Internet Engineering Task Force (IETF)
Request for Comments: 8495
Category: Standards Track
ISSN: 2070-1721

J. Gould
VeriSign, Inc.
K. Feher
Neustar

November 2018

Allocation Token Extension for the Extensible Provisioning Protocol (EPP)

Abstract

This document describes an Extensible Provisioning Protocol (EPP) extension for including an Allocation Token in "query" and "transform" commands. The Allocation Token is used as a credential that authorizes a client to request the allocation of a specific object from the server using one of the EPP transform commands, including "create" and "transfer".

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc8495.

Table of Contents

1. Introduction

This document describes an extension mapping for version 1.0 of the Extensible Provisioning Protocol (EPP) [RFC5730]. This mapping, which is an extension to EPP object mappings similar to the EPP domain name mapping [RFC5731], supports passing an Allocation Token as a credential that authorizes a client to request the allocation of a specific object from the server using one of the EPP transform commands, including "create" and "transfer".

Allocation is when a server assigns the sponsoring client of an object based on the use of an Allocation Token credential. Examples include allocating a registration based on a pre-eligibility Allocation Token, allocating a premium domain name registration based on an auction Allocation Token, allocating a registration based on a founders Allocation Token, and allocating an existing domain name held by the server or by a different sponsoring client based on an Allocation Token that is passed with a transfer command.

Clients pass an Allocation Token to the server for validation, and the server determines if the supplied Allocation Token is one supported by the server. It is up to server policy which EPP transform commands and which objects require 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", "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 the examples are provided only to illustrate element relationships and are not REQUIRED in the protocol.

The XML namespace prefix "allocationToken" is used for the namespace "urn:ietf:params:xml:ns:allocationToken-1.0", 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.

The "abc123" token value is used as a placeholder value in the examples. The server MUST support token values that follow the Security Considerations (Section 6).

The domain-object attribute values, including the "2fooBAR" <domain:pw> value, in the examples are provided for illustration purposes only. Refer to [RFC5731] for details on the domain-object attributes.

2. Object Attributes

This extension adds additional elements to EPP object mappings similar to 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 MAY 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 <allocationToken:allocationToken> element is used for all of the supported EPP commands as well as the info response. If the supplied Allocation Token passed to the server does not apply to the object, the server MUST return an EPP error result code of 2201.

Authorization information, similar to what is defined in the EPP domain name mapping [RFC5731], is associated with objects to facilitate transfer operations. The authorization information is assigned when an object is created. The Allocation Token and the authorization information are both credentials but are used for different purposes and in different ways. The Allocation Token is used to facilitate the allocation of an object instead of transferring the sponsorship of the object. The Allocation Token is not managed by the client but is validated by the server to authorize assigning the initial sponsoring client of the object.

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 can be provisioned, <info> to retrieve 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 similar to the mapping specified in [RFC5731].

This extension allows 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.

The following is an example <check> command for the allocation.example 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>allocation.example</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 a <check> response providing the availability status of the queried object based on the following Allocation Token cases where the object is otherwise available:

1. If an object requires an Allocation Token and the Allocation Token does apply to the object, then the server MUST return the availability status as available (e.g., the "avail" attribute is "1" or "true").
2. If an object requires an Allocation Token and the Allocation Token does not apply to the object, then the server SHOULD return the availability status as unavailable (e.g., the "avail" attribute is "0" or "false").
3. If an object does not require an Allocation Token, the server MAY return the availability status as available (e.g., the "avail" attribute is "1" or "true").

The following is an 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="1">allocation.example</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>
```

The following is an example <check> command with the <allocationToken:allocationToken> extension for the allocation.example and allocation2.example domain names. Availability of allocation.example and allocation2.example 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>allocation.example</domain:name>
C:    <domain:name>allocation2.example</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>
```

The following is an example <check> domain response for multiple domain names in the <check> command using the <allocationToken:allocationToken> extension, where the Allocation Token 'abc123' matches allocation.example but does not match allocation2.example:

```
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="1">allocation.example</domain:name>
S:    </domain:cd>
S:    <domain:cd>
S:     <domain:name avail="0">allocation2.example</domain:name>
S:     <domain:reason>Allocation Token mismatch</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>
```

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

3.1.2. EPP <info> Command

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

The EPP <info> command allows a client to request information associated with an existing object. Authorized clients MAY retrieve the Allocation Token (Section 2.1) along with the other object information by supplying the <allocationToken:info> element in the command. The <allocationToken:info> element is an empty element that serves as a marker to the server to return the <allocationToken:allocationToken> element in the info response. If the client is not authorized to receive the Allocation Token, the

server MUST return an EPP error result code of 2201. If the client is authorized to receive the Allocation Token, but there is no Allocation Token associated with the object, the server MUST return an EPP error result code of 2303. The authorization is subject to server policy.

The following is an example <info> command with the allocationToken:info extension for the allocation.example 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:      <domain:name>allocation.example</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 along with the regular EPP <resData>. The <allocationToken:allocationToken> element is described in Section 2.1.

The following is an 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>allocation.example</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> Query Command

This extension does not add any elements to the EPP <transfer> query command or <transfer> query response described in [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 similar to the mapping specified in [RFC5731].

The EPP <create> command provides a transform operation that allows a client to create an instance of an object. In addition to the EPP command elements described in an object mapping similar to the mapping specified in [RFC5731], the command MUST contain a child <allocationToken:allocationToken> element for the client to be authorized to create and allocate the object. If the Allocation Token does not apply to the object, the server MUST return an EPP error result code of 2201.

The following is an 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>allocation.example</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 [RFC5730].

3.2.2. EPP <delete> Command

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

3.2.3. EPP <renew> Command

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

3.2.4. EPP <transfer> Command

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

The EPP <transfer> 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 similar to the mapping specified in [RFC5731], the command MUST contain a child <allocationToken:allocationToken> element for the client to be authorized to transfer and allocate the object. The authorization associated with the Allocation Token is in addition to, and does not replace, the authorization mechanism defined for the object's <transfer> command. If the Allocation Token is invalid or not required for the object, the server MUST return an EPP error result code of 2201.

The following is an example <transfer> 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 [RFC5730].

3.2.5. EPP <update> Command

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

4. Formal Syntax

One schema is presented here: 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 xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:allocationToken="urn:ietf:params:xml:ns:allocationToken-1.0"
  targetNamespace="urn:ietf:params:xml:ns:allocationToken-1.0"
  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">
    <complexType>
      <complexContent>
        <restriction base="anyType" />
      </complexContent>
    </complexType>
  </element>

  <!-- Allocation Token used in transform
    commands and info response -->
  <element name="allocationToken"
    type="allocationToken:allocationTokenType" />
  <simpleType name="allocationTokenType">
    <restriction base="token">
      <minLength value="1" />
    </restriction>
  </simpleType>

<!-- 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 allocationToken namespace has been registered as follows.

URI: urn:ietf:params:xml:ns:allocationToken-1.0
Registrant Contact: IESG
XML: None. Namespace URIs do not represent an XML specification.

The allocationToken XML schema has been registered as follows.

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

5.2. EPP Extension Registry

The following entry has been added to the Extensions for the Extensible Provisioning Protocol (EPP) registry, as described in [RFC7451].

Name of Extension: Allocation Token Extension for the Extensible Provisioning Protocol (EPP)

Document Status: Standards Track

Reference: RFC 8495

Registrant: IESG <iesg@ietf.org>

TLDs: Any

IPR Disclosure: None

Status: Active

Notes: None

6. Security Considerations

The mapping described in this document does 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.

The mapping acts as a conduit for the passing of Allocation Tokens between a client and a server. The definition of the Allocation Token SHOULD be defined outside of this mapping. The following are security considerations in the definition and use of an Allocation Token:

1. An Allocation Token should be considered secret information by the client; it SHOULD be protected at rest and MUST be protected in transit.
2. An Allocation Token should be single use, meaning it should be unique per object and per allocation operation.
3. An Allocation Token should have a limited life with some form of expiry in the Allocation Token, if generated by a trusted third party, or with a server-side expiry, if generated by the server.
4. An Allocation Token should use a strong random value if it is based on an unsigned code.
5. An Allocation Token should leverage digital signatures to confirm its authenticity if generated by a trusted third party.
6. An Allocation Token that is signed XML should be encoded (e.g., base64 [RFC4648]) to mitigate server validation issues.

7. References

7.1. 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>.

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

[RFC5731] Hollenbeck, S., "Extensible Provisioning Protocol (EPP) Domain Name Mapping", STD 69, RFC 5731, DOI 10.17487/RFC5731, August 2009, <https://www.rfc-editor.org/info/rfc5731>.

[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>.

7.2. Informative References

[RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, DOI 10.17487/RFC4648, October 2006, <https://www.rfc-editor.org/info/rfc4648>.

[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>.

Acknowledgements

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

Special suggestions that have been incorporated into this document were provided by Ben Campbell, Scott Hollenbeck, Benjamin Kaduk, Mirja Kuehlewind, Rubens Kuhl, Alexander Mayrhofer, Patrick Mevzek, Eric Rescoria, and Adam Roach.

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

Kal Feher
Neustar
lvl 8/10 Queens Road
Melbourne, VIC 3004
Australia

Email: ietf@feherfamily.org
URI: http://www.neustar.biz