP provides a number of advantages over WHOIS. It uses a RESTful interface over HTTP to query and receive information. REST is an API which allows HTTP to be used to query a server application for data and receive structured responses. The data returned from an RDAP query is in a well defined JSON format that can be easily parsed with scripts and tools (and is still readable by humans). The HTTP interface can be encrypted (TLS), meaning that the RDAP query and response are secured. This is important for ongoing investigations where queries and responses may be sensitive or confidential, and should not be visibly transmitted in the clear. Access to RDAP data can be authenticated and controlled (dependent on the webserver used).

The amount of information returned in a RDAP query can be selective and based on the authentication provided, for example, authenticated users could have access to more whois data fields than unauthenticated users. This introduces the potential for allowing special investigative access to additional registry data without the need for separate legal assistance infrastructure.

RDAP leverages existing HTTP functionality to provide standard error messages (HTTP 200, HTTP 404, etc.) for queries. The use of redirection (HTTP 301 for example) can be used to refer a client to another, possibly more authoritative, RDAP server. Since the HTTP protocol can be proxied, it is possible to make RDAP requests from behind restrictive firewalls (many corporate networks, for example). RDAP also supports Internationalized Domain Names (IDNs). The RDAP protocol is designed to be extensible, and IANA maintains the list of assigned extensions and JSON values for RDAP.

One major problem RDAP does not solve is the accuracy of the data submitted and stored in the various registries. The accuracy of registry data was an issue with WHOIS and remains an issue with RDAP. The integrity of the data can be guaranteed while it is queried and transferred from an authoritative RDAP server to an RDAP client, but the quality of the registry data submitted by the original registrant depends on the validation checks done by each registry owner (which could be little or none). It is still possible to submit fake or false information to many registries, which will then be served by WHOIS and RDAP servers as is. This is a known risk in the digital forensics community, but even falsified information can be useful or interesting in a forensic or investigative context.

2.2 RDAP HTTP query

The format of RDAP queries is defined in RFC-7482. The standard set of items (object classes) that can be queried using the RDAP protocol include:

- IP addresses (IPv4 or IPv6)
- Network autonomous system numbers (ASNs)
- DNS domain names
- Name servers

- Entities (contact information)

All of these objects are typically served by RIRs (Regional Internet Registries), while DNRs (Domain Name Registries) generally only serve the last three. The RDAP protocol is designed to be extensible, and extensions are maintained by IANA (`https://www.iana.org/assignments/rdap-extensions/rdap-extensions.xhtml#rdap-extensions-1`).

A proper RDAP query is a simple HTTP GET or HEAD request with all the query information embedded in the URL string. The ability to encrypt connections using TLS (HTTPS) is a requirement specified by the RDAP RFCs. The path in the URI (Uniform Resource Identifier) is split into three parts: the location of the RDAP application on the server, the query object type, and the actual query. The query syntax for the standard set of object classes is shown here:

- `https://<rdapserver>/<path>/ip/<IPaddress>`
- `https://<rdapserver>/<path>/ip/<CIDRprefix>/<CIDRlength>`
- `https://<rdapserver>/<path>/autnum/<autonomoussystemnumber>`
- `https://<rdapserver>/<path>/domain/<domainname>`
- `https://<rdapserver>/<path>/nameserver/<nameservername>`
- `https://<rdapserver>/<path>/entity/<handle>`
- `https://<rdapserver>/<path>/help`

An IP address string can be either IPv4 or IPv6. An IP address can have a trailing CIDR prefix number to denote a network block ("/24" for example). When a query contains spaces or other special characters they must be URL encoded ("%20" for example). The *entity* query requests information about contact details and the format dependent on the registry provider. RDAP operators may offer a help query command which provides information about using the service, policies, and getting further support. The standard query for help is simply appending "/help" to the base URL path of the RDAP application. Since RDAP servers are required to support TLS (https), it is recommended that digital forensic investigators use it, both for the confidentiality of ongoing investigations and for evidence integrity reasons. Forensic and investigative tools using RDAP should ensure the validity of TLS certificates.

If a RDAP server has the information requested, it responds with HTTP 200 together with the requested data in JSON format. If the RDAP server doesn't have the information it responds with either a redirect (HTTP 3XX) to another server that might know, or returns HTTP 404 indicating information for the request was not found. If a query cannot be completed for other reasons (policy, server failure, other errors) an appropriate HTTP error response is provided (4XX, 5XX).

The previous list shows queries for fixed known strings. RDAP also permits pattern matching for domains, nameservers, and entities. Search queries for partial strings are created using an asterisk wild-card. The "*" in a query will match zero or more characters. The syntax for pattern matching searches is shown here:

- `https://<rdapserver>/<path>/domains?<object>=<domainsearchpattern>`
- `https://<rdapserver>/<path>/nameservers?<object>=<nameserversearchpattern>`
- `https://<rdapserver>/<path>/entities?<object>=<entitynamesearchpattern>`

Further details on each of these query types are documented in Section 3.2 *Search Path Segment Specification* of RFC-7482. A draft standard is currently being developed to allow POSIX regular expressions.

2.3 RDAP response

After a properly formed HTTP query is sent, the RDAP server will reply with a JSON formatted response. The format of JSON RDAP responses is defined in RFC-7483 which specifies data types, data structures, and object classes.

The JSON standard (RFC-7159) defines various standard data types. The RDAP response standard (RFC-7483) uses additional data types for registry handles (nic-handle, registryId, etc.), IPv4 and IPv6 addresses, country codes, DNS domain names (in LDH or Unicode), time-stamps, URIs, and contact information (vCard in JSON representation).

A number of common data structures are used to construct a valid RDAP response. At the top of the response, the *rdapConformance* array indicates conformity to the RDAP specification and notes any custom extensions. Notices and Remarks arrays allow descriptive text to be provided in a response. Other data structures include links, language, public IDs, and the various RDAP object classes.

Of particular interest to digital forensic investigators are: The current status of an object, event actions associated with an object, and the roles or contact information available. These are defined in RFC-7483, maintained at IANA (`https://www.iana.org/assignments/rdap-json-values/rdap-json-values.xhtml#rdap-json-values-1`).

The *status* data structure indicates the current authoritative state of the RDAP object. The IANA registry defines a number of additional status values which are specified in RFC-8056, Extensible Provisioning Protocol (EPP) and Registration Data Access Protocol (RDAP) Status Mapping (`https://tools.ietf.org/html/rfc8056`). These are listed here:

- validated: Signifies that the data of the object instance has been found to be accurate. This type of status is usually found on entity object instances to note the validity of identifying contact information.

- renew prohibited: Renewal or reregistration of the object instance is forbidden.
- update prohibited: Updates to the object instance are forbidden.
- transfer prohibited: Transfers of the registration from one registrar to another are forbidden.
- delete prohibited: Deletion of the registration of the object instance is forbidden.
- proxy: The registration of the object instance has been performed by a third party.
- private: The information of the object instance is not designated for public consumption.
- removed: Some of the information of the object instance has not been made available and has been removed.
- obscured: Some of the information of the object instance has been altered for the purposes of not readily revealing the actual information of the object instance.
- associated: The object instance is associated with other object instances in the registry.
- active: The object instance is in use.
- inactive: The object instance is not in use.
- locked: Changes to the object instance cannot be made
- pending create: A request has been received for the creation of the object instance, but this action is not yet complete.
- pending renew: A request has been received for the renewal of the object instance, but this action is not yet complete.
- pending transfer: A request has been received for the transfer of the object instance, but this action is not yet complete.
- pending update: A request has been received for the update or modification of the object instance, but this action is not yet complete.
- pending delete: A request has been received for the deletion or removal of the object instance, but this action is not yet complete.

Event Actions are interesting for building investigative timelines, reconstructing past activity, and predicting future events (some events may be future dated). Each event has the action, a timestamp, and possibly the actor responsible for triggering the action. The event actions defined in RFC-7483 are listed here:

- registration: The object instance was initially registered.

- last changed: An action noting when the information in the object instance was last changed.
- expiration: The object instance has been removed or will be removed at a predetermined date and time from the registry.
- deletion: The object instance was removed from the registry at a point in time that was not predetermined.
- reinstantiation: The object instance was reregistered after having been removed from the registry.
- transfer: The object instance was transferred from one registrant to another.
- locked: The object instance was locked.
- unlocked: The object instance was unlocked.

The entity object class lists contact information and the associated roles. The contact information is provided in jCard format, which is a JSON representation of vCard. For digital forensic investigators, this contact information (if accurate) can play a valuable role in attribution, identifying related parties, and building actor/suspect relationship maps. The roles defined in RFC-7483 are listed here:

- registrant: The registrant of the registration
- technical: The technical contact for the registration
- administrative: The administrative contact for the registration
- abuse: Handles network abuse issues on behalf of the registrant of the registration
- billing: Handles payment and billing issues on behalf of the registrant of the registration
- registrar: Represents the authority responsible for the registration in the registry
- reseller: Represents a third party through which the registration was conducted
- sponsor: Represents a domain policy sponsor, such as an ICANN-approved sponsor
- proxy: Represents a proxy for another entity object, such as a registrant
- notifications: Designated to receive notifications about association object instances
- noc: Handles communications related to a network operations center (NOC)

The RDAP standard precisely defines each of the data structures and objects returned by a query. The JSON format allows scripts and investigative tools to directly import or query data from RDAP servers.

3 Authoritative sources and bootstrapping

An important concept in the field of digital forensics is finding authoritative sources of evidence. This can be difficult on the Internet where caching, proxying, relaying, web application front-ends, replication, and abstraction all distance the investigator from the original authoritative evidence source. Evidence is at risk of becoming stale, corrupted, modified, or partially deleted. RDAP helps facilitate the identification and use of authoritative sources. The ability to automatically redirect (via HTTP) a client to a known authority is built into the protocol. In addition, the concept of bootstrapping was developed to support this activity.

3.1 Public RDAP servers

The five Regional Internet Registries (RIRs) all provide public access RDAP servers for their areas of responsibility:

- `https://rdap.db.ripe.net/`
- `https://rdap.arin.net/registry/`
- `https://rdap.apnic.net/`
- `https://rdap.lacnic.net/rdap/`
- `https://rdap.afrinic.net/rdap/`

Local Internet Registries (LIRs), National Internet Registries (NIRs), and ccTLD (country code Top Level Domain) registrars may operate their own RDAP servers. For example, Brasil, The Czech Republic, and Argentina are already providing RDAP services for their countries:

- `https://rdap.registro.br/`
- `https://rdap.nic.cz`
- `https://rdap.nic.ar`

Domain Name Registries (DNRs) can also provide RDAP services for the registry information under their responsibility. Here are a few examples:

- Verisign provides an experimental RDAP service for the TLD and gTLD zones where it is the registry operator (including .cc, .tv, .jobs, and many others). : `https://rdap.verisignlabs.com/rdap/v1/`
- Afilias offers RDAP service for .info domains: `http://rdg.afilias.info/rdap/domain/info.info`
- Spanish registrar Dinahosting offers RDAP service: `https://rdap.dinahosting.com`

As of this writing, adoption of RDAP among domain name registrars is slow. ICANN's *RDAP Operational Profile for gTLD Registries and Registrars* (`https://www.icann.org/resources/pages/rdap-operational-profile-2016-07-26-en`) defines mandatory requirements for gTLD registries and registrars. However, there is reluctance in the domain name registrar community who object to the additional cost of implementing RDAP. In 2016 the Registry Stakeholder Group (RySG) requested that RDAP be removed from ICANN's Consistent Labeling and Display Policy (`https://www.icann.org/en/system/files/files/reconsideration-16-10-rysg-request-redacted-09aug16-en.pdf`). The discussion of how and when RDAP will be implemented in the gTLD space was further discussed at ICANN 58 Copengagen during March 2017 (`https://schedule.icann.org/event/9nnH/icann-gdd-rdap`).

3.2 IANA bootstrap files

There is no central RDAP service that provides authoritative answers to every query requested. There are many RDAP servers, each providing registration data for their own area of responsibility. This causes a chicken and egg problem of finding the right RDAP service for a particular query item. This has been partially solved through a set of bootstrap files maintained by IANA to help find authoritative RDAP servers.

IANA maintains a list of authoritative RDAP servers for Autonomous System Numbers (`https://data.iana.org/rdap/asn.json`), IPv4 addresses (`https://data.iana.org/rdap/ipv4.json`), IPv6 addresses (`https://data.iana.org/rdap/ipv6.json`), and Domain Names (`https://data.iana.org/rdap/dns.json`). These are are known as the Bootstrap Service Registry files.

The ASN and IP address bootstrap files associate ASNs and IPs with an authoritative RDAP service. The Domain Name bootstrap file (as of this writing) contains RDAP servers for .ar, .cz, and .br ccTLDs. Creating bootstrap files for domain names is a challenge due to the large number of registrars (compared to only 5 RIRs for IPs and ASNs). There is currently no single bootstrap file set for gTLDs, ccTLDs, or the newer sponsored sTLDs.

4 RDAP as a forensic evidence source

Several security considerations not found in traditional WHOIS are of interest in a digital forensic context.

The confidentiality of RDAP queries can be guaranteed through TLS, which is a requirement specified in the RFCs. This restricts knowledge of operationally sensitive query activity to the investigator making the query and the RDAP server operator. This is unlike WHOIS requests which are not protected and visible to anyone with access to the underlying network infrastructure. This is especially important where operational secrecy or confidentiality are required.

The integrity of both requests and responses can be guaranteed through two methods. The use of an encrypted TLS channel ensures the integrity of the transmitted request and response data[1]. The use of DNSSEC can also ensure the authenticity of the RDAP server's resolved IP address, and in some cases (ccTLDs and gTLDs) DNSSEC is a requirement for the operation of an RDAP service.

The well defined and standardized structure of the returned JSON data is also beneficial in a forensic context. There is less risk of error (human or machine) when interpreting results because they are standard across registry server operators. Also the use of Representational State Transfer (REST) makes it easier to create forensic tools that automate the collection of data in a reliable way, again reducing the risk of human error.

The concept of identifying and using authoritative sources is designed into the RDAP concept. This is important in a digital forensic context, as an authoritative evidence source is considered more reliable and accurate than non-authoritative sources or indirect access. For example, independently operated 3rd party web interfaces might cache requests for performance reasons, and may not have the most current results.

It is important to note that the accuracy of registration data submitted by registrants to registries is not consistently reliable. It is easy for a registrant to provide false, misleading, or inaccurate data. If a registrant fails to update changes to their information at a particular registry, it will be out of date. This was an issue in the past with WHOIS, and continues to be an issue with RDAP. The accuracy of data submitted to a registry depends on the validation processes and policies of each individual registrar.

Overall, the introduction of RDAP addresses a number of shortcomings that exist in the traditional WHOIS system. These are directly relevant to the use of such data in a forensic context.

5 RDAP tools and resources

As of this writing there are a handful of tools available for querying RDAP servers. Several of these tools are described in the following section which a forensic investigator may find useful. There are also resources available to find more information about RDAP and to follow the effort to replace the traditional WHOIS system.

5.1 RDAP open source tools

Because RDAP uses common HTTP methods, a normal web browser is able to directly query RDAP servers to fetch data. The returned RDAP output can be viewed in the browser's native output format, or with the help of browser

[1] This assumes a trusted CA certificate on the RDAP server, and proper certificate validation on the client.

plugins for better formatting (JSONView for example, a Firefox plugin found here: `https://jsonview.com/`).

When using a shell or scripting, common url fetching commands like wget or curl can be used. It is recommended to include an "Accept:" header for RDAP and JSON in the HTTP request. For example: `curl-HAccept:application/rdap+jsonhttps://SERVER/RDAP-QUERY`.

The *nicinfo* tool is written in Ruby by ARIN Engineering. It can be found here: `https://github.com/arineng/nicinfo.git`

The *ipwhois* package is written in python and combines traditional WHOIS lookups with RDAP queries into an API that fetches registry information for IPv4 and IPv6 addresses and networks. It includes a tool called `ipwhois_cli.py`. It can be found here: `https://github.com/secynic/ipWHOIS.git`

The *rdap-client* tool is written in Go and developed by the Brasilian NIC. It can be found here: `https://github.com/registrobr/rdap-client.git`

The *rdapper* client is an older tool developed while the IETF WEIRDS working group was still finalizing the RDAP standards. It can be found here: `https://github.com/jodrell/rdapper.git`

The *rdap-explorer* web application is written in python and provides a web-based GUI to perform queries. The software can be found here: `https://github.com/cdubz/rdap-explorer.git` and an online functioning version can be found here: `https://rdap-explorer.com/`. The online version publicly displays recent queries and might not be suitable for use in confidential investigations.

The *rdap* client is part of a larger software package which also includes RDAP libraries and a server implementation. It is written in Java and developed by DNS Belgium, a non-profit organization. It can be found here: `https://github.com/DNSBelgium/rdap`

5.2 Other RDAP resources

The primary resource for information about the defined RDAP protocol are the RFCs published by the IETF. These can be found at the IETF website: `https://ietf.org/rfc.html`. Several additional IETF draft documents related to RDAP are being developed as of this writing. Of interest to digital forensic investigators is the introduction of POSIX regular expressions for RDAP search queries. The draft document can be found here: `https://datatracker.ietf.org/doc/draft-fregly-regext-rdap-search-regex/`.

Two initial IETF mailing lists for discussing the development of the RDAP protocol were the *WHOIS-based Extensible Internet Registration Data Service* (`https://mailarchive.ietf.org/arch/browse/weirds/`) and *Extensible Provisioning Protocol Extensions* (`https://mailarchive.ietf.org/arch/browse/eppext/`) lists. Follow on work from these two groups was moved to the IETF *Registration Protocols Extensions list* (`https://mailarchive.ietf.org/arch/browse/regext/`).

The Internet Assigned Numbers Authority (IANA) maintains a registry of several RDAP related items. These include bootstrap information, RDAP extensions, and RDAP JSON values (`https://www.iana.org/protocols`). For policies and complaints regarding accuracy of WHOIS and RDAP registry data, the non-profit body ICANN (Internet Corporation for Assigned Names and Numbers) provides a number of resources here: `https://whois.icann.org/`.

Other relevant resources to follow are the RIRs and LIRs. All the five RIRs (ARIN, RIPE, APNIC, LACNIC, and AFRINIC) currently support RDAP and may provide additional resources. For example, ARIN has web pages describing RDAP in detail (`https://www.arin.net/resources/rdap.html`). The larger DNRs (Domain Name Registars) may also have information about their RDAP implementation. For example, Verisign Labs is actively involved in developing and promoting RDAP (`https://www.verisign.com/en_US/company-information/verisign-labs/registration-data-access-protocol/index.xhtml`).

6 RDAP query examples

Several practical examples of RDAP queries are shown here using different tools and methods. These are typical examples used by digital forensics investigators to collect evidence, gather intelligence, or find other information about various RDAP objects. For each example the full RDAP query URL is shown followed by a different tool usage and output.

6.1 Object name query examples

Queries for exact matches of object names are the simplest form of RDAP request. Examples are shown here to demonstrate finding registration information about IPv4 addresses, IPv4 networks, IPv6 addresses, and DNS domain names.

Since RDAP queries are simple HTTP requests, they can be sent using wget or curl. In this example registration information for an IP address is fetched using curl. The bootstrap files (available at IANA) are manually checked for the authoritative RDAP server, and the IP is determined to be registered with APNIC.

```
$ curl https://rdap.apnic.net/ip/61.135.157.156
{
  "handle" : "61.135.0.0 - 61.135.255.255",
  "startAddress" : "61.135.0.0",
  "endAddress" : "61.135.255.255",
```

```
  "ipVersion" : "v4",
  "name" : "UNICOM-BJ",
  "type" : "ALLOCATED PORTABLE",
  "country" : "CN",
  "objectClassName" : "ip network",
  "entities" : [ {
    "handle" : "CH1302-AP",
    "vcardArray" : [ "vcard", [ [ "version", { }, "text", "4.0" ], [ "fn", { }, "text", "C
      "label" : "No.21,Jin-Rong Street\\nBeijing,100033\\nP.R.China"
...
```

The curl command can specify HTTP Accept headers for receiving the "rdap+json" Content-Type using `curl-HAccept:application/rdap+json` in the request.

The next example demonstrates querying for a network range using nicinfo. The bootstrapping is done automatically to find the authoritative server. The query url `https://rdap.db.ripe.net/ip/129.67.0.0/16` is sent to the RIPE RDAP service. The output from the nicinfo command is as follows:

```
nicinfo 129.67.0.0/16
# NicInfo v.1.1.1 EXPERIMNETAL

Excessive Notices
-----------------
Response contains excessive notices.
Use the "-V" or "--data extra" options to see them.

# Query type is IP4CIDR. Result type is IP.
[ RESPONSE DATA ]

           [ IP NETWORK ]
                  Handle:  129.67.0.0 - 129.67.255.255
           Start Address:  129.67.0.0/32
             End Address:  129.67.255.255/32
                   CIDRs:  129.67.0.0/16
              IP Version:  v4
                 Country:  GB
                    Type:  LEGACY
            Last Changed:  Wed, 25 Nov 2015 12:49:37 -0000
                 Remarks:  Oxford University

# Use "nicinfo 1=" to show 129.67.0.0 - 129.67.255.255
# Use "nicinfo https://rdap.db.ripe.net/ip/129.67.0.0/16" to directly query this resource
# Use "nicinfo -h" for help.
```

The nicinfo command has a number of flags to fetch further information and individual data types.

A web based tool call rdap-explorer can be used to make RDAP queries. In this example registry information about an IPv6 address is fetched. The query url `https://rdap.apnic.net/ip/2001:dc0:2001:11::194` is the raw RDAP request for IPv6. The rdap-explorer user interface looks like this:

Be careful using online tools from third parties. The accuracy and integrity of the data is not guaranteed, and confidential investigative information might be leaked to the provider or even the public (for example, if there is a "Recent Queries" list shown). Some web based tools might perform RDAP queries in a backend application without making it obvious to the user that RDAP is being

used.

The next example shows a query using the Firefox browser with the JSONView browser extension. The RDAP query string (`https://rdap.nic.cz/domain/czech.cz`) is placed directly in the browser and then displayed as follows:

The JSONView plugin makes it easy to collapse and expand the full RDAP response. The bootstrap to find the authoritative RDAP server must be done manually.

Similar queries can be made for nameservers, autonomous system numbers, and any extensions supported by the registry operator.

6.2 Pattern matching query examples

Pattern matching search queries have a slightly different syntax that follow a typical web application request style using a "?" followed by variable with the query string pattern. The formats for search pattern queries are described in Section 3 of RFC-7482 and are listed here:

- `domains?name=<domainsearchpattern>`

- `domains?nsLdhName=<domainsearchpattern>`
- `domains?nsIp=<domainsearchpattern>`
- `nameservers?name=<nameserversearchpattern>`
- `nameservers?ip=<nameserversearchpattern>`
- `entities?fn=<entitynamesearchpattern>`
- `entities?handle=<entityhandlesearchpattern>`

The search pattern my contain a "*" to indicate zero or more matching characters. A draft RFC has been created to include regular expressions in the future.

For example, the following query will search for the contact name starting with "John Doe" (with a space):

`https://rdap.db.ripe.net/entities?fn=John%20Doe*`

Registry operators may not implement all of the pattern searches and will return an HTTP 422 (Unprocessable Entity) error message if they are unable to serve the request. Use the "/help" query for a registry to find more information about their query parameters.

6.3 International character support

The original WHOIS protocol supported plain US-ASCII. The RDAP standard includes internationalization support. Queries for RDAP servers follow the same international encoding standards as any other encoded HTTP URL request. The default encoding for RDAP responses is UTF-8 JSON. Here is an example of an internationalized domain name handled in a response:


```
"unicodeName": "cnnic.cn",
"nameServers": [
    {
        "handle": "ns-1",
        "ldhName": "xn--1-dr6av31f.xn--0zwm56d.xn--fiqs8s",
        "unicodeName": "主机1.测试.中国",
        "status": [
            "delete prohibited"
        ],
```


More information about internationalization can be found in Section 9 of RFC-7480, Section 6 of RCF-7482, and Section 12 of RFC-7483.

7 Conclusion

Knowledge of the RDAP protocol and infrastructure is important for digital forensic investigators. RDAP is intended to replace the traditional WHOIS system with a modern design, and offers a number of features which are useful in a forensic and investigative context. Of particular interest are the improved security of the protocol, the ease of automation and tool integration, and the ability to ensure use of authoritative data sources.

As of this writing, bootstrap files for the IPv4 and IPv6 address space and ASNs are pointing to authoritative RDAP servers among the five RIRs. A small number of domain registrars are listed in the bootstrap files. It not implemented widely enough in the domain registration space to be useful for investigators yet, but policy discussions and adoption are ongoing. It is not clear when (or if) it will completely overtake traditional WHOIS services. It is likely that WHOIS and RDAP will co-exist for some time in the future. As the use of RDAP increases it will provide a more structured and reliable data source for investigations and digital forensic evidence.

References

[1] WHOIS Protocol Specification, IETF, September 2004

[2] Domain Name Forensics: A Systematic Approach to Investigating an Internet Presence, Bruce Nikkel, Digital Investigation Vol 1, No 4 (November 2004),Elsevier doi:10.1016/j.diin.2004.10.001

[3] Referral Whois Protocol (RWhois) V1.5, IETF, June 1997

[4] A Survey of Advanced Usages of X.500, IETF, July 1993

[5] Cross Registry Internet Service Protocol (CRISP) Requirements, IETF, February 2004

[6] IRIS: The Internet Registry Information Service (IRIS) Core Protocol, IETF, January 2005

[7] HTTP Usage in the Registration Data Access Protocol (RDAP), IETF, March 2015

[8] Security Services for the Registration Data Access Protocol (RDAP), IETF, March 2015

[9] Registration Data Access Protocol (RDAP) Query Format, IETF, March 2015

[10] JSON Responses for the Registration Data Access Protocol (RDAP), IETF, March 2015

[11] Finding the Authoritative Registration Data (RDAP) Service, IETF, March 2015

[12] Inventory and Analysis of WHOIS Registration Objects, IETF, March 2015