RDAP Test Suite

Marc Blanchet
Viagenie
marc.blanchet@viagenie.ca

ROW#6, Madrid
2017-05-12

Introduction

- Test suite to be applied against an RDAP server implementation
- Implemented during the development of the RDAP specs (of IETF weirds wg)
  - and continuing to be enhanced
- Supports and tests:
  - RDAP RFCs
  - IP address, AS numbers and Domain names
  - and various other objects: entities, nameservers, ...
- Mix of:
  - RFC specifications testing
  - “torture” tests
- Verifies the container (i.e. RDAP RFCs), not the actual content
- comprehensive:
  - ~150 test cases
  - reports show the result for each test case

RFCs

- 7480 HTTP Usage in the Registration Data Access Protocol (RDAP)
- 7482 Registration Data Access Protocol (RDAP) Query Format.
- 7483 JSON Responses for the Registration Data Access Protocol (RDAP)

Used by

- Domain name registries
- Address registries
- Some (we haven't noted everybody):

Server Profile

- supplied by the RDAP server implementer
- a list of:
  - connection info of the target server
    - test suite is conducted from our test server to the target server over Internet
    - we could provide you with the source IP in case you want to filter
  - supported features:
    - address, asn, domain
    - ...
  - test data:
    - existing records
    - non-existing records

Address Registry Profile example


```
cfg["http_host"] = "rdap.myaddressregistry.net"
cfg["base_path"] = ""
cfg["req_existant_record"] = "/autnum/3"
cfg["req_inexistant_record"] = "/entity/ORG-ABC-TEST"
cfg["req_redirect_301_record"] = "/autnum/1234"
cfg["req_unknown_path_segment"] = "/unknown/example.com"
cfg["req_valid_ipv4_addresses"] = [ "192.123.123.123" ]
cfg["req_valid_ipv6_addresses"] = [ "2001:2002:2003:2004:2005:2006:2007:2008" ]
cfg["req_existing_autnum_2byte"] = '4567'
cfg["req_unknown_autnum_2byte"] = "64001"
cfg["req_existing_autnum_4byte"] = "1888888"
cfg["req_unknown_autnum_4byte"] = "66001"
cfg["support_domain"] = False
cfg["req_existing_ldh_ns"]   = "ns.myaddressregistry.net"
cfg["req_existing_ulabel_idn_ns"] = None
cfg["req_existing_alabel_idn_ns"] = None
cfg["req_existing_entities"] = ["AB-MYADR", "ORG-ABC-MYADR", "A123-MYADR"]
cfg["req_valid_nonexisting_entity"] = "HHHBBTTT-NHHHJJJI"
```

Name Registry Profile example


```
    cfg["http_host"] = "rdap.nic.mytld"
    cfg["base_path"] = "/rdap"
    cfg["req_per_sec"] = 1
    cfg["req_existing_record"] = "/domain/anexistingdomain.mytld"
    cfg["req_valid_nonexisting_record"] =
"/domain/xnxnhxaajaoebldbselhkqsqmapxidccaaahjrgk3chhdip9bclcgddbb4ooioat.my
tld"
    cfg["support_ip"] = False
    cfg["support_asn"] = False
    cfg["req_existing_ldh_fqdn"]   = "cafe.mytld"
    cfg["req_existing_ulabel_idn"] = "café.mytld"
    cfg["req_existing_alabel_idn"] = "xn--caf-dma.mytld"
    cfg["req_valid_nonexisting_ulabel_idn"] =
"σειράτάξησυπουργείωνσύνθεσηυπουργικούσυμβουλίουουουο.mytld"
    cfg["req_valid_nonexisting_alabel_idn"] = "xn--
hxaajaoebldbselhkqsqmapxidccaaahjrgk3chhdip9bclcgddbb4ooioa.mytld"
...

    cfg["req_existing_ldh_ns"] = "ns1.mytld"
    cfg["req_existing_ip_ns"]  = "192.168.1.2"
    cfg["req_existing_entities"] = ["C123-MYTLD"]
```

Reports

- user triggers the run of the test suite
- receives report by email
- consists of the report itself in XML
  - and an XSLT to be viewed in a browser
- ~150 test cases detailed results: loooong report

Example Output

test case | result | query | http code | response body

ts_query_3_1_2_3	OK	GET /autnum/64001	404	{ "errorCode "rdapConfor WHOIS Dat "http://testw "type" : "text
ts_query_3_1_2_4	OK	GET /autnum/4199999999	404	{ "errorCode "rdapConfor WHOIS Dat "http://testw "http://www.
ts_query_3_1_2_5	OK	GET /autnum/.4608	400	{ "errorCode malformed s "This is the "http://testrd "type" : "text
ts_query_3_1_2_6	OK	GET /autnum/04608	400	{ "errorCode malformed s "This is the "http://testrd "type" : "text
ts_query_3_2_2_1	FAIL	GET /nameservers?ip=2001:42d0::200:2:1	400	{ "errorCode "rdapConfor WHOIS Data "http://testrda "http://www.a
ts_query_3_1_3_4A	OK	GET /rdap/domain/xn--hxaajaoebldbselhkqsqmapxidccaaahjrgk3chhdip9bclcgddbb4ooioa		

Results Interpretation

- We do not claim to have the truth in reading the RFC specifications
- However, over time, we have been discussing with the working group, RFC co-authors and implementors and improved the test suite consequently. We could claim to be the most comprehensive (and only one?) RDAP “conformance” test suite.
- The torture tests could be considered “extra” and you may disagree on our results (for example, some implementations just prefer ignoring “badly” formed requests for example)
- Use the results as your own needs.

I'm Interested
What Should I Do?

- contact us: support@viagenie.ca
- we will create you an account
- we will send you a template profile
- you will fill out the profile
- get your server ready
- you activate the tests (full or specific)
- you receive the results
- you are happy (or not ;-))
- all free

Summary

- RDAP test suite
- for RDAP servers implementations
- for address, asn and domain
- has RDAP RFC “compliance” and tortured test cases
- detailed and comprehensive tests and reports
- questions?

marc.blanchet@viagenie.ca