Login Security Extension

James F. Gould
jgould@verisign.com
IETF-102 REGEXT Working Group

Introduction

- Problems
  - RFC 5730 supports passwords with a maximum length of just 16 characters
  - EPP does not support the server providing login security warnings to the client (e.g. password expiry, cert expiry, insecure ciphers, etc.)
  - EPP does not support the client providing user agent information to the server
- Proposing: Login Security Extension (draft-gould-regext-login-security)
  - https://tools.ietf.org/html/draft-gould-regext-login-security
  - Addresses the problems with an extension to the EPP Login command and response

Allow Passwords Longer Than 16 Characters

- Extend the Login Command
    - Enable overriding the current password (<pw>) or new password (<newPw>)
    - Use of *[LOGIN-SECURITY]* constant value to indicate override in extension
        - RFC 5730 requires a 6 to 16 character value for the <pw> or <newPw> element
        - Using the 16 character constant value makes the override explicit by the client
    - Client may continue to use the RFC 5730 elements if the password is 6 to 16 characters long
    - Login Security Extension must be used if the password is greater than 16 characters long
- <loginSec:pw> and <loginSec:newPw>
    - Uses XML schema "token" type with a minimum length of 6 and no maximum

Long Password Example

```
…
    <login>
      <clID>ClientX</clID>
      <pw>[LOGIN-SECURITY]</pw>
      <newPW>[LOGIN-SECURITY]</newPW>
…
    </login>
    <extension>
      <loginSec:loginSec xmlns:loginSec="urn:ietf:params:xml:ns:loginSec-0.1">
        <loginSec:pw>this is a long password</loginSec:pw>
        <loginSec:newPW>new password that is still long</loginSec:newPW>
      </loginSec:loginSec>
    </extension>
…
```

Server: to Provide Login Security Warnings and Errors

- Extend the Login Response
- Support for a list of security events
  - A server may identify many security events for a session
  - Examples include expiring password, expiring certificate, insecure ciphers, etc.
- Extension added to the response only if
  - Client supports the Security Event extension based on the login services
  - There is at least one login security event

Login Security Event Attributes

- “type” – Extensible enumerated list of event types
    - “password”
    - “certificate”
    - “cipher”
    - “tlsProtocol”
    - “newPw”
    - “stat”
    - “custom”
- “name” – Optional name of “custom” or “stat” type event
- “level” – “warning” or “error” (for any event type)
- “exDate” – Optional expiration date for an expiry event (e.g., “password”, “certificate”)
- “value” – Optional value of a “stat” type event
- “duration” – Optional duration of a “stat” type event
- “lang” – Optional language of the event description with “en” default value
- Description – Human-readable description of the event element

Password Expiry Warning Event Example

```
…
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <extension>
      <loginSec:loginSecData xmlns:loginSec="urn:ietf:params:xml:ns:loginSec-0.1">
        <loginSec:event
          type="password"
          level="warning”
          exDate="2018-04-01T22:00:00.0Z">
          Password expiration soon
        </loginSec:event>
      </loginSec:loginSecData>
    </extension>
…
```

Password Expiry Error Event Example

```
…
    <result code="2200">
      <msg>Authentication error</msg>
    </result>
    <extension>
      <loginSec:loginSecData xmlns:loginSec="urn:ietf:params:xml:ns:loginSec-0.1">
        <loginSec:event
          type="password"
          level=”error”
          exDate="2018-03-26T22:00:00.0Z">
          Password has expired
        </loginSec:event>
      </loginSec:loginSecData>
    </extension>
…
```

Client: User Agent Information

- Extend the Login Command
  - Provide option for client to provide client agent information to the server
  - Provides the client software and platform used
  - Server can identify functional and security constraints, current security issues, and potential future functional and security issues for the client

Client User Agent Example

```
…
    <login>
      <clID>ClientX</clID>
      <pw>[LOGIN-SECURITY]</pw>
…
    </login>
    <extension>
      <loginSec:loginSec xmlns:loginSec="urn:ietf:params:xml:ns:loginSec-0.1">
        <loginSec:userAgent>EPP SDK/1.0.0
          (Java SE 1.8.0_131; Macintosh; Intel Mac OS X 10_13)
        </loginSec:userAgent>
        <loginSec:pw>this is a long password</loginSec:pw>
      </loginSec:loginSec>
    </extension>
…
```

Feedback from Mailing List

- Support for additional authentication methods (2FA, Digest)?
- Use of *[LOGIN-SECURITY]* constant value
- Setting of minimum password length to 6 characters
- Format of the password
  - Use of XML schema ”token” type
  - PRECIS framework (RFC 7564 and 8265)

Conclusion

- Login Security Extension addresses the 3 problems via
  - Login Command Extension
    - Extending the password past the RFC 5730 16-character maximum
    - Enabling the client to provide user agent information to the server
  - Login Response Extension
    - Enabling the server to provide login security warnings and errors
- Please review the draft and provide feedback on the mailing list