Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: July 22, 2013

R. Gieben
M. Groeneweg
R. Ribbers
A.L.J. Verschuren
SIDN Labs

January 20, 2013

Key Relay Mapping for the Extensible Provisioning Protocol
draft-gieben-epp-keyrelay-00

Abstract

This document describes an Extensible Provisioning Protocol (EPP) extension mapping for the purpose of relaying DNSSEC key material from a one registrar to another. The mapping introduces <keyrelay> as a new command in EPP.

This command will help facilitating a transfer of a domain while keeping DNSSEC's chain of trust intact.

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 July 22, 2013.

Copyright Notice

Table of Contents

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 BCP 14, RFC 2119 [RFC2119].

In examples, "C:" represents lines sent by a protocol client, and "S:" represents lines returned by a protocol server. "////" is used to note element values that have been shortened to better fit page boundaries. Indentation and white space in examples is provided only to illustrate element relationships and is not a mandatory feature of this protocol.

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.

The term "key material" denotes one more DNSKEY resource records [RFC4034].

In Section 1.2 of [I-D.koch-dnsop-dnssec-operator-change] the terms "losing DNS operator" and "gaining DNS operator" are defined. With EPP a registry can only talk to its registrars, so in this document we will use the terms "loosing registrar" and "gaining registrar".

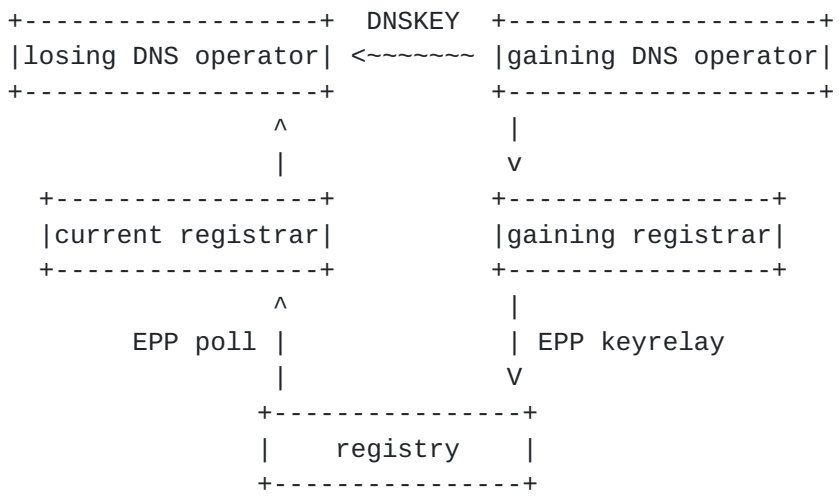
2. Introduction

Certain transactions for DNSSEC signed zones require an authenticated exchange of DNSSEC key material between DNS operators. Often there is no direct secure channel or it is non-scalable.

One of such transactions is changing the DNS operator for DNSSEC signed zones ([I-D.koch-dnsop-dnssec-operator-change]. In this document we define a protocol extension for use in EPP that helps to implement and automate this transaction. This protocol extension introduces a new command called "<keyrelay>".

3. Relaying Key Material

The "<keyrelay>" command uses the existing authenticated EPP channel with the registry. Both registrars can securely talk to the registry and as such the registry can serve as a drop box for relaying key material between them (see Figure 1).



The gaining and losing dns-operators should talk directly to each other (the ~ arrow) to exchange the DNSKEY, but often there is no trusted path between the two. As both can securely interact with the registry through the registrar it can act as a relay for the key material exchange.

Figure 1

The "<keyrelay>" command uploads a new key to the registry. This key material is then relayed to the current registrar's message queue. There is no need for the registry to store the relayed key in the registry system, although the registry MAY save the key for administrative purposes.

The registrar may upload multiple keys in one "<keyrelay>" message. If keys are identical (Flags Field, Protocol Field, Algorithm Field and Public Key Field are equal), the duplicate keys MUST be dropped.

There is no restriction on the type (for instance Key Signing Keys or Zone Signing Keys) of keys that can be put in the message. It is up

to the losing registrar to validate the correctness of the key material.

If for some reason the registry can not process the "<keyrelay>" command an EPP error response MUST be returned. If the registry does process the "<keyrelay>" command it MUST put all (discarding any duplicates) uploaded keys on to the losing registrars' message queue.

4. Rational For a New Command

The keyrelay command is different than the existing EPP commands, because it allows someone to manipulate data without actually being to owner of that data. The EPP transfer command comes close with respect to this functionality. We did not want to overload the transfer command for this purpose, because a keyrelay has nothing to do with that operation.

5. Key Relay Interface

The Key Relay Interface uses a "<keyrelay>" element for relay the key material. It needs a maximum of three elements: a domain name, the key to upload and optionally a token which indicates a future transfer is imminent.

The "<keyrelay>" element MUST contain the following child elements:

- A "<ext:name>" element that contains the domain name for which we upload the key.

- A "<ext:keyData>" element that contains the key material as described in [RFC5910], Section 4.2.

And MAY contain:

- A "<ext:authInfo>" that contains an authorization token ([RFC5931], Section 3.2.4) This can be used as an extra indication that the losing and gaining registrar had prior contact and a possible, future transfer is authorized.

6. Example Key Relay Interface

The following is an example of the "<keyrelay>" command:

```
C:<?xml version="1.0" encoding="UTF-8"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
C:  xmlns:secDNS="urn:ietf:params:xml:ns:secDNS-1.1"
C:  xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
C:  xmlns:ext="urn:ietf:params:xml:ns:keyrelay-1.0">
C:  <extension>
C:     <ext:command>
C:       <ext:keyrelay>
C:         <ext:name>example.org</ext:name>
C:           <ext:authInfo>
C:             <domain:pw>JnSdBAZSxxzJ</domain:pw>
C:           </ext:authInfo>
C:           <ext:keyData>
C:              <secDNS:flags>256</secDNS:flags>
C:              <secDNS:protocol>3</secDNS:protocol>
C:              <secDNS:alg>8</secDNS:alg>
C:              <secDNS:pubKey>AwEAAc////Vesz</secDNS:pubKey>
C:           </ext:keyData>
C:        </ext:keyrelay>
C:     </ext:command>
C:  </extension>
C:</epp>
```

7. Server Reply

Example "<keyrelay>" response:

```
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 succesfully</msg>
S:    </result>
S:    <trID>
S:       <clTRID>ABC-12345</clTRID>
S:       <svTRID>54321-ZYX</svTRID>
S:    </trID>
S:  </response>
S:</epp>
```

As stated an EPP error response MUST be returned if a "<keyrelay>" command can not be processed for any reason.

8. Message Queue Interface

9. Message Queue Format

Example "Key Relay" service message:

```
S:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
S:xmlns:secDNS="urn:ietf:params:xml:ns:secDNS-1.1"
S:xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
S:xmlns:keyrelay="urn:ietf:params:xml:ns:keyrelay-1.0">
S:  <response>
S:     <result code="1301">
S:        <msg>Command completed successfully; ack to dequeue</msg>
S:     </result>
S:     <msgQ count="5" id="12345">
S:         <qDate>1999-04-04T22:01:00.0Z</qDate>
S:         <msg>Key Relay action completed successfully.</msg>
S:     </msgQ>
S:     <resData>
S:        <keyrelay:response>
S:           <keyrelay:panData>
S:               <keyrelay:name paResult="true">example.org
S:                  </keyrelay:name>
S:               <keyrelay:paTRID>
S:                  <clTRID>BCD-23456</clTRID>
S:                  <svTRID>65432-WXY</svTRID>
S:               </keyrelay:paTRID>
S:               <keyrelay:paDate>1999-04-04T22:01:00.0Z
S:                  </keyrelay:paDate>
S:               <keyrelay:authInfo>
S:                  <domain:pw>JnSdBAZSxxzJ</domain:pw>
S:               </keyrelay:authInfo>
S:               <keyrelay:keyData>
S:                  <secDNS:flags>256</secDNS:flags>
S:                  <secDNS:protocol>3</secDNS:protocol>
S:                  <secDNS:alg>8</secDNS:alg>
S:                  <secDNS:pubKey>AwEAAc////Vesz</secDNS:pubKey>
S:               </keyrelay:keyData>
S:           </keyrelay:panData>
S:        </keyrelay:response>
S:     </resData>
S:     <trID>
S:        <clTRID>BCD-23456</clTRID>
S:        <svTRID>65432-WXY</svTRID>
S:     </trID>
S:  </response>
S:</epp>
```

10. Formal Syntax

An EPP object mapping is specified in XML Schema notation. The formal syntax presented here is a complete schema representation of the object mapping suitable for automated validation of EPP XML

instances.

"<keyrelay>" command schema:

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="urn:ietf:params:xml:ns:keyrelay-1.0"
    xmlns:keyrelay="urn:ietf:params:xml:ns:keyrelay-1.0"
    xmlns:secDNS="urn:ietf:params:xml:ns:secDNS-1.1"
    xmlns:epp="urn:ietf:params:xml:ns:epp-1.0"
    xmlns:eppcom="urn:ietf:params:xml:ns:eppcom-1.0"
    xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
    xmlns="http://www.w3.org/2001/XMLSchema"
        elementFormDefault="qualified">

    <annotation>
      <documentation>
        Extensible Provisioning Protocol v1.0 domain name
        extension schema for relaying key material.
      </documentation>
    </annotation>

    <import namespace="urn:ietf:params:xml:ns:epp-1.0"
       schemaLocation="epp-1.0.xsd" />
    <import namespace="urn:ietf:params:xml:ns:eppcom-1.0"
       schemaLocation="eppcom-1.0.xsd" />
    <import namespace="urn:ietf:params:xml:ns:secDNS-1.1"
       schemaLocation="secdns-1.1.xsd" />
    <import namespace="urn:ietf:params:xml:ns:domain-1.0"
       schemaLocation="domain-1.0.xsd" />

    <element name="command" type="keyrelay:commandType" />
    <element name="response" type="keyrelay:responseType" />

    <complexType name="responseType">
     <sequence>
      <element name="panData" type="keyrelay:panKeyRelayDataType" />
     </sequence>
    </complexType>

    <complexType name="commandType">
     <sequence>
      <element name="keyrelay" type="keyrelay:keyRelayType" />
     </sequence>
    </complexType>

    <complexType name="keyRelayType">
     <sequence>
      <element name="name" type="eppcom:labelType" />
       <element name="authInfo" type="domain:authInfoType"
        minOccurs="0" />
       <element name="keyData" type="secDNS:keyDataType"
        minOccurs="1" maxOccurs="unbounded" />
```

```
        </sequence>
       </complexType>

       <complexType name="panKeyRelayDataType">
        <sequence>
```

```
        <element name="name" type="domain:paNameType" />
        <element name="paTRID" type="epp:trIDType" />
        <element name="paDate" type="dateTime" />
        <element name="authInfo" type="domain:authInfoType"
         minOccurs="0" />
        <element name="keyData" type="secDNS:keyDataType"
         minOccurs="1" maxOccurs="unbounded" />
       </sequence>
      </complexType>
  </schema>
```

11. IANA Considerations

This document uses URNs to describe XML namespaces and XML schemas conforming to a registry mechanism described in RFC 3688 [RFC3688].

Two URI assignments must be completed by the IANA.

Registration request for the extension namespace:

URI: urn:ietf:params:xml:ns:keyrelay-1.0

Registrant Contact: IESG

XML: None. Namespace URIs do not represent an XML specification.

Registration request for the extension XML schema:

URI: urn:ietf:params:xml:schema:keyrelay-1.0

Registrant Contact: IESG

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

12. Security Considerations

The "<keyrelay>" EPP extension does not allow for any object transformations.

Any registrar can use this mechanism to put key material on the message queue of another registrar, thus mounting a denial of service attack. However this can, and should be detected by the registry. The "<ext:authInfo>" element can be used as an indication that putting the key material on the losing registar's message queue is allowed.

Communication between a registrar and registry is mostly done over EPP, but communication between dns-operators, registrants or registrars mostly is not. If EPP is not used between these entities, relaying the key between a dns-operator and registrar should be

adequately authenticated for the complete relay channel to remain secure. It's out of scope for this document to describe how to authenticate other methods than EPP.

13. Acknowledgements

Maarten Wullink, Marco Davids and Ed Lewis.

14. References

14.1. 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.

[RFC4034] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", RFC 4034, March 2005.

[RFC5910] Gould, J. and S. Hollenbeck, "Domain Name System (DNS) Security Extensions Mapping for the Extensible Provisioning Protocol (EPP)", RFC 5910, May 2010.

14.2. Informative References

[I-D.koch-dnsop-dnssec-operator-change] Koch, P. and M. Sanz, "Changing DNS Operators for DNSSEC signed Zones", Internet-Draft draft-koch-dnsop-dnssec-operator-change-04, March 2012.

[RFC5931] Harkins, D. and G. Zorn, "Extensible Authentication Protocol (EAP) Authentication Using Only a Password", RFC 5931, August 2010.

Authors' Addresses

R. (Miek) Gieben
SIDN Labs
Meander 501
Arnhem 6825 MD
NL

Email: miek@miek.nl
URI: http://miek.nl/

M. Groeneweg
SIDN Labs
Meander 501
Arnhem 6825 MD

NL

Email: marc.groeneweg@sidn.nl
URI: https://www.sidn.nl/

Rik Ribbers
SIDN Labs
Meander 501
Arnhem 6825 MD
NL

Email: rik.ribbers@sidn.nl
URI: https://www.sidn.nl/

Antoin Verschuren
SIDN Labs
Meander 501
Arnhem 6825 MD
NL

Email: antoin.verschuren@sidn.nl
URI: https://www.sidn.nl/