Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: August 14, 2014
S. Hollenbeck
Verisign Labs
N. Kong
CNNIC
February 10, 2014

Security Services for the Registration Data Access Protocol
draft-ietf-weirds-rdap-sec-06

Abstract

The Registration Data Access Protocol (RDAP) provides "RESTful" web services to retrieve registration metadata from domain name and regional internet registries. This document describes information security services including authentication, authorization, availability, data confidentiality, and data integrity for RDAP.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The Registration Data Access Protocol (RDAP) is specified in multiple documents, including "Registration Data Access Protocol Lookup Format" [I-D.ietf-weirds-rdap-query], "JSON Responses for the Registration Data Access Protocol (RDAP)" [I-D.ietf-weirds-json-response], and "HTTP usage in the Registration Data Access Protocol (RDAP)" [I-D.ietf-weirds-using-http].

One goal of RDAP is to provide security services that do not exist in the WHOIS [RFC3912] protocol, including authentication, authorization, availability, data confidentiality, and data integrity. This document describes how each of these services is achieved by RDAP. Where applicable, informational references to requirements for a WHOIS replacement service [RFC3707] are noted.

2. Conventions Used in This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2.1. Acronyms and Abbreviations

DNR: Domain Name Registry

HTTP: Hypertext Transfer Protocol

JSON: JavaScript Object Notation

RDAP: Registration Data Access Protocol

RIR: Regional Internet Registry

TLS: Transport Layer Security

3. Information Security Services and RDAP

RDAP itself does not include native security services. Instead, RDAP relies on features that are available in other protocol layers to provide needed security services including authentication, authorization, availability, data confidentiality, and data integrity. A description of each of these security services can be found in "Internet Security Glossary, Version 2" [RFC4949]. No requirements have been identified for other security services.

3.1. Authentication

WHOIS does not provide features to identify and authenticate clients. As noted in section 3.1.4.2 of "Cross Registry Internet Service Protocol (CRISP) Requirements" [RFC3707], there is utility in allowing server operators to offer "varying degrees of access depending on policy and need". Clients have to be identified and authenticated to provide that utility.

RDAP's authentication framework needs to accommodate anonymous access as well as verification of identities using a range of authentication methods and credential services. To that end, RDAP clients and servers MUST implement the authentication framework specified in "HTTP Authentication: Basic and Digest Access Authentication" [RFC2617]. The "basic" scheme can be used to send a client's user name and password to a server in plaintext, based64-encoded form. The "digest" scheme can be used to authenticate a client without exposing the client's plaintext password. If the "basic" scheme is used, HTTP Over TLS [RFC2818] MUST be used to protect the client's credentials from disclosure while in transit (see Section 3.4).

Servers MUST support either Basic or Digest authentication; they are not required to support both. Clients MUST support both to interoperate with servers that support one or the other.

The Transport Layer Security Protocol [RFC5246] includes an optional feature to identify and authenticate clients who possess and present a valid X.509 digital certificate [RFC5280]. Support for this feature is OPTIONAL.

RDAP does not impose any unique server authentication requirements. The server authentication provided by TLS fully addresses the needs of RDAP. In general, transports for RDAP must either provide a TLS-protected transport (e.g., HTTPS) or a mechanism that provides an equivalent level of server authentication.

Work on HTTP authentication methods continues. RDAP ought to be agile enough to support additional methods as they are defined.

3.1.1. Federated Authentication

The traditional client-server authentication model requires clients to maintain distinct credentials for every RDAP server. This situation can become unwieldy as the number of RDAP servers increases. Federated authentication mechanisms allow clients to use one credential to access multiple RDAP servers and reduce client credential management complexity. RDAP MAY include a federated authentication mechanism that permits a client to access multiple RDAP servers in the same federation with one credential.

Federated authentication mechanisms used by RDAP are OPTIONAL. If used, they MUST be fully supported by HTTP. OAuth, OpenID, and CA-based mechanisms are three possible approaches to provide federated authentication.

The OAuth authorization framework [RFC6749] describes a method for users to access protected web resources without having to hand out their credentials. Instead, clients are issued access tokens by authorization servers with the permission of the resource owners. Using OAuth, multiple RDAP servers can form a federation and the clients can access any server in the same federation by providing one credential registered in any server in that federation. The OAuth authorization framework is designed for use with HTTP and thus can be used with RDAP.

OpenID [OpenID] is a decentralized single sign-on authentication system that allows users to log in at multiple web sites with one ID instead of having to create multiple unique accounts. An end user can freely choose which OpenID provider to use, and can preserve their Identifier if they switch OpenID providers.

Note that OAuth and OpenID do not consistently require data confidentiality services to protect interactions between providers

and consumers. HTTP Over TLS [RFC2818] can be used as needed to provide protection against man-in-the-middle attacks.

The Transport Layer Security Protocol [RFC5246], Section 7.4.6, describes the specification of a client certificate. Clients who possess and present a valid X.509 digital certificate, issued by an entity called a "Certification Authority" (CA), could be identified and authenticated by a server who trusts the corresponding CA. A certificate authentication method can be used to achieve federated authentication in which multiple RDAP servers all trust the same CAs and then any client with a certificate issued by a trusted CA can access any RDAP server in the federation. This certificate-based mechanism is supported by HTTPS and can be used with RDAP.

3.2. Authorization

WHOIS does not provide services to grant different levels of access to clients based on a client's authenticated identity. As noted in section 3.1.4.2 of "Cross Registry Internet Service Protocol (CRISP) Requirements" [RFC3707], there is utility in allowing server operators to offer "varying degrees of access depending on policy and need". Access control decisions can be made once a client's identity has been established and authenticated (see Section 3.1).

Server operators SHOULD offer varying degrees of access depending on policy and need in conjunction with the authentication methods described in Section 3.1. If such varying degrees of access are supported, an RDAP server MUST provide granular access controls (that is, on a per registration data object basis) in order to implement authorization policies. Some examples:

- Clients will be allowed access only to data for which they have a relationship.

- Unauthenticated or anonymous access status may not yield any contact information.

- Full access may be granted to a special group of authenticated clients.

The type of access allowed by a server will most likely vary from one operator to the next.

3.3. Availability

An RDAP service has to be available to be useful. There are no RDAP-unique requirements to provide availability, but as a general security consideration a service operator needs to be aware of the

issues associated with denial of service. A thorough reading of "Internet Denial-of-Service Considerations" [RFC4732] is advised.

An RDAP service MAY use a throttling mechanism to limit the number of queries that a single client can send in a given period of time. If used, the server SHOULD return a 429 response code as described in "Additional HTTP Status Codes" [RFC6585]. A client that receives a 429 response SHOULD decrease its query rate, and honor the Retry-After header field if one is present. Note that this is not a defense against denial-of-service attacks, since a malicious client could ignore the code and continue to send queries at a high rate. A server might use another response code if it did not wish to reveal to a client that rate limiting is the reason for the denial of a reply.

3.4. Data Confidentiality

WHOIS does not provide the ability to protect data from inadvertent disclosure while in transit. Web services such as RDAP commonly use HTTP Over TLS [RFC2818] to provide that protection by encrypting all traffic sent on the connection between client and server. It is also possible to encrypt discrete objects (such as command path segments and JSON-encoded response objects) at one endpoint, send them to the other endpoint via an unprotected transport protocol, and decrypt the object on receipt. Encryption algorithms as described in "Internet Security Glossary, Version 2" [RFC4949] are commonly used to provide data confidentiality at the object level.

There are no current requirements for object-level data confidentiality using encryption. Support for this feature could be added to RDAP in the future.

As noted in Section 3.1, the HTTP "basic" authentication scheme can be used to authenticate a client. When this scheme is used, HTTP Over TLS MUST be used to protect the client's credentials from disclosure while in transit. If the policy of the server operator requires encryption to protect client-server data exchanges (such as to protect non-public data that can not be accessed without client identification and authentication), HTTP Over TLS MUST be used to protect those exchanges.

3.5. Data Integrity

WHOIS does not provide the ability to protect data from modification while in transit. Web services such as RDAP commonly use HTTP Over TLS [RFC2818] to provide that protection by using a keyed Message Authentication Code (MAC) to detect modifications. It is also possible to sign discrete objects (such as command path segments and

JSON-encoded response objects) at one endpoint, send them to the other endpoint via a transport protocol, and validate the signature of the object on receipt. Digital signature algorithms as described in "Internet Security Glossary, Version 2" [RFC4949] are commonly used to provide data integrity at the object level.

There are no current requirements for object-level data integrity using digital signatures. Support for this feature could be added to RDAP in the future.

The most specific need for this service is to provide assurance that HTTP 30x redirection hints [RFC2616] and response elements returned from the server are not modified while in transit. If the policy of the server operator requires message integrity for client-server data exchanges, HTTP Over TLS MUST be used to protect those exchanges.

4. IANA Considerations

This document does not specify any IANA actions. This section can be removed if this document is published as an RFC.

5. Security Considerations

One of the goals of RDAP is to provide security services that do not exist in the WHOIS protocol. This document describes the security services provided by RDAP and associated protocol layers, including authentication, authorization, availability, data confidentiality, and data integrity. Non-repudiation services were also considered and ultimately rejected due to a lack of requirements. There are, however, currently-deployed WHOIS servers that can return signed responses that provide non-repudiation with proof of origin. RDAP might need to be extended to provide this service in the future.

As an HTTP-based protocol RDAP is susceptible to code injection attacks. Code injection refers to adding code into a computer system or program to alter the course of execution. There are many types of code injection, including SQL injection, dynamic variable or function injection, include file injection, shell injection, and HTML-script injection among others. Data confidentiality and integrity services provide a measure of defense against man-in-the-middle injection attacks, but vulnerabilities in both client-side and server-side software make it possible for injection attacks to succeed. Consistently checking and validating server credentials can help detect man-in-the-middle attacks.

As noted in Section 3.1.1, digital certificates can be used to implement federated authentication. There is a risk of too-promiscuous, or even rogue, CAs being included in the list of

acceptable CAs that the TLS server sends the client as part of the TLS client-authentication handshake and lending the appearance of trust to certificates signed by those CAs. Periodic monitoring of the list of CAs that RDAP servers trust for client authentication can help reduce this risk.

The Transport Layer Security Protocol [RFC5246] includes a null cipher suite that does not encrypt data and thus does not provide data confidentiality. This option must not be used when data confidentiality services are needed.

Data integrity services are sometimes mistakenly associated with directory service operational policy requirements focused on data accuracy. "Accuracy" refers to the truthful association of data elements (such as names, addresses, and telephone numbers) in the context of a particular directory object (such as a domain name). Accuracy requirements are out of scope for this protocol.

Additional security considerations are described in the specifications for HTTP [RFC2616], HTTP basic and digest access authentication [RFC2617], HTTP Over TLS [RFC2818], and additional HTTP status codes [RFC6585]. Security considerations for federated authentication systems can be found in the OAuth [RFC6749] and OpenID [OpenID] specifications.

6. Acknowledgements

The authors would like to acknowledge the following individuals for their contributions to this document: Richard Barnes, Marc Blanchet, Ernie Dainow, Spencer Dawkins, Jean-Philippe Dionne, Byron Ellacott, Stephen Farrell, Tony Hansen, Peter Koch, Murray Kucherawy, Barry Leiba, Andrew Newton, and Linlin Zhou.

7. References

7.1. Normative References

[I-D.ietf-weirds-json-response]
Newton, A. and S. Hollenbeck, "JSON Responses for the Registration Data Access Protocol (RDAP)", draft-ietf-weirds-json-response-06 (work in progress), October 2013.

[I-D.ietf-weirds-rdap-query]
Newton, A. and S. Hollenbeck, "Registration Data Access Protocol Query Format", draft-ietf-weirds-rdap-query-10 (work in progress), February 2014.

[I-D.ietf-weirds-using-http]
Newton, A., Ellacott, B., and N. Kong, "HTTP usage in the Registration Data Access Protocol (RDAP)", draft-ietf-weirds-using-http-08 (work in progress), February 2014.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

[RFC2617] Franks, J., Hallam-Baker, P., Hostetler, J., Lawrence, S., Leach, P., Luotonen, A., and L. Stewart, "HTTP Authentication: Basic and Digest Access Authentication", RFC 2617, June 1999.

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000.

[RFC6585] Nottingham, M. and R. Fielding, "Additional HTTP Status Codes", RFC 6585, April 2012.

7.2. Informative References

[OpenID] OpenID Foundation, "OpenID Authentication 2.0 - Final", December 2007, <http://specs.openid.net/auth/2.0>.

[RFC3707] Newton, A., "Cross Registry Internet Service Protocol (CRISP) Requirements", RFC 3707, February 2004.

[RFC3912] Daigle, L., "WHOIS Protocol Specification", RFC 3912, September 2004.

[RFC4732] Handley, M., Rescorla, E., and IAB, "Internet Denial-of-Service Considerations", RFC 4732, December 2006.

[RFC4949] Shirey, R., "Internet Security Glossary, Version 2", RFC 4949, August 2007.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, May 2008.

[RFC6749] Hardt, D., "The OAuth 2.0 Authorization Framework", RFC 6749, October 2012.

Appendix A. Change Log

Initial -00: Adopted as working group document.

-01: Extensive text additions and revisions based on in-room discussion at IETF-85. Sections for data integrity and non-repudiation have been removed due to a lack of requirements, but both topics are now addressed in the Security Considerations section.

-02: Fixed document names in the Introduction. Modified text in Section 3.1.1 to clarify requirement. Added text to Section 3.3 to describe rate limiting. Added new data integrity section. Updated security considerations to describe injection attacks.

-03: Extensive updates to address WG last call comments: rewrote introduction, removed references to draft documents, changed "HTML" to "HTTP" in Section 5, eliminated upper case words that could be misunderstood to be normative guidance, rewrote Section 3.4 and Section 3.5.

-04: Address AD evaluation comments: In Section 3.1 change "RDAP MUST include an authentication framework that can accommodate" to "RDAP's authentication framework needs to accommodate"; in Section 3.2 change "RDAP MUST include an authorization framework that is capable of providing granular (per registration data object) access controls according to the policies of the operator" to "An RDAP server MUST provide granular access controls (that is, on a per registration data object basis) in order to implement authorization policies"; move RFCs 4732, 5280, and 6749 from normative to informative subsection.

-05: Address IETF last call comments: Added text to Section 3.1.1 to recommend the use of HTTP over TLS. Modified Section 3.2 to clarify granular access control text. Added additional Security Considerations. Made references to RFC 5246 and OpenID informative. Minor typo fixes.

-06: Keepalive refresh. No content updates.

Authors' Addresses

Scott Hollenbeck
Verisign Labs
12061 Bluemont Way
Reston, VA 20190
US

Email: shollenbeck@verisign.com
URI: http://www.verisignlabs.com/

Ning Kong
China Internet Network Information Center
4 South 4th Street, Zhongguancun, Haidian District
Beijing 100190
China

Phone: +86 10 5881 3147
Email: nkong@cnnic.cn