ï»¿

Internet Engineering Task Force (IETF)
Request for Comments: 8807
Category: Standards Track
ISSN: 2070-1721
August 2020

J. Gould
M. Pozun
VeriSign, Inc.

Login Security Extension for the Extensible Provisioning Protocol (EPP)

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

Copyright Notice

Table of Contents

1. Introduction

This document describes an Extensible Provisioning Protocol (EPP) extension for enhancing the security of the EPP login command in EPP [RFC5730]. EPP [RFC5730] includes a maximum password length of 16 characters, which inhibits implementing stronger password security policies with higher entropy. 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 [RFC5730] 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 an identified event type or a subtype, an indicated 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. In examples, indentation and whitespace 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. Instead, they are to employ a proper namespace-aware XML parser and serializer to interpret and output the XML documents.

"whitespace" is defined by the XML Schema whiteSpace data type in [W3C.REC-xmlschema-2-20041028], which only includes the ASCII whitespace characters #x9 (tab), #xA (linefeed), #xD (carriage return), and #x20 (space).

2. Migrating to Newer Versions of This Extension

Servers that implement this extension SHOULD provide a way for clients to progressively update their implementations when a new version of the extension is deployed. A newer version of the extension is expected to use an XML namespace with a higher version number than the prior versions.

Servers SHOULD (for a temporary migration period up to server policy) 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 that 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 that 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. The <loginSec:event> element is contained in a list of one or more elements in the <loginSec:loginSecData> element, so 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. The "exDate" attribute MUST be set with the password expiry date and time.
>
> "certificate": Identifies a client certificate expiry event where the client certificate will expire at the "exDate" date and time. The "exDate" attribute MUST be set with the certificate expiry date and time.
>
> "cipher": Identifies the use of an insecure or deprecated TLS cipher suite. The "name" attribute MUST be set with the name of the cipher suite, which is free-form and is not expected to be parsed and automatically addressed by the client. An example of cipher suite names can be found in the TLS Cipher Suites of the "Transport Layer Security (TLS) Parameters" registry (https://www.iana.org/assignments/tls-parameters/tls-parameters.xhtml#tls-parameters-4).
>
> "tlsProtocol": Identifies the use of an insecure or deprecated TLS protocol. The "name" attribute MUST be set with the name of the TLS protocol, which is free-form and is not expected to be parsed and automatically addressed by the client.
>
> "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 subtype 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". The possible set of "name" values, by event type, can be discovered/negotiated out of band to EPP or using a separate EPP extension designed to provide server policy information to the client.

"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 a connection failure or MAY be a login failure. An example is an expired certification that will result in a connection failure or an expired password that may result in a login failure.

"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 data type in Section 3.2.6 of [W3C.REC-xmlschema-2-20041028].

"lang": Identifies the negotiated language of the free-form description. The format of the language is defined by the language primitive data type in Section 3.3.3 of [W3C.REC-xmlschema-2-20041028]. The default is "en" (English).

Example login security event for password expiration, where the current date is 2020-03-25:

```
<loginSec:event
  type="password"
  level="warning"
  exDate="2020-04-01T22:00:00.0Z"
  lang="en">
  Password expiration soon
</loginSec:event>
```

Example login security event for identifying 100 failed logins over the last day, using the "stat" subtype 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

When the [RFC5730] <pw> element contains the predefined value of "[LOGIN-SECURITY]", the <loginSec:pw> element overrides the <pw> element, which is a constant value for the server to use the <loginSec:pw> element for the password. Similarly, when the [RFC5730] <newPw> element contains the predefined value of "[LOGIN-SECURITY]", the <loginSec:newPw> element overrides the <newPw> element, which is a constant value for the server to use the <loginSec:newPW> element for the new password. The "[LOGIN-SECURITY]" predefined 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 information 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 server may use the information for real-time identification and client notification of security issues, such as keying off of the client application software for executing security rule checks. The server may capture the information to identify future security policy issues, such as deprecating or removing TLS cipher suites or TLS protocols. 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". The <loginSec:app> element value can be created by appending the version number to the name of the application software, such as the Augmented Backus-Naur Form (ABNF) grammar [RFC5234] format:

```
app = name SP version
name = 1*VCHAR
version = 1*VCHAR
```

<loginSec:tech>: OPTIONAL technology used for the client software with version if available, such as "Vendor Java 11.0.6". The <loginSec:tech> element value can be created by including the technology vendor, technology name, and technology version, such as the Augmented Backus-Naur Form (ABNF) grammar [RFC5234] format:

```
tech = vendor SP name SP version
vendor = 1*VCHAR
name = 1*VCHAR
version = 1*VCHAR
```

<loginSec:os>: OPTIONAL client operating system used with version if available, such as "x86_64 Mac OS X 10.15.2". The <loginSec:os> element value can be created by including the operating system architecture, operating system name, and operating system version, such as the Augmented Backus-Naur Form (ABNF) grammar [RFC5234] format:

```
os = arch SP name SP version
arch = 1*VCHAR
name = 1*VCHAR
version = 1*VCHAR
```

<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 set 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 set if the [RFC5730] <newPW> element is set to the "[LOGIN-SECURITY]" value.

It is RECOMMENDED that the plain text password in the <loginSec:pw> and <loginSec:newPw> elements use printable ASCII characters #x20 (space) - #x7E (~) with high entropy, such as 128 bits. If non-ASCII characters are supported with the plain text password, then use a standard for passwords with international characters; the OpaqueString PRECIS profile in [RFC8265] is recommended in the absence of other considerations.

Example login command that uses the <loginSec:pw> element instead of the <pw> element ([RFC5730]) 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:        <objURI>urn:ietf:params:xml:ns:obj3</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=
C:          "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>Vendor Java 11.0.6</loginSec:tech>
C:          <loginSec:os>x86_64 Mac OS X 10.15.2</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 <pw> element ([RFC5730]) to establish the session and that uses the <loginSec:newPW> element instead of the <newPW> element ([RFC5730]) 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:        <objURI>urn:ietf:params:xml:ns:obj3</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=
C:          "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 <pw> element ([RFC5730]) to establish the session and that uses the <loginSec:newPW> element instead of the <newPW> element ([RFC5730]) 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:        <objURI>urn:ietf:params:xml:ns:obj3</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=
C:          "urn:ietf:params:xml:ns:epp:loginSec-1.0">
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>
```

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

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 2020-03-25, 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="2020-04-01T22:00:00.0Z"
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="2020-03-24T22:00:00.0Z">
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="password"
S:          level="warning"
S:          exDate="2020-04-01T22:00:00.0Z"
S:          lang="en">
S:          Password expiration soon
S:        </loginSec:event>
S:        <loginSec:event
S:          type="certificate"
S:          level="warning"
S:          exDate="2020-04-02T22:00:00.0Z"/>
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

The EPP Login Security Extension schema is presented here.

The formal syntax shown here is a complete XML Schema representation of the object mapping suitable for automated validation of EPP XML instances. The <CODE BEGINS> and <CODE ENDS> tags are not part of the XML Schema; they are used to note the beginning and ending of the XML Schema for URI registration purposes.

5.1. Login Security Extension Schema

<CODE BEGINS>

```
<?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>
  <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 -->
  <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 -->
  <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" />
           <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>
   <CODE ENDS>
```

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 has been made by 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 has been registered by IANA in the "Extensions for the Extensible Provisioning Protocol (EPP)" 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: RFC 8807
Registrant Name and Email Address: IESG, <iesg@ietf.org>
Top-Level Domains(TLDs): Any
IPR Disclosure: None
Status: Active
Notes: None

7. Security Considerations

The security considerations of [RFC5730] apply in this document, and this document enhances these considerations.

The extension leaves the password (<pw> element) and new password (<newPW> element) minimum length greater than 6 characters and the maximum length up to server 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 (https://pages.nist.gov/800-63-3/sp800-63b.html).

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. The server returning of security events to unauthenticated users needs to take into account the security/privacy issues of returning information to potential attackers.

The user-agent information represents the client system of a system-to-system interface, so the user-agent information MUST NOT provide any ability to track individual users or classes of users.

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <https://www.rfc-editor.org/info/rfc3688>.

[RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <https://www.rfc-editor.org/info/rfc5234>.

[RFC5730] Hollenbeck, S., "Extensible Provisioning Protocol (EPP)", STD 69, RFC 5730, DOI 10.17487/RFC5730, August 2009, <https://www.rfc-editor.org/info/rfc5730>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[W3C.REC-xmlschema-2-20041028]
Biron, P. and A. Malhotra, "XML Schema Part 2: Datatypes Second Edition", W3C Recommendation REC-xmlschema-2-20041028, October 2004, <http://www.w3.org/TR/2004/REC-xmlschema-2-20041028>.

8.2. Informative References

[RFC7451] Hollenbeck, S., "Extension Registry for the Extensible Provisioning Protocol", RFC 7451, DOI 10.17487/RFC7451, February 2015, <https://www.rfc-editor.org/info/rfc7451>.

[RFC8265] Saint-Andre, P. and A. Melnikov, "Preparation, Enforcement, and Comparison of Internationalized Strings Representing Usernames and Passwords", RFC 8265, DOI 10.17487/RFC8265, October 2017, <https://www.rfc-editor.org/info/rfc8265>.

Acknowledgements

The authors wish to thank the following persons for their feedback and suggestions: Martin Casanova, Scott Hollenbeck, Barry Leiba, Patrick Mevzek, and Joseph Yee.

Authors' Addresses

James Gould
VeriSign, Inc.
12061 Bluemont Way
Reston, VA 20190
United States of America

Email: jgould@verisign.com
URI: http://www.verisign.com

Matthew Pozun
VeriSign, Inc.
12061 Bluemont Way
Reston, VA 20190
United States of America

Email: mpozun@verisign.com
URI: http://www.verisign.com