CentralNic Registry

How to implement a thin registry with eligibility requirements

Gavin Brown, Chief Innovation Officer
<gavin.brown@centralnic.com>

ROW #8, Bangkok, May 2019

Background

- The need to validate registrant eligibility is often given as the principle purpose for maintaining thick registries
- COM and NET are just fine being thin, so why do other TLDs need to be thick?
- I contend that it is possible to implement eligibility validation in a thick registry
- Therefore, according to the principle of data minimisation, all gTLDs should be thin!

Note

- This proposal is superficially similar to the Verification Code Extension
- The key difference is that the Verification Code Extension relies on an external third party
- This proposal puts the validation process in the hands of the registry operator
- Also, the Verification Code Extension does not provide a way for a registry to re-certify a registrant

The Inspiration

CentralNic Registry

The Inspiration

- At the immigration desk, the officer checks your passport to make sure you are who you say you are
- If you pass the checks, you're allowed in
- Usually, the officer doesn't take a copy of your passport or keep it while you're in the country*
- You might be asked to present your passport while you are visiting

*yes, I know this does sometimes happen

The Proposal

- The registry needs to verify your eligibility, but do they really need to keep your contact information? Why not just validate and then forgetit, like the immigration officer?

- I propose "inline contacts" which extend `<create>` and other transform commands, but whose contents are not persistently stored at the registry

- The registry reviews and validates the info during the `<create>` process, but then discards it once the domain is registered

```
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <create>
      <domain:create
        xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>example.com</domain:name>
        <domain:period unit="y">2</domain:period>
        <domain:authInfo>
          <domain:pw>2fooBAR</domain:pw>
        </domain:authInfo>
      </domain:create>
    </create>
    <extension>
      <ic:create
        xmlns="urn:ietf:params:xml:ns:epp:ic-1.0">
        <ic:registrant>
          <ic:postalInfo type="int">
            <ic:name>John Doe</ic:name>
            <ic:org>Example Inc.</ic:org>
            <ic:addr>
              <ic:street>123 Example Dr.</ic:street>
              <ic:street>Suite 100</ic:street>
              <ic:city>Dulles</ic:city>
              <ic:sp>VA</ic:sp>
              <ic:pc>20166-6503</ic:pc>
              <ic:cc>US</ic:cc>
            </ic:addr>
          </ic:postalInfo>
          <ic:voice x="1234">+1.7035555555</ic:voice>
          <ic:email>jdoe@example.com</ic:email>
        </ic:registrant>
      </ic:create>
    </extension>
    <clTRID>ABC-12345</clTRID>
  </command>
</epp>
```

CentralNic Registry

Registrant Re-certification

- The registry can require the registrar to re-submit registrant info as part of `<renew>` or `<transfer>` commands (or any other transform command)

```
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <renew>
      <domain:renew
        xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>example.com</domain:name>
        <domain:period unit="y">2</domain:period>
      </domain:renew>
    </create>
    <extension>
      <ic:renew
        xmlns="urn:ietf:params:xml:ns:epp:ic-1.0">
        <ic:registrant>
          <ic:postalInfo type="int">
            <ic:name>John Doe</ic:name>
            <ic:org>Example Inc.</ic:org>
            <ic:addr>
              <ic:street>123 Example Dr.</ic:street>
              <ic:street>Suite 100</ic:street>
              <ic:city>Dulles</ic:city>
              <ic:sp>VA</ic:sp>
              <ic:pc>20166-6503</ic:pc>
              <ic:cc>US</ic:cc>
            </ic:addr>
          </ic:postalInfo>
          <ic:voice x="1234">+1.7035555555</ic:voice>
          <ic:email>jdoe@example.com</ic:email>
        </ic:registrant>
      </ic:renew>
    </extension>
    <clTRID>ABC-12345</clTRID>
  </command>
</epp>
```

CentralNic Registry

Audit Mechanism

- Registry sends registrar a `<poll>` message informing them of an audit for a domain
- Registrar sends a no-op `<update>` command including registrant info
- The registry reviews and validates the info and then discards it
- If registrar does not respond within a certain time frame, registry adds `serverHold`

```
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <update>
      <domain:update
        xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>example.com</domain:name>
        <chg/>
      </domain:update>
    </update>
    <extension>
      <ic:update
        xmlns="urn:ietf:params:xml:ns:epp:ic-1.0">
        <ic:registrant>
          <ic:postalInfo type="int">
            <ic:name>John Doe</ic:name>
            <ic:org>Example Inc.</ic:org>
            <ic:addr>
              <ic:street>123 Example Dr.</ic:street>
              <ic:street>Suite 100</ic:street>
              <ic:city>Dulles</ic:city>
              <ic:sp>VA</ic:sp>
              <ic:pc>20166-6503</ic:pc>
              <ic:cc>US</ic:cc>
            </ic:addr>
          </ic:postalInfo>
          <ic:voice x="1234">+1.7035555555</ic:voice>
          <ic:email>jdoe@example.com</ic:email>
        </ic:registrant>
      </ic:update>
    </extension>
    <clTRID>ABC-12345</clTRID>
  </command>
</epp>
```

CentralNic Registry

Conclusions

- Registries do not need to store registrant data in order to be able to validate it; EPP can provide a straightforward mechanism for validating and auditing data without storing it

- EPDP participants did not seem aware that this sort of model was possible in principle

- If a thin model can meet all the requirements for gTLD registry operations (which it obviously can), then the thick model cannot be compliant with the principle of data minimisation

CentralNic Registry

I fully retract my CircleID article from 2007!

.COM and .NET: Thick Or Thin?

By **Gavin Brown**

Apr 02, 2007 9:33 AM PDT | Comments: 4 | Views: 29,478 Comment | Print

The fallout from the failure of RegisterFly has been largely addressed as an issue of regulation and enforcement. ICANN needs to enable registrants to transfer their domain names away from RegisterFly, or to "bulk transfer" all of RegisterFly's sponsored domain names to another registrar. However, RegisterFly has control of all the customer data so it's impossible to match registrant to domain name, in order to release the all-important AuthInfo code.

The Registrar Accreditation Agreement (RAA) requires that registrars place this customer data into an escrow system with ICANN, so that in the event of a business failure at a

https://www.circleid.com/posts/com_net_thick_or_thin/