Login Security Policy Extensions Mapping for the Extensible Provisioning Protocol (EPP)
draft-gould-regext-login-security-policy-03

Abstract

This document describes an Extensible Provisioning Protocol (EPP) extension of the Registry Mapping to define the server policy of the Login Security EPP extension. The server policy of the Login Security EPP extension includes the MAYs, SHOULDs, and options implemented by the server.

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 February 3, 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. Object Attributes ......................................... 3
   2.1. Dates and Times ...................................... 3
   2.2. Event Types ........................................ 4
   2.3. System Object ....................................... 4
3. EPP Command Mapping ....................................... 8
   3.1. EPP Query Commands .................................. 8
      3.1.1. EPP <check> Command ............................... 8
      3.1.2. EPP <info> Command ............................... 8
      3.1.3. EPP <transfer> Query Command ................... 10
   3.2. EPP Transform Commands ............................... 10
      3.2.1. EPP <create> Command ......................... 10
      3.2.2. EPP <delete> Command ........................... 10
      3.2.3. EPP <renew> Command ............................. 10
      3.2.4. EPP <transfer> Command ......................... 10
      3.2.5. EPP <update> Command ........................... 10
4. Formal Syntax ............................................ 10
   4.1. Login Security Policy Schema ....................... 11
5. IANA Considerations ....................................... 13
   5.1. XML Namespace ....................................... 13
   5.2. EPP Extension Registry .............................. 14
6. Implementation Status ..................................... 14
   6.1. Verisign EPP SDK .................................. 15
7. Security Considerations ................................... 15
8. Acknowledgements ......................................... 16
9. References ................................................. 16
   9.1. Normative References ................................ 16
   9.2. Informative References ............................. 17
   9.3. URIs ................................................. 17
Appendix A. Change History .................................... 17
   A.1. Change from 00 to 01 ................................ 17
   A.2. Change from 01 to 02 ................................ 17
   A.3. Change from 02 to 03 ................................ 17
Author’s Address ............................................. 18

1. Introduction

This document describes an extension mapping for version 1.0 of the Extensible Provisioning Protocol (EPP) [RFC5730]. This document describes an extension of the Registry Mapping [I-D.gould-carney-regext-registry] to define the server policy of the
Login Security EPP extension [I-D.gould-regext-login-security] for a registry system.

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.

The XML namespace prefix "loginSecPolicy" is used for the namespace "urn:ietf:params:xml:ns:epp:loginSecPolicy-0.4", but implementations MUST NOT depend on it and instead employ a proper namespace-aware XML parser to interpret and output the XML documents.

The XML namespace prefix "loginSec" is used for the namespace "urn:ietf:params:xml:ns:loginSec-0.1", as defined in [I-D.gould-regext-login-security], but implementations MUST NOT depend on it and instead employ a proper namespace-aware XML parser to interpret and output the XML documents.

2. Object Attributes

An EPP login security policy has attributes and associated values that may be viewed and modified by the sponsoring client or the server. This section describes each attribute type in detail. The formal syntax for the attribute values described here can be found in the "Formal Syntax" section of this document and in the appropriate normative references.

2.1. 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 XML Schema Part 2 [1] 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.

2.2. Event Types

The <loginSecPolicy:event> element has a REQUIRED "type" attribute and an OPTIONAL "name" attribute that defines the security event type. The "name" attribute is used to define a sub-type or the type name when the "type" attribute is "custom". The enumerated list of "type" values include:

"password": Identifies a password expiry event policy.
"certificate": Identifies a client certificate expiry event policy.
"cipher": Identifies an insecure or deprecated TLS cipher suite event policy.
"tlsProtocol": Identifies an insecure or deprecated TLS protocol event policy.
"newPW": Identifies the new password complexity requirements event policy.
"stat": Identifies the login security statistical warning event policy. The "name" attribute defines the statistic.
"custom": Identifies a custom event type that MUST set the "name" attribute with the custom event type name.

2.3. System Object

The System object, represented by the <registry:system> element in the Registry Mapping [I-D.gould-carney-regext-registry], is the object that is extended by this extension with the <loginSecPolicy:system> element. The <loginSecPolicy:system> element contains the following child elements:

<loginSecPolicy:pw>: The login password format policy. The <loginSecPolicy:pw> element contains the following child elements:

<loginSecPolicy:expression>: The login password format regular expression. The regular expression MUST conform to the Perl-compatible Regular Expression (PCRE) [pcre] syntax. Programming languages support different sets of PCRE features, so the server SHOULD define a PCRE that leverages features that are supported by a broad set of client programming languages.
<loginSecPolicy:description>: The OPTIONAL human readable description of the login password format policy. The
"lang" attribute MAY be present to identify the language of the description if the negotiated value is something other than the default value of "en" (English).

<loginSecPolicy:specialRules>: The OPTIONAL boolean element, with a default value of "false", that indicates additional special format rules apply that cannot be represented in the regular expression. A value of "1" (or "true") means that special format rules do apply that MUST be described in the <loginSecPolicy:description> element for manual review. The server SHOULD represent the most specific regular expression possible with the <loginSecPolicy:expression> element. A value of "0" (or "false") means that the <loginSecPolicy:expression> element fully represents the format requirements.

<loginSecPolicy:restrictedWords>: The OPTIONAL boolean element, with a default value of "false", that indicates that the server has a list of restricted words that cannot be used in the password. The optional "url" attribute references a description of the list of restricted words.

<loginSecPolicy:userAgentSupport>: OPTIONAL boolean value that indicates the server supports the <loginSec:userAgent> element, with the default value of "0" (or "false"). A value of "1" (or "true") means that the server processes the <loginSec:userAgent> element. A value of "0" (or "false") means that the server ignores the <loginSec:userAgent> element if passed by the client.

<loginSecPolicy:event>: Zero or more <loginSecPolicy:event> elements that defines the policies associated with the supported security events. The required "type" attribute and the OPTIONAL "name" attribute defines the event type, as described in Section 2.2. The <loginSecPolicy:event> element contains the following child elements:

<loginSecPolicy:level>: One or two <loginSecPolicy:level> elements that indicate the possible set of event levels ("warning" or "error") the server will return to the client for the event type.

<loginSecPolicy:exDate>: OPTIONAL boolean element that indicates whether the event type includes a <loginSec:exDate> element with the default value of "0" (or "false").

<loginSecPolicy:exPeriod>: OPTIONAL duration element that the event type must be reset. For example, the password will expire 30 days after being set.

<loginSecPolicy:warningPeriod>: OPTIONAL duration element that indicates how long prior to expiry the server will include a warning event. For example, the server will
include a password expiry warning event 15 days prior to expiry.

<loginSecPolicy:errorAction>: OPTIONAL indication of what action will occur with an error, including at expiry when <loginSecPolicy:exDate> is "1" (or "true"). The possible <loginSecPolicy:errorAction> values include:

"connect": The client connection will fail. For example, when the client certificate expires, the TLS handshake will fail.
"login": The client login will fail. For example, when the new password does not meet the server password complexity requirements or when the password expires, the login will fail.
"none": There is no predefined failure action. For example, when the password expires, the server will not fail the login.

<loginSecPolicy:threshold>: OPTIONAL threshold value that triggers a warning event for a specific "stat" event. For example, a "failedLogins" "stat" warning event will occur if the number of failed logins exceeds 100.

<loginSecPolicy:period>: OPTIONAL period value that is associated with a warning event for a specific "stat" event. For example, a "failedLogins" "stat" warning event will occur if the number of failed logins exceeds the <loginSecPolicy:threshold> value over a period of 1 day.

Example of a <loginSecPolicy:system> element that defines the password policy and the policy of each of the supported login security events:

<loginSecPolicy:system>
  <loginSecPolicy:pw>
    <loginSecPolicy:expression>
      (?=.*\d)
      (?=.*[a-zA-Z])
      (?=.*[!\x21-\x2F\x3A-\x40\x5B-\x7E])
      (?!.^\s+)
      (?!.^\s{2,})
      ^[\x20-\x7e]{16,32}$
    </loginSecPolicy:expression>
    <loginSecPolicy:description>
      16 to 32 printable characters (alphanumeric, space, and special characters) with at least one number, letter, and special character, with no leading or trailing whitespace, and with no consecutive spaces.
<loginSecPolicy:description>
</loginSecPolicy:description>
<loginSecPolicy:specialRules>false
</loginSecPolicy:specialRules>
<loginSecPolicy:restrictedWords
    url="http://example.com/restrictedwords">
    true
</loginSecPolicy:restrictedWords>
</loginSecPolicy:pw>
<loginSecPolicy:userAgentSupport>true
</loginSecPolicy:userAgentSupport>
<loginSecPolicy:event type="newPW">
    <loginSecPolicy:level>error
</loginSecPolicy:level>
    <loginSecPolicy:errorAction>login
</loginSecPolicy:errorAction>
</loginSecPolicy:event>
<loginSecPolicy:event type="password">
    <loginSecPolicy:level>warning
</loginSecPolicy:level>
    <loginSecPolicy:level>error
</loginSecPolicy:level>
    <loginSecPolicy:exDate>true
</loginSecPolicy:exDate>
    <loginSecPolicy:exPeriod>P90D
</loginSecPolicy:exPeriod>
    <loginSecPolicy:warningPeriod>P15D
</loginSecPolicy:warningPeriod>
    <loginSecPolicy:errorAction>login
</loginSecPolicy:errorAction>
</loginSecPolicy:event>
<loginSecPolicy:event type="certificate">
    <loginSecPolicy:level>warning
</loginSecPolicy:level>
    <loginSecPolicy:level>error
</loginSecPolicy:level>
    <loginSecPolicy:exDate>true
</loginSecPolicy:exDate>
    <loginSecPolicy:exPeriod>P90D
</loginSecPolicy:exPeriod>
    <loginSecPolicy:warningPeriod>P15D
</loginSecPolicy:warningPeriod>
    <loginSecPolicy:errorAction>connect
</loginSecPolicy:errorAction>
</loginSecPolicy:event>
<loginSecPolicy:event type="cipher">
    <loginSecPolicy:level>warning
</loginSecPolicy:level>
    <loginSecPolicy:exDate>false
</loginSecPolicy:exDate>
</loginSecPolicy:event>
3. EPP Command Mapping

A detailed description of the EPP syntax and semantics can be found in the EPP core protocol specification [RFC5730]. The command mappings described here are specifically for use in provisioning and managing login security policy via EPP.

3.1. EPP Query Commands

EPP [RFC5730] provides three commands to retrieve object information: <check> to determine if an object is known to the server, <info> to retrieve detailed information associated with an object, and <transfer> to retrieve object transfer status information.

3.1.1. EPP <check> Command

This extension does not define any extension of the EPP <check> command or response described in the Registry Mapping.

3.1.2. EPP <info> Command

This extension does not add any elements to the EPP <info> command described in the Registry Mapping [I-D.gould-carney-regext-registry].
However, additional elements are defined for the <info> response to a query for the registry system attributes.

When an <info> command has been processed successfully, the EPP <resData> element MUST contain a child elements as described in the Registry Mapping [I-D.gould-carney-regext-registry]. In addition, the EPP <extension> element SHOULD contain a child <lp:infData> element that identifies the extension namespace if the system object has data associated with this extension and based on server policy. The <loginSecPolicy:infData> element contains the following child elements:

<lp:system>: Element that contains the full set of login security policy attributes for the system as defined in Section 2.3.

Example <info> response to query for the registry system attributes including the login security policy attributes:

```
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:     <resData>
S:       <registry:infData
S:         xmlns:registry="urn:ietf:params:xml:ns:registry-0.1">
S:         ...
S:       </registry:infData>
S:     </resData>
S:     <extension>
S:       <loginSecPolicy:infData
S:         xmlns:loginSecPolicy="urn:ietf:params:xml:ns:epp:loginSecPolicy-0.4">
S:         ...
S:       </loginSecPolicy:infData>
S:     </extension>
S:   </response>
S: </epp>
```
3.1.3. EPP <transfer> Query Command

Transfer semantics do not directly apply to system objects, so there is no extension defined for the EPP <transfer> query command.

3.2. EPP Transform Commands

EPP provides five commands to transform objects: <create> to create an instance of an object, <delete> to delete an instance of an object, <renew> to extend the validity period of an object, <transfer> to manage object sponsorship changes, and <update> to change information associated with an object.

3.2.1. EPP <create> Command

This extension does not add any elements to the EPP <create> command or <create> response described in the Registry Mapping.

3.2.2. EPP <delete> Command

This extension does not add any elements to the EPP <delete> command or <delete> response described in the Registry Mapping.

3.2.3. EPP <renew> Command

Renew semantics do not directly apply to system objects, so there is no extension defined for the EPP <renew> command.

3.2.4. EPP <transfer> Command

Transfer semantics do not directly apply to system objects, so there is no extension defined for the EPP <transfer> command.

3.2.5. EPP <update> Command

This extension does not add any elements to the EPP <update> command or <update> response described in the Registry Mapping.

4. Formal Syntax

One schema presented here is the EPP Login Security Policy 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.
4.1. Login Security Policy Schema

BEGIN
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
   xmlns:loginSecPolicy=
     "urn:ietf:params:xml:ns:epp:loginSecPolicy-0.4"
   targetNamespace="urn:ietf:params:xml:ns:epp:loginSecPolicy-0.4"
   elementFormDefault="qualified">
   <annotation>
     <documentation>Extensible Provisioning Protocol v1.0
     Login Security Policy Extension Schema.</documentation>
   </annotation>
   <!-- Child response elements. -->
   <element name="infData"
     type="loginSecPolicy:systemContainerType" />
   <!-- Container for system login security policy -->
   <complexType name="systemContainerType">
     <sequence>
       <element name="system"
         type="loginSecPolicy:systemType" />
     </sequence>
   </complexType>
   <!-- System login security policies -->
   <complexType name="systemType">
     <sequence>
       <element name="pw"
         type="loginSecPolicy:pwType" />
       <element name="userAgentSupport"
         type="boolean" minOccurs="0" default="false" />
       <element name="event"
         type="loginSecPolicy:eventType"
         minOccurs="0" maxOccurs="unbounded" />
     </sequence>
   </complexType>
   <complexType name="pwType">
     <sequence>
       <element name="expression"
         type="string" />
       <element name="description" minOccurs="0">
         <complexType>
           <simpleContent>

RESERVED

</documentation>
   </annotation>
   <!-- Child response elements. -->
   <element name="infData"
     type="loginSecPolicy:systemContainerType" />
   <!-- Container for system login security policy -->
   <complexType name="systemContainerType">
     <sequence>
       <element name="system"
         type="loginSecPolicy:systemType" />
     </sequence>
   </complexType>
   <!-- System login security policies -->
   <complexType name="systemType">
     <sequence>
       <element name="pw"
         type="loginSecPolicy:pwType" />
       <element name="userAgentSupport"
         type="boolean" minOccurs="0" default="false" />
       <element name="event"
         type="loginSecPolicy:eventType"
         minOccurs="0" maxOccurs="unbounded" />
     </sequence>
   </complexType>
   <complexType name="pwType">
     <sequence>
       <element name="expression"
         type="string" />
       <element name="description" minOccurs="0">
         <complexType>
           <simpleContent>
<extension base="normalizedString">
    <attribute name="lang" type="language" default="en" />
</extension>
</complexType>
</element>
<element name="specialRules"
type="boolean" default="false" minOccurs="0"/>
<element name="restrictedWords"
type="loginSecPolicy:restrictedWordsType" default="false"
    minOccurs="0"/>
</sequence>
</complexType>
<complexType name="restrictedWordsType">
    <simpleContent>
        <extension base="boolean">
            <attribute name="url" type="token"/>
        </extension>
    </simpleContent>
</complexType>
<complexType name="eventType">
    <sequence>
        <element name="level"
type="loginSecPolicy:levelEnum" minOccurs="1" maxOccurs="2" />
        <element name="exDate"
type="boolean" minOccurs="0" default="false" />
        <element name="exPeriod"
type="duration" minOccurs="0" />
        <element name="warningPeriod"
type="duration" minOccurs="0" />
        <element name="errorAction"
            type="loginSecPolicy:errorActionType" minOccurs="0" />
        <element name="threshold"
type="integer" minOccurs="0" />
        <element name="period"
type="duration" minOccurs="0" />
    </sequence>
    <attribute name="type"
type="loginSecPolicy:typeEnum" use="required" />
    <attribute name="name" type="token" />

\[</complexType>\]
\[<!--
Enumerated list of event types, with extensibility via "custom".
-->\]
\[<simpleType name="typeEnum">
  <restriction base="token">
    <enumeration value="password" />
    <enumeration value="certificate" />
    <enumeration value="cipher" />
    <enumeration value="tlsProtocol" />
    <enumeration value="newPW" />
    <enumeration value="stat" />
    <enumeration value="custom" />
  </restriction>
</simpleType>\]
\[<!--
Enumerated list of levels.
-->\]
\[<simpleType name="levelEnum">
  <restriction base="token">
    <enumeration value="warning" />
    <enumeration value="error" />
  </restriction>
</simpleType>\]
\[<!--
Enumerated list of error actions
-->\]
\[<simpleType name="errorActionType">
  <restriction base="TokenType">
    <enumeration value="connect" />
    <enumeration value="login" />
    <enumeration value="none" />
  </restriction>
</simpleType>\]
\[<!--
End of schema.
-->\]
\[</schema>\]

5. IANA Considerations

5.1. XML Namespace

This document uses URNs to describe XML namespaces and XML schemas
conforming to a registry mechanism described in [RFC3688].

Registration request for the login security policy namespace:
5.2. 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 Policy Extensions Mapping 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

6. 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".

6.1. 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-gould-regext-login-security-policy.

Level of maturity: Development

Coverage: All aspects of the protocol are implemented.

Licensing: GNU Lesser General Public License

Contact: jgould@verisign.com


7. Security Considerations

The mapping extensions described in this document provide additional security services beyond those described by EPP [RFC5730] and protocol layers used by EPP. The security considerations described in these other specifications apply to this specification as well.

This mapping does define the login security policy of the server, where there are security considerations with defining the policy, which include:

1. The server SHOULD follow login password complexity best practices, such as the NIST Special Publication 800-63B [2].
2. The server MAY have a login password expiry that requires the client to regularly change the login password.
3. The server SHOULD inform the client of the expiry of the login password.
4. The server MUST store the login password at rest securely, such as hashing or encrypting the login password.
5. The server SHOULD deprecate and eliminate insecure ciphers and protocols.
6. The server SHOULD inform the client of the use of insecure ciphers and protocols.
7. The server SHOULD inform the client of an expiring client certificate.

8. Acknowledgements

TBD

9. References

9.1. Normative References

[I-D.gould-carney-regext-registry]

[I-D.gould-regext-login-security]


9.2. Informative References


9.3. URIs


Appendix A. Change History

A.1. Change from 00 to 01

1. Changed the <loginSecPolicy:exError> element to the <loginSecPolicy:errorAction> element, to be more generic to identify the action taken for an error level event for any event type. The XML namespace was changed to "urn:ietf:params:xml:ns:epp:loginSecPolicy-0.3" based on the change to the XML schema.

2. Updated the Implementation Status section.

A.2. Change from 01 to 02

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:epp:loginSecPolicy-0.3".

A.3. Change from 02 to 03

1. Added support for the <loginSecPolicy:specialRules> and <loginSecPolicy:restrictedWords> elements in the <loginSecPolicy:pw> element to support cases where the server has special format rules or restricts certain words for the password. Adding the new elements resulted in updating the XML schema to "urn:ietf:params:xml:ns:epp:loginSecPolicy-0.4".