Network Working Group
Internet-Draft
Expires: April 27, 2012

J. Snyder
Opus One
K. O'Donoghue
ISOC
M. Shore
TBS

October 25, 2011

A Survey of Trust Models and Relationships in Internet Protocols
draft-snyder-trust-relationships-00

Abstract

This document reviews common Internet protocols and discusses how each protocol establishes, maintains, and tears down trust relationships within the protocol. This document includes discussion of "meta" trust issues, including extra-protocol trust creation, management, and destruction. In cases where specific issues related to establishment of trust have been documented, these are discussed as well. By examining both similarities and differences between different protocols, this document can help protocol designers and maintainers in IETF working groups learn from successful (and un-successful) Internet protocols.

Table of Contents

1. Introduction

Many Internet protocols need to establish some type of trust between the parties participating in the protocol in order to be effective. For example, the Internet Key Establishment (IKE) protocol ([insert] [references] [here]) passes through an authentication phase between the two IKE peers before it moves to a second phase where cryptographic material is established for encrypting and authenticating IPsec traffic. The authentication phase serves to establish trust between the two IKE peers. For example, if the IKE peers use pre-shared secrets, then each IKE peer is willing to trust the other once they have mutually proven knowledge of a pre-shared secret.

Please note that this document was derived from existing protocols and does not attempt to define or re-define the function of any Internet protocol. This document is entirely non-definitive and should not be used by implementers as an authoritative source of information about protocol behavior or description.

WHY IS THIS IMPORTANT?

NEED A BETTER DESCRIPTION OF "TRUST" HERE AND WHAT WE WILL BE LOOKING AT EXACTLY.

The protocols described in the document were chosen for their exemplar value. This document is not meant to be an exhaustive description of all protocols and their trust establishment models.

2. Terminology

Trust: This is the definition of Trust.

Authentication: This is the definition of Authentication.

Identification: This is the definition of Identification.

Reputation: This is the definition of Reputation.

3. Overview and Problem Statement

In this section, we would provide as much background and other related information as we can to help describe some things including...

WHY ARE WE DOING THIS?

WHAT IS THE VALUE OF THIS CONTRIBUTION?

WHAT ARE WE NOT INCLUDING IN THIS DOCUMENT AND WHY?

4. DKIM (Domain Keys Identified Mail)

4.1. DKIM Background and Overview

Protocol Overview

4.2. Trust Relationships in DKIM

Trust Models and Relationships in DKIM

4.3. DKIM Diagrams

Diagrams go here

5. DNSSEC (Domain Name System Security Extensions)

5.1. DNSSEC Background and Overview

Protocol Overview

5.2. Trust Relationships in DNSSEC

Trust Models and Relationships in DNSSEC

5.3. DNSSEC Diagrams

Diagrams go here

6. PKI (Public Key Infrastructure)

6.1. PKI Background and Overview

The IETF PKIX working group has specified an X.509v3 profile, and that profile and set of associated specifications are colloquially referred to as PKIX. The core specification is RFC 5280.

Throughout this section we look at how trust is conveyed in PKIX from two perspectives:

(1) from the perspective of a relying party -- an entity that receives an assertion (credential) and needs to make a decision whether or not to trust it, and

(2) from the perspective of an end entity -- an entity that needs to assert its identity in a way that can be accepted by a relying party.

By way of terminology, the entity which signed a certificate and which is vouching for the authenticity of both the certificate and the certificate holder is referred to as the issuer. The entity to which the certificate was issued is the subject.

Trust in PKIX is instantiated through the use of trust anchors. A trust anchor is itself a certificate, but one about which a human has made an explicit trust decision. In this context, subsequent trust decisions must successfully chain back to that initial decision -- that a certification authority is reliable, secure, and honest, and that its business practices provide assurance that it will only be issuing certificates to entities which are also reliable, secure, and honest.

This document does not yet discuss the Trust Anchor Management Protocol. [insert][reference][here] TAMP does not change the underlying trust model or the trust lifecycle, although it does provide mechanisms for implementing it.

6.2. Trust Relationships in PKI

6.2.1. Basic Model

Perhaps the key assumption around which PKIX is built is that it is not necessary for two entities to have an existing relationship in order to make a decision whether or not to accept the otherE1/4s assertions as 1) correct, and 2) trustworthy. Rather than negotiating in advance of any communication, those decisions are mediated through the use of third party agents, and consequently whether or not a given entity is trustworthy comes down to the question of whether or not the agent (and its agent, and on up the chain) can be seen as trustworthy and authoritative, and can make reliable assertions about the credentials it has issued.

A certification authority, which may or may not be a commercial entity, issues signed credentials for its customers. These credentials are known as end entity certificates. Its signature is essentially an attestation that the CA has some level of confidence that the entity to which the certificate was issued really is who it claims to be. Certificates may be chained from a trust anchor --

that is to say, there may be from zero to n certification authorities between the trust anchor and the end entity to which the certificate has been issued.[insert][reference][here]

Trust is instantiated by provisioning a root certificate in a local cache or in some logically local data store. This root certificate functions as a trust anchor. If the process of validating an end entity certificate does not terminate at a trust anchor, the validation fails.

The data model is essentially hierarchical, and tree-shaped. While a CA may issue multiple (typically many) certificates, a certificate may have only one issuer. At the very top of the trust tree is a person or organization who determines which root certificates represent a trusted CA (note that this decision and associated information are basically determined manually and out-of-band, typically requiring human judgment).

Bidirectional trust may be established between two CAs and their subjects through the use of cross-certification. In this case the two CAs issue certificate to each other. It is still the case, however, that a certificate will have one issuer, and that a CA may issue multiple (many) certificates. The decision to cross-certify is still out-of-band, and human. The question of what the trust anchor is in this situation is still being debated on the pkix mailing list5 and is unresolved as of this writing. (Oct/2011)

Self-signed certificates merit special mention, because they are so commonly deployed. A self-signed certificate is one in which the issuer and the subject are the same. It is very rarely the case that a self-signed certificate is already installed in a root cert cache and is functioning as a trust anchor, but it is very common for users to accept and install self-signed certificates when they are offered by a visited website.

6.2.2. Creating and instantiating trust

There are two aspects to creating trust and instantiating it through PKIX technologies. The first aspect relates to the determination made by a user or systems administrator (i.e. a relying party) that a given certification authority is a reliable source of authority regarding the identity of the entities represented in the certificate it issues. The second relates to the determination made by the end entity that a given certification authority is a reliable agent -- that they are who they say they are, that their business practices are sound, that the operation of their certificate infrastructure is secure, and, perhaps most importantly, that the chain to the trust anchor contains only issuers who are also secure, reliable, and

trustworthy. The relying party also needs to have assurances about intermediate CAs and certificates in a chain, but this comes into play during validation, not during provisioning.

6.2.2.1. Bootstrapping trust in a relying party

From an end entity perspective, trust is instantiated, or verified, through the presence of trust anchors in a local store. A decision to install or provision a root CA certificate as a trust anchor is an out-of-band, human decision and represents a decision to trust that the CA represented by that certificate is secure, reliable, and authoritative. It also represents a decision that the intermediate CAs underneath the root CA are also secure, reliable, and authoritative (this has turned out to be a problem, in practice).

It is typically the case that web browsers are distributed with a cache of root certificates, which have been vetted with varying degrees of rigor by the browser developers. When a user decides to use a browser with an existing cache, theyE1/4re implicitly trusting the browser developers. This is not unreasonable -- in theory, the browser developer has the resources and expertise to evaluate trust anchors for inclusion, and will exclude certificates from unreliable CAs.

In other cases, often in cases where a local CA is issuing certificates, a local systems administrator makes the decision to add a root CA certificate from a local (or neighboring) CA.

A special case of bootstrapping trust, and one which poses a security problem, is that a user may be offered an unknown certificate, be asked by the browser whether or not to accept it, and will not only accept the certificate as authentic but also install it locally for future use. In this situation there is an apparent disconnect between whatE1/4s happening conceptually in the security transaction (the user is being asked whether or not to accept a credential as both authentic and trustworthy) and the userE1/4s understanding of whatE1/4s going on (the user just wants the connection to complete and may not understand the underlying security model).

6.2.2.2. Bootstrapping trust in an end entity

In this case, bootstrapping essentially means investigating certification authorities, making a decision to acquire a certificate from one, and installing that certificate. Again,this is a human decision thatE1/4s instantiated through technical means (the provisioning of the certificate).

Unfortunately there really is no way, as a relying party, to

determine the soundness of the end entityE1/4s decision to acquire a cert from a particular CA. It may be that they chose one CA over another on the basis of business practices but it may also be the case that they chose the least expensive vendor regardless of soundness. When things are working as they should a CA will only sell certificates to other very reliable CAs, and on down the chain, but there have been several issues with compromised or sloppy intermediate CAs in the recent past that call this model into question.

6.2.2.3. A brief digression on EV certs

The CA/Browser forum has published guidelines for identity verification, including specification of specific identity criteria. These center around three goals:

(1) establish the legal identity of the certificate applicant;

(2) establish that the applicant has legal ownership of the entity for which the certificate is to be issued (the Subject), and

(3) confirm the identity and the authority of the ownerE1/4s agents.

Certificates issued under these criteria are called Extended Validation Certificates. Browser markers provide visual clues, such as color in the address bar, when an EV certificate is present and has been validated. A CA must typically pass an independent audit to be accepted by browser vendors as an issuer of EV certs.

6.2.3. Validating Trust

In PKIX, trust is chained back to a trust anchor. Validation essentially consists of path validation, with the assumption that youE1/4ll trust who your anchor vouches for, and so on up the chain.

It may also be the case that a non-root certificate - an end-entity certificate thatE1/4s not a trust anchor, is explicitly trusted, usually through local installation in a browser or other cache. Unfortunately itE1/4s often the case that the user is making a decision to get a connection to work rather than making an explicit trust decision.

6.2.4. Revoking Trust

The X.509 lifecycle model typically is based on a long-lived credential (months or, more often, years) which may expire without being reissued, or may be explicitly revoked. Explicit revocation may be accomplished through a variety of measures:

(1) Manual removal from a browser or other certificate cache,

(2) Blacklist checking by the relying party as part of the validation process. This, in turn, may take one of several forms:

(a) Certificate revocation lists, issued by the certification authority which issued the original certificate. These should be created and published on a regular, timely schedule and must be checked as part of the certificate validation process.

(b) A status query at validation time, through the use of the Online Certificate Status Protocol

(c) Blacklisting by the browser vendor

The technical means for revoking trust is essentially the same as that for revoking a non-trust anchor certificate. If the trust anchor is gone, certificates which chain back to it will fail the validation check.

6.3. PKI Diagrams

Diagrams go here

7. RPKI (Resource Public Key Infrastructure)

7.1. RPKI Background and Overview

Protocol Overview

7.2. Trust Relationships in RPKI

Trust Models and Relationships in RPKI

7.3. RPKI Diagrams

Diagrams go here

8. IANA Considerations

None.

9. Security Considerations

To be supplied.

10. Acknowledgements

Insert list of key collaborators..

11. References

11.1. Normative References

11.2. Informative References

Authors' Addresses

Joel Snyder
Opus One, Inc.
1404 East Lind Road
Tucson, Arizona 85719
US

Phone: +1 520 324 0494
Email: jms@opus1.com
URI: http://www.opus1.com/jms

Karen O'Donoghue
The Internet Society
7167 Goby Lane
King George, Virginia 22485
US

Email: odonoghue@isoc.org
URI: http://www.isoc.org

Melinda Shore
TBS