Network Working Group
Internet-Draft
Intended status: Informational
Expires: March 22, 2018

P. Hallam-Baker
Comodo Group Inc.

September 18, 2017

Strong Internet Names (SIN)
draft-hallambaker-sin-01

Abstract

A Strong Internet Name is a DNS name that contains a cryptographic binding to a security policy governing interpretation of the name. This document describes the use of Strong Internet Names formed using a Uniform Data Fingerprint of a PKIX trust root and outlines the additional capabilities that might be supported in a purpose written policy language.

This document is also available online at http://prismproof.org/Documents/draft-hallambaker-sin.html [1] .

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 March 22, 2018.

Table of Contents

1. Introduction

This document is written as a submission to the IAB workshop on Explicit Internet Naming Systems. The proposal described here is a mechanism that allows a security policy and cryptographic root of trust to be introduced into any Internet identifier scheme that includes a DNS name without introducing new syntax.

Strong Internet Names (SINs) bring together concepts introduced in the use of Strong Names introduced in the .Net security framework, fingerprints as used in OpenPGP and security policy description.

Incorporating a fingerprint of a root of trust into an identifier produces an identifier whose interpretation is objective and does not depend on subjective assumptions as to whether a root of trust is trustworthy or not.

Since the SIN only includes the fingerprint of the root of trust, rather than the root of trust itself, the process of interpretation will of course require a means of retrieving the additional information.

2. Definitions

This section presents the related specifications and standard, the terms that are used as terms of art within the documents and the terms used as requirements language.

2.1. Requirements Language

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 [RFC2119] .

2.2. Related Specifications

Strong Internet Names make use of the Uniform Data Fingerprint described in [draft-hallambaker-udf] .

2.3. Defined Terms

No terms of art are defined.

2.4. Implementation Status

The implementation status of the reference code base is described in the companion document [draft-hallambaker-mesh-developer] .

3. Overview

A SIN is an Internet Identifier that contains a fingerprint of a root of trust that may be used to verify the interpretation of the identifier. This section describes the manner in which SINs are used. The following section describes their construction using Uniform Data Fingerprints [I-D.hallambaker-udf]

For example, Example Inc holds the domain name example.com and has deployed a private CA whose root of trust is a PKIX certificate with the UDF fingerprint MB2GK-6DUF5-YGYYL-JNY5E-RWSHZ.

Alice is an employee of Example Inc., she uses three email addresses:

alice@example.com A regular email address (not a SIN).

alice@mm--mb2gk-6duf5-ygyyl-jny5e-rwshz.example.com A strong email address that is backwards compatible.

alice@example.com.mm--mb2gk-6duf5-ygyyl-jny5e-rwshz A strong email address that is backwards incompatible.

All three forms of the address are valid RFC822 addresses and may be used in a legacy email client, stored in an address book application, etc. But the ability of a legacy client to make use of the address differs. Addresses of the first type may always be used. Addresses of the second type may only be used if an appropriate MX record is provisioned. Addresses of the third type will always fail unless the resolver understands that it is a SIN requiring special processing.

When specified as the destination address in a Mail User Application (MUA), these addresses have the following interpretations:

alice@example.com Send mail to Alice without requiring security enhancements.

alice@mm--mb2gk-6duf5-ygyyl-jny5e-rwshz.example.com Send mail to Alice. If the MUA is SIN-Aware, it MUST resolve the security policy specified by the fingerprint and apply security enhancements as mandated by that policy.

alice@example.com.mm--mb2gk-6duf5-ygyyl-jny5e-rwshz Only send mail to Alice if the MUA is SIN-Aware, it MUST resolve the security policy specified by the fingerprint and apply security enhancements as mandated by that policy.

These rules allow Bob to send email to Alice with either ?best effort? security or mandatory security as the circumstances demand.

3.1. Resolution

Since a SIN only contains the fingerprint of a root of trust rather than the root of trust itself, a mechanism is required to resolve the root of trust from the fingerprint. The mechanism by which this is achieved is outside the scope of this document.

The Mathematical Mesh [I-D.hallambaker-mesh-architecture] is an infrastructure that is designed to resolve fingerprints to policy. But it is not necessarily the case that an entirely new infrastructure is required. The Mesh architecture was conceived as a means of achieving resolution of a SIN at a very granular level. In the Mesh, every user is their own personal root of trust and decides which resources and trust providers to delegate decisions to.

For cases in which we do not require resolution of security policy with resolution finer than a DNS domain, the DNS may be used for resolution using the existing CERT record [RFC4398]

3.2. Implicit Security Policy

A security policy may be implicit or explicit depending on the root of trust referenced and the context in which it is used.

Since many Internet applications are already designed to make use of a PKIX based trust infrastructure, the fingerprint of a PKIX root of trust provides sufficient information to deduce an appropriate security policy in many instances. For example:

https://mb2gk-6duf5-ygyyl-jny5e-rwshz.example.com/ Connect to example.com using a TLS connection with a certificate that is valid in a chain of trust that contains a certificate with the fingerprint mb2gk.

IMAP Server: mb2gk-6duf5-ygyyl-jny5e-rwshz.example.com Connect to the IMAP server example.com over a TLS connection with a certificate that is valid in a chain of trust that contains a certificate with the fingerprint mb2gk.

mailto:alice@example.com.mm--mb2gk-6duf5-ygyyl-jny5e-rwshz Encrypt mail messages using S/MIME using an S/MIME certificate that is valid in a chain of trust that contains a certificate with the fingerprint mb2gk.

3.3. Explicit Security Policy

While the implicit security policy model is sufficient for some purposes, it is less than ideal. An explicit security policy language permits much more detailed policy descriptions and links to resources that allow the policy to be realized.

A comprehensive security policy for Example Inc. should contain:

- The public key for the DNSSEC root of trust for example.com
- The public key for the DNSSEC root of trust for the DNS root
- The roots of trust for TLS, S/MIME, etc.
- The set of WebPKI trust roots to be trusted by Web browsers.
- The security enhancements (S/MIME, OpenPGP) to be applied to messages.
- Security requirements for specific services (e.g. must use TLS for inbound SMTP)

Security policy may also be specified for particular applications. For example, an email security policy for an individual user might specify:

- Message security format (OpenPGP, S/MIME) and encryption key(s)
- Authentication requirement(s)
- Content restrictions (e.g. no executable attachments)

It is very likely that to mitigate abuse a user would specify separate security policies for known and unknown senders so that use of end-to-end messaging, transfer of executable attachments, etc. are restricted to authorized senders.

One option for expressing explicit security policy is to encode the information in the DNS. Another, likely to be more satisfactory is to design a language for describing security policy.

4. Specification

The specification consists of three parts, the description of the fingerprint format itself, the means of encoding fingerprints within DNS names and the means of describing the security policy.

4.1. The UDF Format

The Uniform Data Fingerprint (UDF) format was designed to provide common format for representing fingerprints of data objects formed using a cryptographic digest function such as SHA-2 that was easier on the eye than existing URI schemes such as ni. A UDF fingerprint is formed using Base32 with optional digit separators to improve readability. The following is an example of a UDF:

mb2gk-6duf5-ygyyl-jny5e-rwshz-sv75j

Unlike traditional fingerprints calculated from the digest of the data itself, a UDF is a strong function of both the referenced data and the IANA content type.

Fingerprint = <Version-ID> + H (<Content-ID> + ':' + H (<Data>))

This approach provides semantic separation between domains. This is necessary to defeat substitution attacks such as presenting an artfully constructed PKIX certificate in a context where a JSON data structure is expected.

The Version-ID parameter specifies both the digest function and the method of application. Version-IDs are currently defined for SHA-2-512 and SHA-3-512. The values of these code points have been intentionally chosen to cause the first digit to be either an M (Merkle-Damgard) or an S (Sponge).

The specification allows for fingerprint compression in the case that the leading 25, 40, 50 or 55 bits are all zero. This allows a fingerprint of a public key represented in 20 characters (120 bits) to present the same work factor to the attacker as a 25 character fingerprint but at the cost of accepting a 225 increase in key generation difficulty.

4.2. Precision

A UDF fingerprint may be specified at any level of precision with the proviso that the work factor of a fingerprint must never be less than 2117 operations. The precision of a fingerprint may be reduced by simply truncating the text presentation.

Since verification of a fingerprint requires the verifier to compute the full SHA-2-512 hash value, an application may ?strengthen? the fingerprint by storing it with higher precision (provided this does not cause a field length limit to be exceeded.

For example, to configure her outbound email address, Alice enters the following as her email address:

mm--mb2gk-6duf5-ygyyl-jny5e-rwshz.example.com

The client resolves and verifies the root of trust and records the following in the configuration file:

Service: example.com

Trust-root: mb2gk-6duf5-ygyyl-jny5e-rwshz-sv75j-c4ozq-5gin2

This presents a work factor of 2192 for subsequent interactions while only requiring the user to type enough digits for a 2117 .

4.3. UDF Strong Labels

A Strong Internet Name is a DNS name in which one of the labels is a prefixed UDF fingerprint of a document describing the security policy governing interpretation of the name.

For example, we may form a strong internet name from the fingerprint above as follows:

mm--mb2gk-6duf5-ygyyl-jny5e-rwshz.example.com

example.com.mm--.mb2gk-6duf5-ygyyl-jny5e-rwshz

The use of a prefix of the form xx-- to identify a DNS label with a special interpretation was introduced to support internationalized DNS names. The MM?prefix is proposed as Strong Internet Names were originally developed as part of the Mathematical Mesh which builds on and extends the capabilities of strong names.

The placement of the UDF entry in the string has no effect on semantics but does affect resolution. In the first strong name, the fingerprint appears at the leftmost of the name allowing it to be resolved by any Internet application (provided the necessary DNS records are provisioned). In the second case, the fingerprint appears as the root element. This means that the name cannot be resolved by an application unless either the application or the DNS resolution service it uses understands that the name is a SIN.

5. References

5.1. Normative References

[draft-hallambaker-udf]
Hallam-Baker, P., "Uniform Data Fingerprint (UDF)", draft-hallambaker-udf-06 (work in progress), August 2017.

[I-D.hallambaker-udf]
Hallam-Baker, P., "Uniform Data Fingerprint (UDF)", draft-hallambaker-udf-06 (work in progress), August 2017.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997.

[RFC4398] Josefsson, S., "Storing Certificates in the Domain Name System (DNS)", RFC 4398, DOI 10.17487/RFC4398, March 2006.

5.2. Informative References

[draft-hallambaker-mesh-developer]
Hallam-Baker, P., "Mathematical Mesh: Reference Implementation", draft-hallambaker-mesh-developer-04 (work in progress), September 2017.

[I-D.hallambaker-mesh-architecture]
Hallam-Baker, P., "Mathematical Mesh: Architecture", draft-hallambaker-mesh-architecture-03 (work in progress), May 2017.

5.3. URIs

[1] http://prismproof.org/Documents/draft-hallambaker-sin.html

Author's Address

Phillip Hallam-Baker
Comodo Group Inc.

Email: philliph@comodo.com