Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: November 24, 2016

J. Gould
VeriSign, Inc.
S. Wodjenski
Neustar

May 23, 2016

Allocation Token Extension for the Extensible Provisioning Protocol (EPP)
draft-gould-allocation-token-04

Abstract

This document describes an Extensible Provisioning Protocol (EPP) extension for including an allocation token or code for allocating an object like a domain name to the client. The allocation token MAY be transferred out-of-band to a client to give them authorization to allocate an object using one of the EPP transform commands including create, update, and transfer.

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 http://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 November 24, 2016.

Copyright Notice

Copyright (c) 2016 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 (http://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. Allocation Token 3
3. EPP Command Mapping 4
3.1. EPP Query Commands 4
3.1.1. EPP <check> Command 4
3.1.2. EPP <info> Command 8
3.1.3. EPP <transfer> Command 10
3.2. EPP Transform Commands 11
3.2.1. EPP <create> Command 11
3.2.2. EPP <delete> Command 12
3.2.3. EPP <renew> Command 12
3.2.4. EPP <transfer> Command 12
3.2.5. EPP <update> Command 13
4. Formal Syntax 14
4.1. Allocation Token Extension Schema 15
5. IANA Considerations 15
5.1. XML Namespace 15
5.2. EPP Extension Registry 16
6. Security Considerations 16
7. Acknowledgements 16
8. Normative References 16
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
A.4. Change from 03 to 04 17
Authors' Addresses 17

1. Introduction

This document describes an extension mapping for version 1.0 of the Extensible Provisioning Protocol (EPP) [RFC5730]. This mapping, an extension to EPP object mappings like the EPP domain name mapping [RFC5731], for passing an allocation token one of the EPP transform commands including create, update, and transfer. The allocation token is known to the server to authorize a client that passes a matching allocation token with one of the supported EPP transform commands. It is up to server policy which EPP transform commands and which objects support the allocation token. The allocation token MAY

be returned to an authorized client for passing out-of-band to a client that uses it with an EPP transform command.

1.1. Conventions Used in This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

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.

"allocationToken-1.0" is used as an abbreviation for "urn:ietf:params:xml:ns:allocationToken-1.0". The XML namespace prefix "allocationToken" 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. Object Attributes

This extension adds additional elements to EPP object mappings like the EPP domain name mapping [RFC5731]. Only those new elements are described here.

2.1. Allocation Token

The Allocation Token is a simple XML "token" type. The exact format of the Allocation Token is up to server policy. The server MUST have the allocation token for each object to match against the allocation token passed by the client to authorize the allocation of the object. The same <allocationToken:allocationToken> element is used for all of the supported EPP transform commands as well as the info response. If an invalid allocation token is passed the server MUST return an EPP error result code of 2201.

An example <allocationToken:allocationToken> element with value of "abc123":

```
<allocationToken:allocationToken xmlns:allocationToken=
          "urn:ietf:params:xml:ns:allocationToken-1.0">
  abc123
</allocationToken:allocationToken>
```

3. EPP Command Mapping

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

3.1. EPP Query Commands

EPP 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 defines additional elements to extend the EPP <check> command of an object mapping like [RFC5731].

This extension allow clients to check the availability of an object with an allocation token, as described in Section 2.1. Clients can check if an object can be created using the allocation token. The allocation token is applied to all object names included in the EPP <check> command.

Example <check> command for the example.tld domain name using the <allocationToken:allocationToken> extension with the allocation token of 'abc123':

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <check>
C:      <domain:check
C:       xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
C:        <domain:name>example.tld</domain:name>
C:      </domain:check>
C:    </check>
C:    <extension>
C:      <allocationToken:allocationToken
C:        xmlns:allocationToken=
C:          "urn:ietf:params:xml:ns:allocationToken-1.0">
C:        abc123
C:      </allocationToken:allocationToken>
C:    </extension>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

If the query was successful, the server replies with an <check> response providing availability status of queried object.

Example <check> domain response for a <check> command using the <allocationToken:allocationToken> extension:

```
S:<?xml version="1.0" encoding="UTF-8"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S: <response>
S:  <result code="1000">
S:   <msg lang="en-US">Command completed successfully</msg>
S:  </result>
S:  <resData>
S:   <domain:chkData
S:     xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
S:    <domain:cd>
S:     <domain:name avail="0">example.tld</domain:name>
S:     <domain:reason>Invalid domain-token pair</domain:reason>
S:    </domain:cd>
S:   </domain:chkData>
S:  </resData>
S:  <trID>
S:   <clTRID>ABC-DEF-12345</clTRID>
S:   <svTRID>54321-XYZ</svTRID>
S:  </trID>
S: </response>
S:</epp>
```

Example <check> command with the <allocationToken:allocationToken> extension for the example.tld and example2.tld domain names. Availability of example.tld and example2.tld domain names are based on the allocation token 'abc123':

```
C:<?xml version="1.0" encoding="UTF-8"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C: <command>
C:  <check>
C:   <domain:check
C:     xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
C:    <domain:name>example.tld</domain:name>
C:    <domain:name>example2.tld</domain:name>
C:   </domain:check>
C:  </check>
C:  <extension>
C:   <allocationToken:allocationToken
C:     xmlns:allocationToken=
C:       "urn:ietf:params:xml:ns:allocationToken-1.0">
C:     abc123
C:   </allocationToken:allocationToken>
C:  </extension>
C:  <clTRID>ABC-DEF-12345</clTRID>
C: </command>
C:</epp>
```

Example <check> domain response for multiple domain names in the <check> command using the <allocationToken:allocationToken> extension:

```
S:<?xml version="1.0" encoding="UTF-8"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S: <response>
S:  <result code="1000">
S:   <msg lang="en-US">Command completed successfully</msg>
S:  </result>
S:  <resData>
S:   <domain:chkData
S:     xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
S:    <domain:cd>
S:     <domain:name avail="0">example.tld</domain:name>
S:     <domain:reason>Invalid domain-token pair</domain:reason>
S:    </domain:cd>
S:    <domain:cd>
S:     <domain:name avail="1">example2.tld</domain:name>
S:    </domain:cd>
S:   </domain:chkData>
S:  </resData>
S:  <trID>
S:   <clTRID>ABC-DEF-12345</clTRID>
S:   <svTRID>54321-XYZ</svTRID>
S:  </trID>
S: </response>
S:</epp>
```

This extension does not add any elements to the EPP <check> response described in the [RFC5730].

3.1.2. EPP <info> Command

This extension defines additional elements to extend the EPP <info> command of an object mapping like [RFC5731].

The EPP <info> command allows a client to request information on an existing object. Authorized clients MAY retrieve the allocation token (Section 2.1) along with the other object information using the <allocationToken:info> element that identifies the extension namespace. The <allocationToken:info> element is an empty element that serves as a marker to the server to return the <allocationToken:allocationToken> element, defined in Section 2.1, in the info response. If the client is not authorized to receive the allocation token (Section 2.1), the server MUST return an EPP error result code of 2201. If the client is authorized to receive the

allocation token (Section 2.1), but there is no allocation token (Section 2.1) associated with the object, the server MUST return an EPP error result code of 2303 object referencing the <allocationToken:info> element.

Example <info> command with the allocationToken:info extension for the example.tld domain name:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:   <info>
C:    <domain:info
C:      xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
C:      xsi:schemaLocation="urn:ietf:params:xml:ns:domain-1.0
C:      domain-1.0.xsd">
C:      <domain:name>example.tld</domain:name>
C:    </domain:info>
C:   </info>
C:   <extension>
C:      <allocationToken:info
C:        xmlns:allocationToken=
C:          "urn:ietf:params:xml:ns:allocationToken-1.0/>
C:   </extension>
C:   <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

If the query was successful, the server replies with an <allocationToken:allocationToken> element, as described in Section 2.1.

Example <info> domain response using the
<allocationToken:allocationToken> extension:

```
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:      <domain:infData
S:       xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
S:        <domain:name>example.tld</domain:name>
S:        <domain:roid>EXAMPLE1-REP</domain:roid>
S:        <domain:status s="pendingCreate"/>
S:        <domain:registrant>jd1234</domain:registrant>
S:        <domain:contact type="admin">sh8013</domain:contact>
S:        <domain:contact type="tech">sh8013</domain:contact>
S:        <domain:clID>ClientX</domain:clID>
S:        <domain:crID>ClientY</domain:crID>
S:        <domain:crDate>2012-04-03T22:00:00.0Z</domain:crDate>
S:        <domain:authInfo>
S:          <domain:pw>2fooBAR</domain:pw>
S:        </domain:authInfo>
S:      </domain:infData>
S:    </resData>
S:    <extension>
S:      <allocationToken:allocationToken
S:        xmlns:allocationToken=
S:          "urn:ietf:params:xml:ns:allocationToken-1.0">
S:        abc123
S:      </allocationToken:allocationToken>
S:    </extension>
S:    <trID>
S:      <clTRID>ABC-12345</clTRID>
S:      <svTRID>54321-XYZ</svTRID>
S:    </trID>
S:  </response>
S:</epp>
```

3.1.3. EPP <transfer> Command

This extension does not add any elements to the EPP <transfer> query command or <transfer> response described in the [RFC5730].

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 defines additional elements to extend the EPP <create> command of an object mapping like [RFC5731].

The EPP <create> command provides a transform operation that allows a client to create an object. In addition to the EPP command elements described in an object mapping like [RFC5731], the command MUST contain a child <allocationToken:allocationToken> element, as defined in Section 2.1, that identifies the extension namespace for the client to be authorized to create and allocate the object. If the allocation token (Section 2.1) does not match the object's allocation token (Section 2.1), the server MUST return an EPP error result code of 2201.:

Example <create> command to create a domain object with an allocation token:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <create>
C:      <domain:create
C:       xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
C:        <domain:name>example.tld</domain:name>
C:        <domain:registrant>jd1234</domain:registrant>
C:        <domain:contact type="admin">sh8013</domain:contact>
C:        <domain:contact type="tech">sh8013</domain:contact>
C:        <domain:authInfo>
C:          <domain:pw>2fooBAR</domain:pw>
C:        </domain:authInfo>
C:      </domain:create>
C:    </create>
C:    <extension>
C:      <allocationToken:allocationToken
C:        xmlns:allocationToken=
C:          "urn:ietf:params:xml:ns:allocationToken-1.0">
C:        abc123
C:      </allocationToken:allocationToken>
C:    </extension>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

This extension does not add any elements to the EPP <create> response described in the [RFC5730].

3.2.2. EPP <delete> Command

This extension does not add any elements to the EPP <delete> command or <delete> response described in the [RFC5730].

3.2.3. EPP <renew> Command

This extension does not add any elements to the EPP <renew> command or <renew> response described in the [RFC5730].

3.2.4. EPP <transfer> Command

This extension defines additional elements to extend the EPP <transfer> request command of an object mapping like [RFC5731].

The EPP <transfer> request command provides a transform operation that allows a client to request the transfer of an object. In addition to the EPP command elements described in an object mapping like [RFC5731], the command MUST contain a child <allocationToken:allocationToken> element, as defined in Section 2.1, that identifies the extension namespace for the client to be authorized to transfer and allocate the object. If the allocation token (Section 2.1) does not match the object's allocation token (Section 2.1), the server MUST return an EPP error result code of 2201.:

Example <transfer> request command to allocate the domain object with the allocation token:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <transfer op="request">
C:      <domain:transfer
C:        xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
C:        <domain:name>example1.tld</domain:name>
C:        <domain:period unit="y">1</domain:period>
C:        <domain:authInfo>
C:          <domain:pw>2fooBAR</domain:pw>
C:        </domain:authInfo>
C:      </domain:transfer>
C:    </transfer>
C:    <extension>
C:      <allocationToken:allocationToken
C:        xmlns:allocationToken=
C:          "urn:ietf:params:xml:ns:allocationToken-1.0">
C:        abc123
C:      </allocationToken:allocationToken>
C:    </extension>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

This extension does not add any elements to the EPP <transfer> response described in the [RFC5730].

3.2.5. EPP <update> Command

This extension defines additional elements to extend an extension of an empty EPP <update> command of an object mapping like [RFC5731]. An example of an extension of an empty EPP <update> command is the definition of the restore command within [RFC3915].

An extension of an empty EPP <update> command defines a new verb that transforms an object. In addition to the EPP command elements described in an object mapping like [RFC5731], the command MUST contain a child <allocationToken:allocationToken> element, as defined in Section 2.1, that identifies the extension namespace for the client to be authorized to allocate the object. If the allocation token (Section 2.1) does not match the object's allocation token (Section 2.1), the server MUST return an EPP error result code of 2201.:

Example use an extension of an empty <update> command to release a domain object with an allocation token:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <update>
C:      <domain:update
C:        xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
C:        <domain:name>example1.tld</domain:name>
C:      </domain:update>
C:    </update>
C:    <extension>
C:      <release:release
C:        xmlns:release="urn:ietf:params:xml:ns:release-1.0"/>
C:      <allocationToken:allocationToken
C:        xmlns:allocationToken=
C:          "urn:ietf:params:xml:ns:allocationToken-1.0">
C:        abc123
C:      </allocationToken:allocationToken>
C:    </extension>
C:    <clTRID>ABC-12345-XYZ</clTRID>
C:  </command>
C:</epp>
```

This extension does not add any elements to the EPP <update> response described in the [RFC5730].

4. Formal Syntax

One schema is presented here that is the EPP Allocation Token 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.

4.1. Allocation Token Extension Schema

```
BEGIN
<?xml version="1.0" encoding="UTF-8"?>

<schema targetNamespace="urn:ietf:params:xml:ns:allocationToken-1.0"
  xmlns:allocationToken="urn:ietf:params:xml:ns:allocationToken-1.0"
  xmlns="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">

  <annotation>
    <documentation>
      Extensible Provisioning Protocol v1.0
      Allocation Token Extension.
    </documentation>
  </annotation>

  <!-- Element used in info command to get allocation token. -->
  <element name="info"/>

  <!-- Allocation Token used in transform
  commands and info response -->
  <element name="allocationToken"
    type="allocationToken:allocationTokenType"/>

   <complexType name="allocationTokenType">
    <simpleContent>
      <extension base="token"/>
    </simpleContent>
  </complexType>

   <!-- End of schema.-->
</schema>
END
```

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]. The following URI assignment is requested of IANA:

URI: ietf:params:xml:ns:allocationToken-1.0

Registrant Contact: See the "Author's Address" section of this document.

XML: See the "Formal Syntax" section of this document.

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: "Allocation Token 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

6. Security Considerations

The mapping extensions described in this document do not provide any 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.

7. Acknowledgements

The authors wish to acknowledge the original concept for this draft and the efforts in the initial versions of this draft by Trung Tran.

8. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, January 2004.

[RFC3915] Hollenbeck, S., "Domain Registry Grace Period Mapping for the Extensible Provisioning Protocol (EPP)", RFC 3915, September 2004.

[RFC5730] Hollenbeck, S., "Extensible Provisioning Protocol (EPP)", STD 69, RFC 5730, August 2009.

[RFC5731] Hollenbeck, S., "Extensible Provisioning Protocol (EPP) Domain Name Mapping", STD 69, RFC 5731, August 2009.

[RFC7451] Hollenbeck, S., "Extension Registry for the Extensible Provisioning Protocol", RFC 7451, February 2015.

Appendix A. Change History

A.1. Change from 00 to 01

1. Amended XML Namespace section of IANA Considerations, added EPP Extension Registry section.
2. Moved Change History to the back section as an Appendix.

A.2. Change from 01 to 02

1. Ping update.

A.3. Change from 02 to 03

1. Ping update.

A.4. Change from 03 to 04

1. Updated the authors for the draft.

Authors' Addresses

James Gould
VeriSign, Inc.
12061 Bluemont Way
Reston, VA 20190
US

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

Sharon Wodjenski
Neustar
21575 Ridgetop Circle
Sterling, VA 20166
US

Email: Sharon.Wodjenski@neustar.biz
URI: http://www.neustar.biz