Network Working Group
Internet-Draft
Intended status: Informational
Expires: July 24, 2014

P. Wallstrom, Ed.
.SE

January 20, 2014

EPP Registrant Security Problem Statement
draft-wallstrom-epp-registrant-problem-statement-00.txt

Abstract

This document collects a number of requirements on securing the provisioning of DNS data between a Registrant and a Registry. The most common attack in the chain of Registrant-Registrar-Registry is to inject false information into the Registrar system, which in turn forwards the injected data to the Registry using EPP, the Extensible Provisioning Protocol.

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 24, 2014.

Table of Contents

1. Introduction

The most common attack on DNS today is to somehow force a DNS Registrar to change any DNS related information by sending an EPP change ([RFC5730]) for a domain to its parent Registry. The attack could be performed at the Registrar level due to bad password handling, weak web security or any customer service being vulnerable to social engineering. Not only could DNS records be changed, but also other information about a domain name, such as the e-mail address used, the holder of the domain, or any other data needed in order to take some sort of control over the domain. There is clearly a need to protect the Registrant from this type of attack.

The standard way of shareing a DNS registry today is using the model described in [RFC3375]. This model describes the terms Registry, Registrar and Registrant (also called RRR) which will be used in this document to describe the provisioning of DNS related data.

A new somewhat new solution to protect the Registrant from any non-authorized change is for the Registry to offer the Registrant a "Registry Lock". The idea of the lock is to forbid the Registrar to provision any change to the Registry without the Registry (without any confirmation from the Registrant or the Registrar either manually or outside of the EPP protocol) temporarily removing the lock for any cheing being requested. This method has not been standardized, and there is no coherent way this locks are being implemented by a Registry, making it harder for a Registrar or a Registrant to have a single process for doing changes to all their domains. The lock can be provisioned by either the Registrar, or in direct communication

between the Registrant and the Registry. The latter completely bypasses the EPP model.

2. The role of the Registry

When using EPP to provision DNS data, the role of the Registry is to allow authenticated Registrars to publish DNS data typically coming from a Registrant. Normally the only interface available to the Registrar is the EPP interface. In some cases there might be other interfaces available that has a different feature set than EPP, this draft does only cover EPP.

The authentication is handled by The PLAIN Simple Authentication and Security Layer (SASL) mechanism presented in [RFC4616] using a a user identifier, an authorization identifier, and a password as part of a single plain-text string as documented in [RFC5730]. This document does not intend to require a change of this layer of authentication.

When the Registrar submits a transformation request to the EPP service run by a Registry, the EPP service can handle the request in a number of ways. The result can be negative, where the request is denied by the EPP service. For successful transformation commands, the command can be immediately processed by the server, or the server can acknowledge the request and postpone the action and perform it after some other action has been performed on the server side. When the change has been accepted in the Registry, any DNS change can be pushed out to the parent DNS zone, or any other data can be viewed in Whois.

The Registrant has no role in this Registrar-Registry communication at all.

In the current EPP model the AUTH data type has a special function. It is normally used for initiating a transfer of an object between Registrars. Any Registrar that has access to the AUTH data can initiate a transfer of the object, meaning that the receiving Registrar can move an object from another Registrar.

3. Improving the protocol for the Registrant

Since the Registrar in plain EPP has full control over any domain name that it is authoritative for, there is a need to improve this protocol in order to avoid attacks on the domain name through the Registrar. The Registrant wants protection against any unauthorized changes coming from the Registrar. One possible way to do this is to extend the EPP protocol in order to have a piece of data in the Registry database that authorizes any transformation request coming from the Registrar.

In order to add a control mechanism at the Registry level so that the Registrar cannot perform any changes without confirmation by the Registrar, the Registry could have a shared secret with the Registrant. This shared secret can be a token that must be present in any request sent to the Registry.

One such token can be published in the DNS zone for the domain being changed, as well as in the EPP request. The Registry can then validate the token coming from EPP by looking up the token in the current DNS zone for the domain, with extra validation from DNSSEC. When using the token in this way, it should also reflect the change being made, so that the Registrar cannot perform any other change by looking at the token available in DNS. However, the operation of doing DNS lookups in the Registry level for a large EPP operator is expensive since it adds some overhead. The Registry must also have some sort of indication that any change in its database must be protected by doing this extra operation, since not all domains for a Registrar can be locked at the same time.

Another method is to use an assymetric cryptographic key to indicate a Registry lock. A public key can be stored in the Registry database. Any change coming over EPP can then either be signed or encrypted (or both) with the private key. The Registry can verify the change by using the public key, and perform the change if the validation is succesful. If the incoming EPP request does not contain the change signed or encrypted (or both) using any existing public key for the domain, the request is denied. Using this model, either the Registrar or Registrant can have access to the private key, depending on the model of trust.

4. Bootstrapping

So how does this token or key end up in the Registry? There is still a need to keep the RRR model intact. One way to do this is to trust the Registrar completely to bootstrap the Registry and relay the token or key from the Registrant unprotected. And this is also the problem we want to avoid.

One way to bootstrap this extra security is to use DNS, since the Registrant already have control over DNS. Extra security for DNS is added with DNSSEC. Prior to sending the token or key to the Registrar for the Registry database is to publish the same data in DNS. For keys there is already a record type that can be used, TLSA. For tokens we might have to use TXT, or define a new record type.

5. Multiple tokens or keys

A private key can be lost or even compromised. In these cases you must be able to change key at the Registry. Any such change must be authorized by using a key that is already in the Registry. To avoid a situation where the Registrant has a compromised key and this leads to manual work for the Registry, the Registry should allow for multiple keys for a Registrant. Adding a new key must be done by using an already existing key, so to avoid having only a compromised key in the Registry, a Registrant should probably bootstrap with multiple keys and have an extra key in a secure backup. This secure backup key can then be used to remove any lost or compromised keys, and add new keys when needed. However, when the last key is removed, there is no Registry Lock left, and the domain is insecure.

6. Key algorithms

Since the IETF mandates algorithm agility, there must be support for multiple key algorithms. However, there will probably not be a need for protecting against downgrading attacks. But it will become a problem when new algorithms are defined since not all Registries will have support for the same algorithms. Some sort of signalling mechanism must therefore be in place.

7. Registrant interfaces

For the registrant to sign or encrypt any EPP change request, it is preferred if the Registrant can operate on the exact command being sent to the Registry. This means that the Registrant must be able to create the EPP command, encrypt and/or sign it, and send this command to the Registrar for re-distribution to the Registry. Some Registrars offer the Registrant an API for performing changes in bulk, but it is still most common for the Registrant to use any web interface offered by the Registrar. Any such API has still not been standardized by the IETF, or any other body. To solve this API problem the Registrant might offer an EPP service to the Registrant, and the Registrar becomes an EPP proxy for any secure changes. However, this does probably not make life easier for the Registrant, since the multitude of different EPP extensions in use by the different Registries (a problem Registrars already have). Perhaps a subset of EPP can be used instead. This might also give better control of any mechanism used for proxying and validating changes in XML.

8. Acknowledgements

This document is a result of many discussions with several collegues, in no special order: Einar Lonn, Ulrich Wisser, Jan Saell, Jakob Schlyter, Fredrik Ljungren and Leif Johansson.

9. IANA Considerations

This document has no actions for IANA.

10. Security Considerations

The current implementations of EPP lacks any end-to-end security from the Registrant to the Registry. This document describes a way to improve on the current model. For this mechanism to work there is a need for the Registrant to protect the private key, and the provisioning system in use. There are attacks directly targeted at the Registrar such as social engineering, spear phishing and other techniques. These are issues that are outside the scope of this document. Since XML is used in EPP, you can use XMLsig to implement cryptographic signatures directly in XML. Using signatures in XML is hard, and any implementor at either end of the system to construct and validate XML signatures.

11. Informative References

[RFC3375] Hollenbeck, S., "Generic Registry-Registrar Protocol Requirements", RFC 3375, September 2002.

[RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, July 2003.

[RFC4616] Zeilenga, K., "The PLAIN Simple Authentication and Security Layer (SASL) Mechanism", RFC 4616, August 2006.

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

Author's Address

Patrik Wallstrom (editor)
.SE
Stockholm
SE

Phone: +46 733 173 956
Email: pawal@iis.se