Login Security Extension for the Extensible Provisioning Protocol (EPP)
draft-ietf-regext-login-security-05

Abstract

The Extensible Provisioning Protocol (EPP) includes a client
authentication scheme that is based on a user identifier and
password. The structure of the password field is defined by an XML
Schema data type that specifies minimum and maximum password length
values, but there are no other provisions for password management
other than changing the password. This document describes an EPP
extension that allows longer passwords to be created and adds
additional security features to the EPP login command and response.

Status of This Memo

This Internet-Draft is submitted in full conformance with the
provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering
Task Force (IETF). Note that other groups may also distribute
working documents as Internet-Drafts. The list of current Internet-
Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months
and may be updated, replaced, or obsoleted by other documents at any
time. It is inappropriate to use Internet-Drafts as reference
material or to cite them other than as "work in progress."

This Internet-Draft will expire on May 1, 2020.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the
document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal
Provisions Relating to IETF Documents
(https://trustee.ietf.org/license-info) in effect on the date of
publication of this document. Please review these documents
carefully, as they describe your rights and restrictions with respect
to this document. Code Components extracted from this document must
include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction ..................................... 2
   1.1. Conventions Used in This Document ............ 3
2. Migrating to Newer Versions of This Extension ........ 3
3. Object Attributes .................................. 4
   3.1. Event ........................................ 4
   3.2. "[LOGIN-SECURITY]" Password .................. 5
   3.3. Dates and Times ................................ 6
4. EPP Command Mapping ................................ 6
   4.1. EPP <login> Command ........................... 6
5. Formal Syntax ....................................... 14
   5.1. IANA Considerations ............................. 14
6. Implementation Status ................................ 17
   6.1. Verisign EPP SDK ................................ 18
7. Security Considerations .............................. 18
8. Acknowledgements .................................... 19
9. References .......................................... 19
   10.1. Normative References ............................ 19
   10.2. Informative References .......................... 19
   10.3. URIs .......................................... 20
Appendix A. Change History ............................. 20
   A.1. Change from 00 to 01 ............................ 20
   A.2. Change from 01 to 02 ............................ 20
   A.3. Change from 02 to 03 ............................ 20
   A.4. Change from 03 to REGEXT 00 .................... 20
   A.5. Change from REGEXT 00 to REGEXT 01 ............ 20
   A.6. Change from REGEXT 01 to REGEXT 02 ............ 21
   A.7. Change from REGEXT 02 to REGEXT 03 ............ 21
   A.8. Change from REGEXT 03 to REGEXT 04 ............ 22
   A.9. Change from REGEXT 04 to REGEXT 05 ............ 22
   Authors’ Addresses .................................. 22

1. Introduction

This document describes an Extensible Provisioning Protocol (EPP) extension for enhancing the security of the EPP login command in EPP RFC 5730. The enhancements include supporting longer passwords (or passphrases) than the 16-character maximum and providing a list of security events in the login response. The password (current and new) in EPP RFC 5730 can be overridden by the password included in
the extension to extend past the 16-character maximum. The security events supported include: password expiry, client certificate expiry, insecure cipher, insecure TLS protocol, new password complexity, login security statistical warning, and a custom event. The attributes supported by the security events include identifying the event type or sub-type, indicating the security level of warning or error, a future or past-due expiration date, the value that resulted in the event, the duration of the statistical event, and a free-form description with an optional language.

1.1. Conventions Used in This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

XML is case sensitive. Unless stated otherwise, XML specifications and examples provided in this document MUST be interpreted in the character case presented in order to develop a conforming implementation.

In examples, "C:" represents lines sent by a protocol client and "S:" represents lines returned by a protocol server. Indentation and white space in examples are provided only to illustrate element relationships and are not a required feature of this protocol.

"loginSec-1.0" is used as an abbreviation for "urn:ietf:params:xml:ns:epp:loginSec-1.0". The XML namespace prefix "loginSec" is used, but implementations MUST NOT depend on it and instead employ a proper namespace-aware XML parser and serializer to interpret and output the XML documents.

2. Migrating to Newer Versions of This Extension

Servers which implement this extension SHOULD provide a way for clients to progressively update their implementations when a new version of the extension is deployed.

Servers SHOULD (for a temporary migration period) provide support for older versions of the extension in parallel to the newest version, and allow clients to select their preferred version via the <svcExtension> element of the <login> command.

If a client requests multiple versions of the extension at login, then, when preparing responses to commands which do not include extension elements, the server SHOULD only include extension elements
in the namespace of the newest version of the extension requested by the client.

When preparing responses to commands which do include extension elements, the server SHOULD only include extension elements for the extension versions present in the command.

3. Object Attributes

This extension adds additional elements to [RFC5730] login command and response. Only those new elements are described here.

3.1. Event

A security event, using the <loginSec:event> element, represents either a warning or error identified by the server after the client has connected and submitted the login command. There MAY be multiple events returned that provide information for the client to address. The <loginSec:event> MAY include a free-form description. All of the security events use a consistent set of attributes, where the exact set of applicable attributes is based on the event type. The supported set of <loginSec:event> element attributes include:

"type": A REQUIRED attribute that defines the type of security event. The enumerated list of "type" values includes:

"password": Identifies a password expiry event, where the password expires in the future or has expired based on the "exDate" date and time.
"certificate": Identifies a client certificate expiry event, where the client certificate will expire at the "exDate" date and time.
"cipher": Identifies the use of an insecure or deprecated TLS cipher suite.
"tlsProtocol": Identifies the use of an insecure or deprecated TLS protocol.
"newPW": The new password does not meet the server password complexity requirements.
"stat": Provides a login security statistical warning that MUST set the "name" attribute to the name of the statistic subtype.
"custom": Custom event type that MUST set the "name" attribute with the custom event type name.
"name": Used to define a sub-type when the "type" attribute is not "custom" or the full type name when the "type" attribute is "custom". The "name" attribute MUST be set when the "type" attribute is "stat" or "custom".
"level": Defines the level of the event as either "warning" for a warning event that needs action, or "error" for an error event that requires immediate action.
"exDate": Contains the date and time that a "warning" level has or will become an "error" level. At expiry there MAY be an error to connect or MAY be an error to login. An example is an expired certificate that will result in an error to connect or an expired password that may result in a failed login.
"value": Identifies the value that resulted in the login security event. An example is the negotiated insecure cipher suite or the negotiated insecure TLS protocol.
"duration": Defines the duration that a statistical event is associated with, ending when the login command was received. The format of the duration is defined by the duration primitive datatype in [W3C.REC-xmlschema-2-20041028].
"lang": Identifies the negotiated language of the free-form description. The default is "en" (English).

Example login security event for password expiration, where the current date is 2018-03-25:

<loginSec:event
type="password"
level="warning"
exDate="2018-04-01T22:00:00.0Z"
lang="en">
Password expiration soon
</loginSec:event>

Example login security event for identifying 100 failed logins over the last day, using the "stat" sub-type of "failedLogins":

<loginSec:event
type="stat"
name="failedLogins"
level="warning"
value="100"
duration="P1D">
Excessive invalid daily logins
</loginSec:event>

3.2. "[LOGIN-SECURITY]" Password

The <loginSec:pw> element MUST override the [RFC5730] <pw> element only if the <pw> contains the predefined value of "[LOGIN-SECURITY]", which is a constant value for the server to use the <loginSec:pw> element for the password. Similarly, the <loginSec:newPW> element MUST override the [RFC5730] <newPW> element only if the <newPW>
contains the predefined value of "[LOGIN-SECURITY]", which is a
constant value for the server to use the <loginSec:newPW> element for
the new password. The "[LOGIN-SECURITY]" pre-defined string MUST be
supported by the server for the client to explicitly indicate to the
server whether to use <loginSec:pw> element in place of the [RFC5730]
<pw> element or to use the <loginSec:newPW> in place of the [RFC5730]
<newPW> element. The server MUST NOT allow the client to set the
password to the value "[LOGIN-SECURITY]".

3.3. Dates and Times

Date and time attribute values MUST be represented in Universal
Coordinated Time (UTC) using the Gregorian calendar. The extended
date-time form using upper case "T" and "Z" characters defined in
[W3C.REC-xmlschema-2-20041028] MUST be used to represent date-time
values, as XML Schema does not support truncated date-time forms or
lower case "T" and "Z" characters.

4. EPP Command Mapping

A detailed description of the EPP syntax and semantics can be found
in the EPP core protocol specification [RFC5730].

4.1. EPP <login> Command

This extension defines additional elements to extend the EPP <login>
command and response to be used in conjunction with [RFC5730].

The EPP <login> command is used to establish a session with an EPP
server. This extension overrides the password that is passed with
the [RFC5730] <pw> or the <newPW> element as defined in Section 3.2.
A <loginSec:loginSec> element is sent along with the [RFC5730]
<login> command and MUST contain at least one of the following child
elements:

<loginSec:userAgent>: OPTIONAL client user agent that identifies the
client application software, technology, and operating system
used by the server to identify functional or security
constraints, current security issues, and potential future
functional or security issues for the client. The
<loginSec:userAgent> element MUST contain at least one of the
following child elements:

<loginSec:app>: OPTIONAL name of the client application software
with version if available, such as the name of the client SDK
"EPP SDK 1.0.0".
<loginSec:tech>: OPTIONAL technology used for the client software with version if available, such as "Java 11.0.2".
<loginSec:os>: OPTIONAL client operating system used with version if available, such as "x86_64 Mac OS X 10.11.6".
<loginSec:pw>: OPTIONAL plain text password that is case sensitive, has a minimum length of 6 characters, and has a maximum length that is up to server policy. All leading and trailing whitespace is removed, and all internal contiguous whitespace that includes \#x9 (tab), \#xA (linefeed), \#xD (carriage return), and \#x20 (space) is replaced with a single \#x20 (space). This element MUST only be used if the [RFC5730] <pw> element is set to the "[LOGIN-SECURITY]" value.
<loginSec:newPW>: OPTIONAL plain text new password that is case sensitive, has a minimum length of 6 characters, and has a maximum length that is up to server policy. All leading and trailing whitespace is removed, and all internal contiguous whitespace that includes \#x9 (tab), \#xA (linefeed), \#xD (carriage return), and \#x20 (space) is replaced with a single \#x20 (space). This element MUST only be used if the [RFC5730] <newPW> element is set to the "[LOGIN-SECURITY]" value.
Example login command that uses the `<loginSec:pw>` element instead of the [RFC5730] `<pw>` element to establish the session and includes the `<loginSec:userAgent>` element:

C: <?xml version="1.0" encoding="UTF-8" standalone="no"?>
C: <epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <login>
C:      <clID>ClientX</clID>
C:      <pw>[LOGIN-SECURITY]</pw>
C:      <options>
C:        <version>1.0</version>
C:        <lang>en</lang>
C:      </options>
C:      <svcs>
C:        <objURI>urn:ietf:params:xml:ns:obj1</objURI>
C:        <objURI>urn:ietf:params:xml:ns:obj2</objURI>
C:        <svcExtension>
C:          <extURI>urn:ietf:params:xml:ns:epp:loginSec-1.0</extURI>
C:        </svcExtension>
C:      </svcs>
C:    </login>
C:    <extension>
C:      <loginSec:loginSec
C:        xmlns:loginSec="urn:ietf:params:xml:ns:epp:loginSec-1.0">
C:        <loginSec:userAgent>
C:          <loginSec:app>EPP SDK 1.0.0</loginSec:app>
C:          <loginSec:tech>Java 11.0.2</loginSec:tech>
C:          <loginSec:os>x86_64 Mac OS X 10.11.6</loginSec:os>
C:        </loginSec:userAgent>
C:        <loginSec:pw>this is a long password</loginSec:pw>
C:      </loginSec:loginSec>
C:    </extension>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
Example login command that uses the `<loginSec:pw>` element instead of the [RFC5730] `<pw>` element to establish the session, and uses the `<loginSec:newPW>` element instead of the [RFC5730] `<newPW>` element to set the new password:

```
C: <?xml version="1.0" encoding="UTF-8" standalone="no"?>
C: <epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <login>
C:      <clID>ClientX</clID>
C:      <pw>[LOGIN-SECURITY]</pw>
C:      <newPW>[LOGIN-SECURITY]</newPW>
C:      <options>
C:        <version>1.0</version>
C:        <lang>en</lang>
C:      </options>
C:      <svcs>
C:        <objURI>urn:ietf:params:xml:ns:obj1</objURI>
C:        <objURI>urn:ietf:params:xml:ns:obj2</objURI>
C:        <svcExtension>
C:          <extURI>urn:ietf:params:xml:ns:epp:loginSec-1.0</extURI>
C:        </svcExtension>
C:      </svcs>
C:    </login>
C:    <extension>
C:      <loginSec:loginSec
C:        xmlns:loginSec="urn:ietf:params:xml:ns:epp:loginSec-1.0">
C:        <loginSec:pw>this is a long password
C:        </loginSec:pw>
C:        <loginSec:newPW>new password that is still long
C:        </loginSec:newPW>
C:      </loginSec:loginSec>
C:    </extension>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```
Example login command that uses the [RFC5730] <pw> element to establish the session, and uses the <loginSec:newPW> element instead of the [RFC5730] <newPW> element to set the new password:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <login>
C:      <clID>ClientX</clID>
C:      <pw>shortpassword</pw>
C:      <newPW>[LOGIN-SECURITY]</newPW>
C:      <options>
C:        <version>1.0</version>
C:        <lang>en</lang>
C:      </options>
C:      <svcs>
C:        <objURI>urn:ietf:params:xml:ns:obj1</objURI>
C:        <objURI>urn:ietf:params:xml:ns:obj2</objURI>
C:        <svcExtension>
C:          <extURI>urn:ietf:params:xml:ns:epp:loginSec-1.0</extURI>
C:        </svcExtension>
C:      </svcs>
C:    </login>
C:    <extension>
C:      <loginSec:loginSec
C:        xmlns:loginSec="urn:ietf:params:xml:ns:epp:loginSec-1.0">
C:        <loginSec:newPW>new password that is still long</loginSec:newPW>
C:      </loginSec:loginSec>
C:    </extension>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

Upon a completed login command (success or failed), the extension MUST be included in the response based on both of the following conditions:

- Client supports extension: The client supports the extension based on the <svcExtension> element of the <login> command.
- At least one login security event: The server has identified at least one login security event to communicate to the client.

The extension to the EPP response uses the <loginSec:loginSecData> element that contains the following child elements:
<loginSec:event>: One or more <loginSec:event> elements defined in Section 3.1.

Example EPP response to a successful login command on 2018-03-25, where the password will expire in a week:

S: <?xml version="1.0" encoding="UTF-8" standalone="no"?>
S: <epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S:  <response>
S:    <result code="1000">
S:      <msg>Command completed successfully</msg>
S:    </result>
S:    <extension>
S:      <loginSec:loginSecData
S:        xmlns:loginSec=
S:          "urn:ietf:params:xml:ns:epp:loginSec-1.0">
S:        <loginSec:event
S:          type="password"
S:          level="warning"
S:          exDate="2018-04-01T22:00:00.0Z"
S:          lang="en">
S:          Password expiring in a week
S:        </loginSec:event>
S:      </loginSec:loginSecData>
S:    </extension>
S:    <trID>
S:      <clTRID>ABC-12345</clTRID>
S:      <svTRID>54321-XYZ</svTRID>
S:    </trID>
S:  </response>
S:</epp>
Example EPP response to a failed login command where the password has expired and the new password does not meet the server complexity requirements:

S: <?xml version="1.0" encoding="UTF-8" standalone="no"?>
S: <epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S:  <response>
S:    <result code="2200">
S:      <msg>Authentication error</msg>
S:    </result>
S:    <extension>
S:      <loginSec:loginSecData
S:        xmlns:loginSec=
S:          "urn:ietf:params:xml:ns:epp:loginSec-1.0">
S:        <loginSec:event
S:          type="password"
S:          level="error"
S:          exDate="2018-03-26T22:00:00.0Z">
S:          Password has expired
S:        </loginSec:event>
S:        <loginSec:event
S:          type="newPW"
S:          level="error">
S:          New password does not meet complexity requirements
S:        </loginSec:event>
S:      </loginSec:loginSecData>
S:    </extension>
S:    <trID>
S:      <clTRID>ABC-12345</clTRID>
S:      <svTRID>54321-XYZ</svTRID>
S:    </trID>
S:  </response>
S: </epp>

Example EPP response to a successful login command where there is a set of login security events:

S: <?xml version="1.0" encoding="UTF-8" standalone="no"?>
S: <epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S:  <response>
S:    <result code="1000">
S:      <msg>Command completed successfully</msg>
S:    </result>
S:    <extension>
S:      <loginSec:loginSecData
S:        xmlns:loginSec=
S:          "urn:ietf:params:xml:ns:epp:loginSec-1.0">
S:        <loginSec:event
S:          type="newPW"
S:          level="error">
S:          New password does not meet complexity requirements
S:        </loginSec:event>
S:      </loginSec:loginSecData>
S:    </extension>
S:    <trID>
S:      <clTRID>ABC-12345</clTRID>
S:      <svTRID>54321-XYZ</svTRID>
S:    </trID>
S:  </response>
S: </epp>
type="password"
S: level="warning"
S: exDate="2018-04-01T22:00:00.0Z"
S: lang="en">
S: Password expiration soon
S: </loginSec:event>
S: <loginSec:event
S: type="certificate"
S: level="warning"
S: exDate="2018-04-02T22:00:00.0Z"/>
S: <loginSec:event
S: type="cipher"
S: level="warning"
S: value="TLS_RSA_WITH_AES_128_CBC_SHA">
S: Non-PFS Cipher negotiated
S: </loginSec:event>
S: <loginSec:event
S: type="tlsProtocol"
S: level="warning"
S: value="TLSv1.0">
S: Insecure TLS protocol negotiated
S: </loginSec:event>
S: <loginSec:event
S: type="stat"
S: name="failedLogins"
S: level="warning"
S: value="100"
S: duration="P1D">
S: Excessive invalid daily logins
S: </loginSec:event>
S: <loginSec:event
S: type="custom"
S: name="myCustomEvent"
S: level="warning">
S: A custom login security event occurred
S: </loginSec:event>
S: </loginSec:loginSecData>
S: </extension>
S: <trID>
S: <clTRID>ABC-12345</clTRID>
S: <svTRID>54321-XYZ</svTRID>
S: </trID>
S: </response>
S:</epp>
5. Formal Syntax

One schema is presented here that is the EPP Login Security Extension schema.

The formal syntax presented here is a complete schema representation of the object mapping suitable for automated validation of EPP XML instances. The BEGIN and END tags are not part of the schema; they are used to note the beginning and ending of the schema for URI registration purposes.

5.1. Login Security Extension Schema

BEGIN

<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:epp="urn:ietf:params:xml:ns:epp-1.0"
  xmlns:eppcom="urn:ietf:params:xml:ns:eppcom-1.0"
  xmlns:loginSec="urn:ietf:params:xml:ns:epp:loginSec-1.0"
  targetNamespace="urn:ietf:params:xml:ns:epp:loginSec-1.0"
  elementFormDefault="qualified">
  <!-- Import common element types. -->
  <import namespace="urn:ietf:params:xml:ns:eppcom-1.0" />
  <import namespace="urn:ietf:params:xml:ns:epp-1.0" />
  <annotation>
    <documentation>Extensible Provisioning Protocol v1.0 Login Security Extension Schema.</documentation>
  </annotation>
  <!-- Login command extension elements -->
  <element name="loginSec" type="loginSec:loginSecType" />
  <!-- Attributes associated with the login command extension. -->
  <complexType name="loginSecType">
    <sequence>
      <element name="userAgent"
        type="loginSec:userAgentType" minOccurs="0" />
      <element name="pw"
        type="loginSec:pwType" minOccurs="0" />
      <element name="newPW"
        type="loginSec:pwType" minOccurs="0" />
    </sequence>
  </complexType>
  <simpleType name="pwType">
    <restriction base="token">
      <minLength value="6" />
    </restriction>
  </simpleType>
</schema>

END
### User Agent Type

```xml
<complexType name="userAgentType">
  <choice>
    <sequence>
      <element name="app" type="token" />
      <element name="tech" type="token" minOccurs="0" />
      <element name="os" type="token" minOccurs="0" />
    </sequence>
    <sequence>
      <element name="tech" type="token" />
      <element name="os" type="token" minOccurs="0" />
    </sequence>
    <element name="os" type="token" />  
  </choice>
</complexType>
```

### Login Response Extension Elements

```xml
<!-- Login response extension elements -->
<element name="loginSecData" type="loginSec:loginSecDataType" />
<complexType name="loginSecDataType">
  <sequence>
    <element name="event" type="loginSec:eventType" minOccurs="1" maxOccurs="unbounded" />  
  </sequence>
</complexType>
```

### Security Event Element

```xml
<!-- Security event element -->
<complexType name="eventType">
  <simpleContent>
    <extension base="normalizedString">
      <attribute name="type" type="loginSec:typeEnum" use="required" />
      <attribute name="name" type="token" />
      <attribute name="level" type="loginSec:levelEnum" use="required" />
      <attribute name="exDate" type="dateTime" />
      <attribute name="value" type="token" />
      <attribute name="duration" type="duration" />
    </extension>
  </simpleContent>
</complexType>
```
<attribute name="lang" type="language" default="en" />
</extension>
</simpleContent>
</complexType>
<!--
Enumerated list of event types, with extensibility via "custom".
-->
<simpleType name="typeEnum">
<restriction base="token">
  <enumeration value="password" />
  <enumeration value="certificate" />
  <enumeration value="cipher" />
  <enumeration value="tlsProtocol" />
  <enumeration value="newPW" />
  <enumeration value="stat" />
  <enumeration value="custom" />
</restriction>
</simpleType>
<!--
Enumerated list of levels.
-->
<simpleType name="levelEnum">
<restriction base="token">
  <enumeration value="warning" />
  <enumeration value="error" />
</restriction>
</simpleType>
<!--
End of schema.
-->
</schema>

6. IANA Considerations

6.1. XML Namespace

This document uses URNs to describe XML namespaces and XML schemas conforming to a registry mechanism described in [RFC3688]. The following URI assignment is requested of IANA:

Registration request for the loginSec namespace:

URI: urn:ietf:params:xml:ns:epp:loginSec-1.0
Registrant Contact: IESG
XML: None. Namespace URIs do not represent an XML specification.
Registration request for the loginSec XML schema:

URI: urn:ietf:params:xml:schema:epp:loginSec-1.0
Registrant Contact: IESG
XML: See the "Formal Syntax" section of this document.

6.2. EPP Extension Registry

The EPP extension described in this document should be registered by the IANA in the EPP Extension Registry described in [RFC7451]. The details of the registration are as follows:

Name of Extension: "Login Security Extension for the Extensible Provisioning Protocol (EPP)"

Document status: Standards Track

Reference: (insert reference to RFC version of this document)

Registrant Name and Email Address: IESG, <iesg@ietf.org>

TLDs: Any

IPR Disclosure: None

Status: Active

Notes: None

7. Implementation Status

Note to RFC Editor: Please remove this section and the reference to RFC 7942 [RFC7942] before publication.

This section records the status of known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft, and is based on a proposal described in RFC 7942 [RFC7942]. The description of implementations in this section is intended to assist the IETF in its decision processes in progressing drafts to RFCs. Please note that the listing of any individual implementation here does not imply endorsement by the IETF. Furthermore, no effort has been spent to verify the information presented here that was supplied by IETF contributors. This is not intended as, and must not be construed to be, a catalog of available implementations or their features. Readers are advised to note that other implementations may exist.
According to RFC 7942 [RFC7942], "this will allow reviewers and working groups to assign due consideration to documents that have the benefit of running code, which may serve as evidence of valuable experimentation and feedback that have made the implemented protocols more mature. It is up to the individual working groups to use this information as they see fit".

7.1. Verisign EPP SDK

Organization: Verisign Inc.
Name: Verisign EPP SDK
Description: The Verisign EPP SDK includes both a full client implementation and a full server stub implementation of draft-ietf-regext-login-security.
Level of maturity: Development
Coverage: All aspects of the protocol are implemented.
Licensing: GNU Lesser General Public License
Contact: jgould@verisign.com

8. Security Considerations

The extension leaves the password (<pw> element) and new password (<newPW> element) minimum length beyond 6 characters and the maximum length up to sever policy. The server SHOULD enforce minimum and maximum length requirements that are appropriate for their operating environment. One example of a guideline for password length policies can be found in section 5 of NIST Special Publication 800-63B [1].

The client SHOULD NOT decrease the security of a new password by decreasing the length of the current password. For example, a client with a 20 character password set using the extension, should not use the login command in [RFC5730] without using the extension, to set a new password that is less than or equal to 16 characters.

The extension provides an extensible list of login security events to inform clients of connection and login warnings and errors.
9. Acknowledgements

The authors wish to thank the following persons for their feedback and suggestions:

- Martin Casanova
- Scott Hollenbeck
- Patrick Mevzek
- Joseph Yee

10. References

10.1. Normative References


10.2. Informative References


10.3. URIs


Appendix A. Change History

A.1. Change from 00 to 01

1. Based on the feedback from Patrick Mevzek and a proposal from Scott Hollenbeck, changed the minimum length of the password from 8 to 6, revised the description of the password, and added text in the Security Considerations section for the server password length policy.

A.2. Change from 01 to 02


A.3. Change from 02 to 03

1. Updates based on the review by Patrick Mevzek, that include:
   1. Fix the inconsistent case for newPW, that required a global change in the draft text and an update to the XML schema to "urn:ietf:params:xml:ns:loginSec-0.3".
   2. Changed "contains the following child elements" to "MUST contain at least one of the following child elements", section "EPP <login> Command" to ensure that an empty <loginSec:loginSec> element is not passed.
   3. Add "The client SHOULD NOT decrease the security of a new password by decreasing the length of the current password." along with an example to the "Security Considerations" section.

A.4. Change from 03 to REGEXT 00

1. Changed to regext working group draft by changing draft-gould-regext-login-security to draft-ietf-regext-login-security.

A.5. Change from REGEXT 00 to REGEXT 01

1. Changed the <loginSec:userAgent> element to be structured with the <loginSec:app>, <loginSec:tech>, and <loginSec:os> sub-elements. This was based on the feedback from Martin Casanova.
This resulted in the need to change the XML namespace from
urn:ietf:params:xml:ns:epp:loginSec-0.3 to
urn:ietf:params:xml:ns:epp:loginSec-0.4.

A.6. Change from REGEXT 01 to REGEXT 02

1. Updated the Implementation Status section from "TBD" to include
   the Verisign EPP SDK implementation.

A.7. Change from REGEXT 02 to REGEXT 03

1. Revised the description of the "duration" attribute to clarify
   that it ends when the login command was received and to clarify
   the format, based on the feedback from Martin Casanova.
2. Revised the sentence ‘Upon a completed login command (success or
   failed), the extension MUST be included in the response based on
   the following conditions:’ to ‘Upon a completed login command
   (success or failed), the extension MUST be included in the
   response based on both of the following conditions:’ based on the
   feedback from Patrick Mevzek.
3. Updates based on the review by Joseph Yee, that include:
   1. Revised the description of the <loginSec:event> "name"
      attribute read ‘Used to define a sub-type when the "type"
      attribute is not "custom" or the full type name when the
      "type" attribute is "custom"’. The definition of the "stat"
      type was updated to ‘Provides a login security statistical
      warning that MUST set the "name" attribute to the name of the
      statistic.’
2. Added the following sentence ‘The server MUST NOT allow the
   client to set the password to the value "[LOGIN-SECURITY]".’
   to address the corner case where the constant is used as the
   password.
3. Revised the description of the <loginSec:userAgent> element
   to read ‘The <loginSec:userAgent> element MUST contain at
   least one of the following child elements:’.
4. Revised the description of the <loginSec:userAgent> to match the
   child elements that can be passed, by changing "client software"
   to "client application software" and change "language" to
   "technology".
5. Changed the XML namespace from
   urn:ietf:params:xml:ns:epp:loginSec-0.4 to
A.8. Change from REGENEXT 03 to REGENEXT 04

Updates based on the review by Joseph Yee, that include:

1. Update the definition of the "stat" security event type to reference sub-type to match the language for the "name" attribute.
2. Added the sentence 'The "name" attribute MUST be set when the "type" attribute is "stat" or "custom".' to the definition of the "name" attribute for clarity.
3. Update the definition of the "userAgentType" in the XML schema to require at least one sub-element using a <choice> element.

A.9. Change from REGENEXT 04 to REGENEXT 05

Updates based on the review by Barry Leiba, that include:

1. In section 1.1, updated to use BCP 14 boilerplate and references as defined in RFC 8174.
2. In section 1.1, change "REQUIRED" to "required".
3. Keep the "Migration to Newer Versions of This Extension" section by removing the note for removal to the RFC Editor.
4. In section 3.1, change "MAY be multiple events returned that provides information" to "MAY be multiple events returned that provide information".
5. In section 3.1, change "free form" to "free-form".
6. In section 3.1, change "The enumerated list of "type" values include:" to "The enumerated list of "type" values includes:".
7. In section 3.1, change "Identifies the language of the free-form description if the negotiated language is something other than the default value of "en" (English)." to "Identifies the negotiated language of the free-form description. The default is "en" (English)."
8. In section 3.1, change example description from "Example login security event for a password expiring in a week:" to "Example login security event for password expiration, where the current date is 2018-03-25:".
9. In section 4.1, change "Example EPP response to a successful login command where the password will expire in a week:" to "Example EPP response to a successful login command on 2018-03-25, where the password will expire in a week:".

Authors’ Addresses
James Gould
VeriSign, Inc.
12061 Bluemont Way
Reston, VA 20190
US
Email: jgould@verisign.com
URI: http://www.verisign.com

Matthew Pozun
VeriSign, Inc.
12061 Bluemont Way
Reston, VA 20190
US
Email: mpozun@verisign.com
URI: http://www.verisign.com