The Omnibroker: Improving Internet Trust Services

Phillip Hallam-Baker
Comodo Inc

The Real Problem of Internet Trust

In this paper we focus on the issue of Web site security because that is the problem that existing solutions have evolved to address and thus the problems that are most likely to be familiar to readers. It is nevertheless important to remember that the greatest vulnerabilities arise from the cases where no security is used at all rather than from what some may consider to be less than perfect solutions for those cases that are secure.

We have the technology that could enable every communication between a browser and a Web site to be authenticated and encrypted and for every email to be carried over an encrypted transport and for every document to be stored in an encrypted form. What we lack is the necessary infrastructure to enable that technology to be applied.

The Internet Trust Infrastructure

The deployed Internet Trust Infrastructure has two principal parts. The Public Key Infrastructure (PKI) is commonly considered to be the 'trust' infrastructure. But equally important in establishing Internet trust is the Internet naming system, the Domain Name System (DNS) (1).

The Internet PKI is based on the IETF PKIX standards (2) which are in turn based on the ISO/ITU standard X.509 (3). Although the most visible application of the Internet Trust Infrastructure is securing Web sites using SSL/TLS (4), code signing is at least as important in maintaining the security of hosts connected to the Internet.

Figure 1 shows the high level trust relationships in the standard PKIX model. The arrows show trust relationships, the arrow pointing from the party conferring trust to the party being trusted. The green arrows show the direct trust relationships from which a trust relationship between the Relying Party and the Internet Service may be established if warranted. The red arrows show the constructed trust relationships.

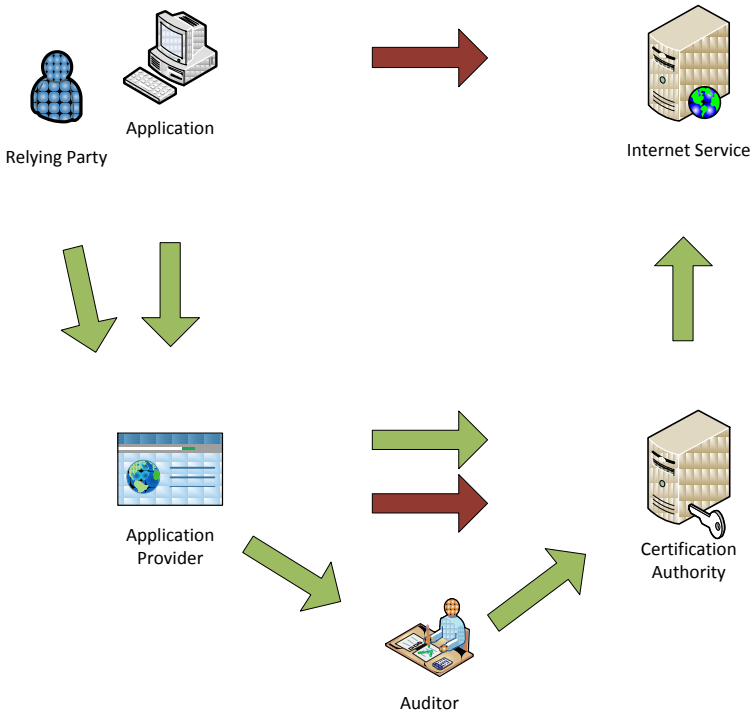

Figure 1 The PKIX Trust Model.

A *Relying Party* (e.g a *User*) uses an *Application* (e.g. a browser) to access an *Internet Service* (typically but not necessarily a *Web site*). In practice this is a constructed relationship since the user is typically selecting the *Application Provider* rather than the Application itself. The first link in the trust chain is thus the relying party selects an Application Provider and delegates the task of trust management to them.

The Internet Service authenticates itself to the relying party by means of a *private key* and a Certificate that binds the corresponding *public key* to attributes of the Internet Service. At minimum these attributes comprise the *DNS name* of the Internet Service but may also include the X.500 Organization Name of the subject. Even though the X.500 protocol is essentially defunct as a directory infrastructure, the X.500 names are often preferred over DNS names as being closer to the 'familiar' or 'ordinary' name.

The Certificate describes the manner in which the stated attributes have been validated by means of a *Certification Policy (CP)* identifier (an ASN.1 OID) which corresponds to a document describing the validation criteria.

The Certificate is issued by a *Certification Authority (CA)* after performing a *Validation Process* described in the Certification Authority *Certificate Practices Statement (CPS)*. The CPS is a description of a specific process that meets the requirements specified by the CP.

The Application trusts certificates issued by Certification Authorities that have been selected as *Trustworthy* by the *Application Provider.* The selection criteria usually include requiring that the CP

and/or CPS meet specific criteria and require that the CA hold a current audit under an approved audit regime by an auditor that meets the auditor selection criteria of the Application Provider.

The basic trust model shown in Figure 1 is an idealized form. In practice the performance of the Internet Trust Infrastructure is subject to important technical limitations, limitations which also affect any attempts to change it.

Downgrade Attack

The most significant limitation on the Internet Trust Infrastructure is that security is an optional feature disabled by default. There is no secure means of knowing when security is enabled let alone 'how much' security should be required. The simplest and most effective form of attack against an Internet application is almost always to simply turn off the security enhancement.

If a user types a domain name into the address bar of a Web browser, the browser will attempt to establish a connection using an unencrypted connection. Sites that have deployed SSL typically issue a HTTP redirect to tell the browser to switch to use of SSL.

Figure 2 shows how an attacker can defeat this form of TLS by turning it off completely. The application attempts to connect to the Web site but the connection is intercepted and redirected to a different site, the Man-In-The-Middle (MITM) site. In the simplest form of this attack the MITM simply suppresses the instruction from the legitimate Web site to switch to use of SSL.

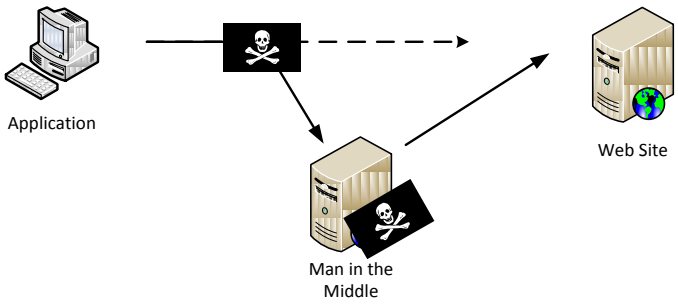

Figure 2: Man in the Middle Attack

Note that this attack cannot be defeated merely by trying to establish a secure connection first. The vulnerability to downgrade attack exists if the application is willing to accept *any* connection that does not provide authentication.

The vulnerability to downgrade attack was recognized in the design of the SSL protocol and led to the display of a padlock icon when an encrypted connection had been established. This makes the user responsible for foiling a downgrade attack, presenting them with a security signal that implies they are safe when all that is assured is the use of encryption.

In a more sophisticated MITM attack, the connection between the Application and the MITM is also encrypted. Instead of issuing a redirect to the encrypted form of the legitimate site, the MITM sends a redirect to an impostor site with an SSL certificate, thus ensuring display of the padlock icon. The success

of this strategy depends on the vigilance of the user and the ability of the application to provide information necessary to detect the substitution.

In the most powerful form of the attack, the attacker obtains a Certificate from a trusted CA by defeating the validation checks designed to mitigate or prevent mis-issuance.

Turning security off completely is one form of *downgrade attack*. Another form of downgrade attack is to substitute a lower security solution for one that offers higher security. This type of attack is a major concern when attempting to develop stronger forms of Internet security as the weakest link in the infrastructure is typically the link that advertises support for the higher security feature.

The Internet has over two billion active users. Any attempt to change the Internet infrastructure must take account of the constraints imposed by deployment or it will fail. New Internet security proposals must thus face the dilemma that they must support incremental deployment to succeed in deployment without the risk of being nullified through a downgrade attack.

Platform Provider

Some Operating Systems (e.g. Windows) provide a platform level Certificate Root store to which Application Providers may choose to delegate the selection of trusted CAs.

Application Providers supporting multiple platforms (e.g. Windows, OSX, Linux) can achieve consistency across their product line by performing the selection themselves. Delegating the selection task to the platform provider achieves consistency across applications on the same platform, ensuring that browser X has the same trust characteristics as browser Y on that particular machine.

Root, Intermediate and Cross Certificates

PKIX distinguishes between two types of certificate:

End-Entity Certificates
: Certificates that bind attributes of an end entity to a public key.

Certificate Signing Certificates
: Certificates that delegate trust to a public key used to sign certificates.

The two types of certificate are mutually exclusive in the PKIX model. A certificate that advertises both roles is not a valid PKIX certificate. While a ‘self-signed’ end entity certificate is a legal X.509v3 certificate, it is not a valid PKIX certificate. Since use of such certificates on the Internet is common it follows that PKIX describes one approach to using X.509v3 certificates on the Internet rather than the only valid one.

Early models of Internet trust assumed that certificates would be arranged as a hierarchy with a root CA issuing certificates to sub-CAs which would in turn issue certificates to sub-sub CAs (Figure 3). Privacy Enhanced Mail (PEM)(5) being the exemplar of this approach. This model proved impossible to deploy as liability and control were unacceptably concentrated at the root of the hierarchy. The root CA would potentially be a party in every dispute arising from an Internet transaction, the ability to manage

litigation risk was thus in serious doubt. Key holders would have no recourse were the root CA to decide to impose unacceptable terms and/or costs on certificate renewals.

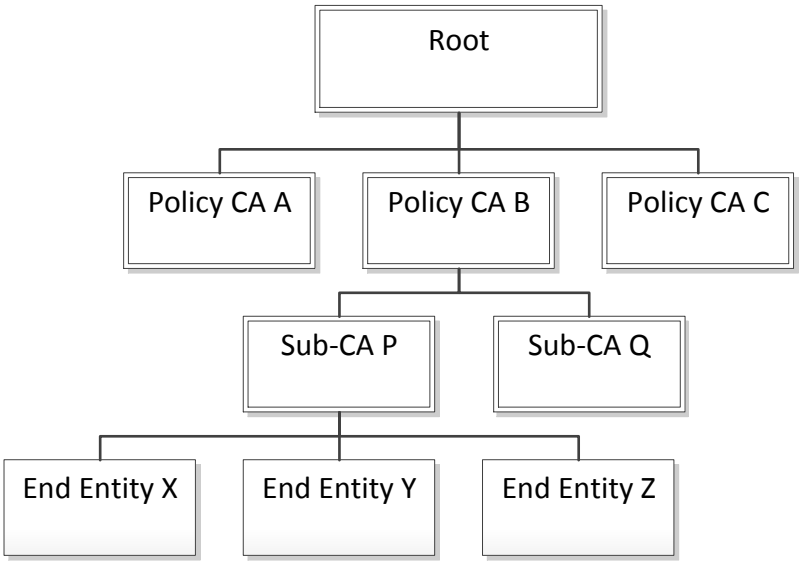

Figure 3: The unimplemented PEM hierarchy

In the model that evolved in the place of the monolithic PEM hierarchy, each CA established its own root of trust which is traditionally expressed as a self signed certificate in which the issuer and subject name are the same and the signing key is the same as the subject key. It is these 'root certificates' that are embedded in Internet applications such as Web browsers.

It should be noted that even though root certificates are expressed in the same syntax as other certificates, processing by Internet applications is very different. In particular, applications are not required to check the revocation status of root certificates, nor do they verify signatures. It is likely that a very different structure would have been chosen for root key distribution had the original X.509 infrastructure anticipated the existence of more than one root.

According to current best practice, a CA should maintain the private keys corresponding to embedded root certificates in an offline 'air-gapped' environment and limit the network exposure of all other certificate signing keys to the maximum practical extent. This means that a CA should establish an internal hierarchy with at least two levels corresponding to the offline and online private keys. The additional certificates between the root certificate and End Entity certificate are known as Intermediate certificates (Figure 4).

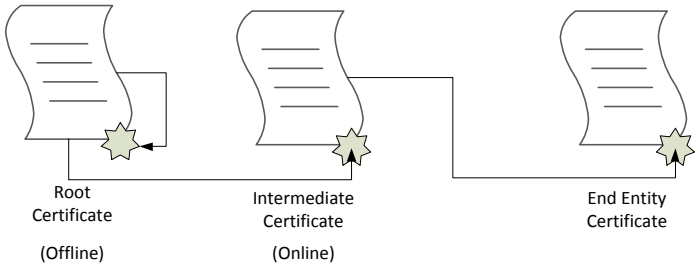

Figure 4: A Certificate Chain with Intermediate certificates

A CA may also issue an intermediate certificate to identify a group of certificates with a certain property. This is often done to identify a group of certificates issued to a particular organization or group of organizations. Such certificates provide authorization attributes in addition to authentication since applications can determine if a host is authorized part of the organization trust network by checking to see if the trust chain contains the correct intermediate root (Figure 5).

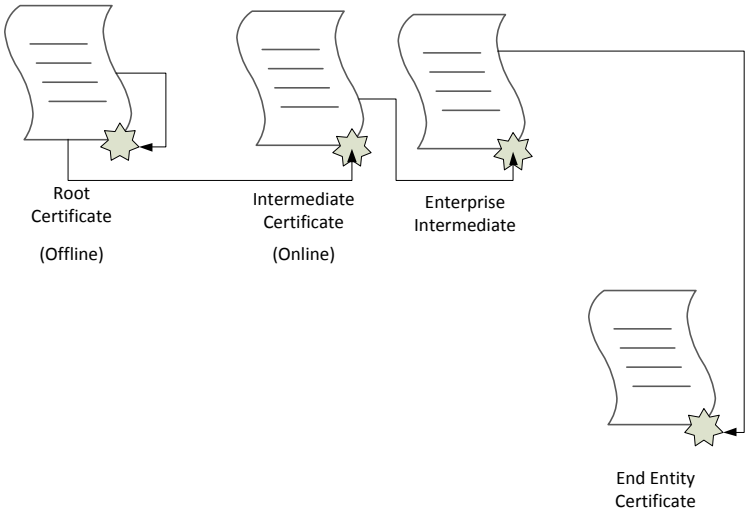

Figure 5: Enterprise Intermediate Certificate

In certain circumstances it is desirable for one CA to accredited certificates issued by another CA. To do this the CA issues an intermediate certificate to the root (or an intermediate certificate) of the CA being accredited. Such a certificate is known as a *Cross-Certificate*.

Figure 6 shows a cross certificate from CA A to the online issuing key of CA B. This will allow CA B to issue certificates that are accepted by browsers deployed before CA B began distributing its root key. Since this type of cross certificate will enable CA B to compete with CA for business it should be exceptionally rare.

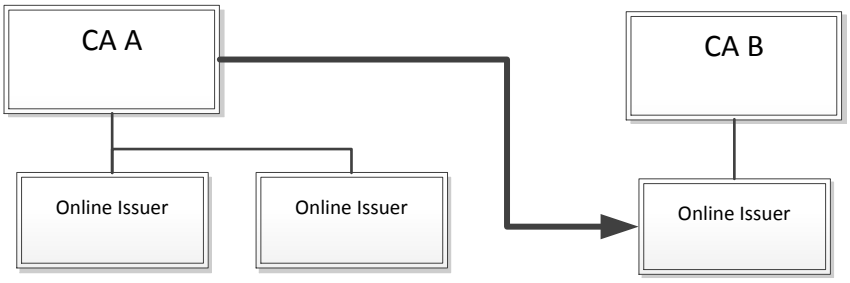

Figure 6: Cross Certification

The distinction between a cross-certificate and an intermediate certificate is legal rather than technical. It is not possible to distinguish a PKIX cross certificate from another type of intermediate certificate by examining the bits. From a technical point of view, a cross certificate is simply another link in a certificate chain. The key difference between a cross certificate and an ordinary intermediary certificate is that it delegates trust to a different, independently operated CA. The liability, policy and practices concerns are thus much more complex.

A notable application of cross certification is the US Federal Government *Bridge CA*. Although the US federal government is a single organization, there is a separation of powers between the executive judicial and legislative branches. Within the executive branch the individual departments and agencies have a long tradition of jealously guarding their independence. While the President is the head of the executive branch, the Executive Office of the President was determined to avoid mediating inter-agency disputes relating to the technical standards for operating PKI.

The US Federal Bridge CA(6) is a cunning solution to the problem of creating a root authority for a group of organizations that refuse to accept a peer as their root CA. Each member of the bridge CA operates their own CA and configures their systems to recognize their own root of trust. Each recognizes the bridge CA as a peer, exchanging accreditation with the bridge by means of a pair of cross certificates. The bridge thus serves as a hub, allowing a trust path to be established from any root CA that is a member to any End-Entity certificate issued by a member.

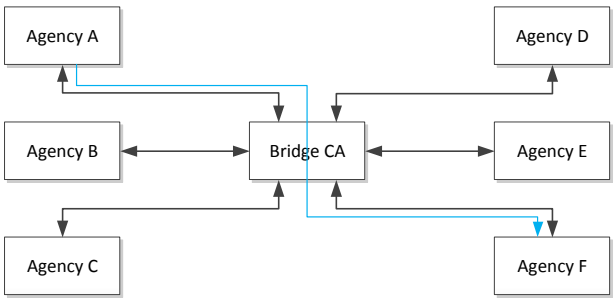

Figure 7: Bridge CA

Figure 7 shows a Bridge CA with six members. Agency A recognizes certificates issued by Agency F through the path shown in blue. Each agency recognizes certificates issued by the other members without ceding control of its PKI to any other party.

From an operations point of view, the bridge CA is functionally equivalent to establishing a single root CA with the one difference that each member retains a credible exit option.

Path Constraints

The PKIX specification allows a Certificate Signing Certificate to be constrained in various ways:

- Signing for particular DNS or X.500 names
- Signing for specific policies
- Limiting the length of subordinate trust paths.

The last constraint is particularly important as it permits a Certificate Signing Certificate to be limited to signing End Entity Certificates. Use of this constraint in a certificate for an online certificate signing key prevents it being used to create a new Certificate Signing Certificate.

Although most CAs make use of path length constraints, very few employ name constraints which are only supported by some of the deployed Web browsers. To be compliant with the DRAFT PKIX standard, a name constraint extension must be marked critical which requires any client that does not process the extension to reject the certificate. Since certificates that do not work with deployed clients are unacceptable to customers, Certification Authorities wanting to make use of name constraints have the option of waiting for the PKIX standard to be modified or issuing certificates that are not compliant with the standard.

Revocation

A PKIX End Entity certificate is typically valid for one to three years after issue. In some cases a CA will discover a reason to withdraw its accreditation of the subject or the subject's public key before the expiry date is reached. These include:

- The subject made a mistake in their original certificate request
- The subject has lost control of (disclosed/destroyed) their private key
- There has been a material change to the subject identity
    - Change of name
    - Bankruptcy
- The subject did not pay for the certificate or the payment was cancelled
- The original certificate request was found to be fraudulent or otherwise invalid
- The original certificate was issued in error

Of these reasons for revocation, the overwhelming majority result from administrative error rather than an intentional default. Very few subjects have extensive cryptographic expertise and many of the tools the administrators are expected to use have execrable usability.

In such cases the CA should revoke the original certificate. PKIX provides two mechanisms for advertising changes to certificate status; CRLs and OCSP.

A Certificate Revocation List (CRL) is a list of the serial numbers of the certificates that the issuer intends to issue a change of status for. This mechanism is simple but scales very poorly as the CRLs issued by large CAs become unmanageably large. Partitioning of the CRL allows the length of the CRLs to be reduced but also reduces the benefit of CRL caching.

Online Certificate Status Protocol(7) overcomes the scaling limitations inherent in the CRL approach by replacing the CRL with an online service that provides an authenticated report of the validation status of a certificate (Figure 8).

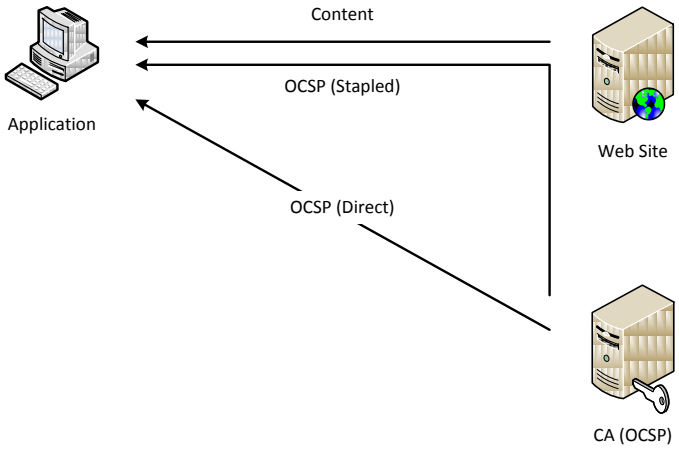

Figure 8: OCSP Online Certificate Status

Multiple CAs

The Certification Authority is selected (and paid by) the Internet Service. This gives rise to a question of conflict of interest as Internet Services seek a Certification Authority that offers the lowest price and/or least effort to comply with the requirements of the Validation Process.

Another consequence is that it is not necessary for an attacker to compromise the CA selected by a subject to gain a bogus certificate for that subject. Compromise of any CA will allow the attacker to gain a bogus certificate.

This attack is illustrated in Figure 9. The subject has selected CA C as their CA but the attacker can compromise the site by obtaining a certificate from CA B, a ‘rogue’ CA.

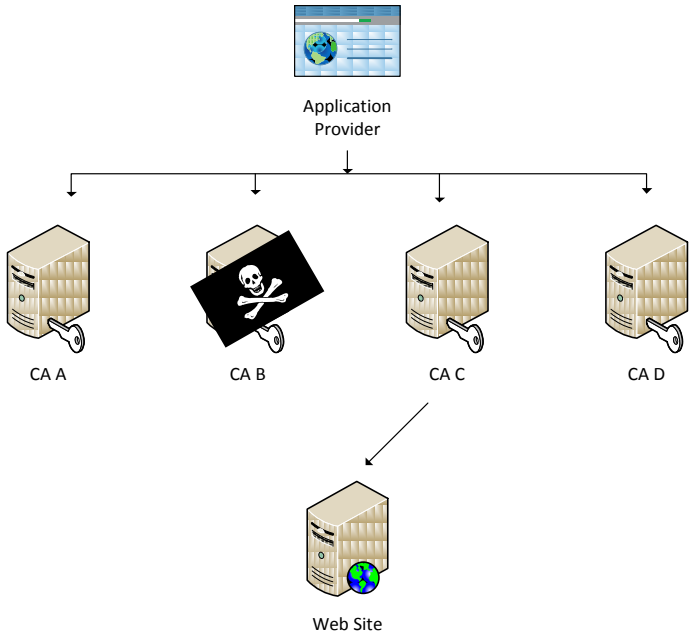

Figure 9: Multiple CAs

Registration Authorities and Resellers

Another important detail in the CA infrastructure is that the PKIX protocols are designed to permit certificate issue and certificate validation to be separated by means of a *Registration Authority*.

Figure 10 shows an example of a certificate being issued through a Registration Authority. The Web site applies for the certificate through a Registration Authority which is the sole point of contact for the subject. The certificate itself is issued by the Certification Authority which is (by definition) the only party capable of creating the certificate.

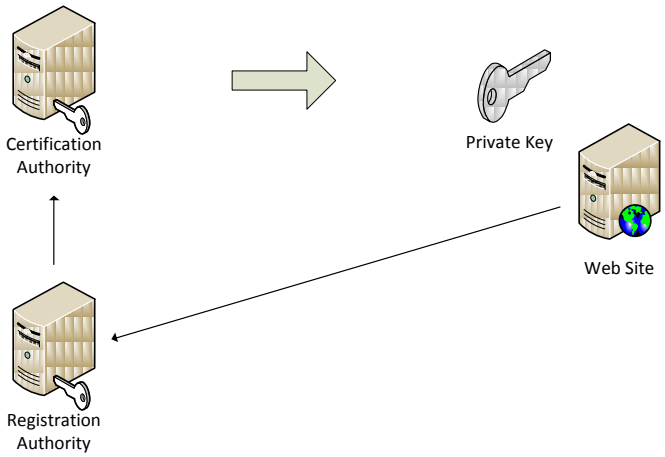

Figure 10: Registration Authority

In PKIX terms, a Registration Authority is responsible for performing at least some validation of a certificate request. This arrangement reduces some security risks but may increase others. Commercial

CAs often issue certificates through affiliate or *Reseller* relationships that have similar external properties but do not involve delegating any part of the certificate validation process to the reseller.

A CA based in the US may be more likely to achieve accurate validation of certificate requests from the country of Elbonia if the validation is carried out by a native speaker of Elbonian who is familiar with the particular business incorporation practices in that country. It may therefore be desirable for a CA to delegate validation of these aspects of a certificate request by an Elbonian affiliate acting as an RA.

While the use of RAs can reduce risk, there is also a risk that the introduction of the RA may introduce an additional point of vulnerability that an attacker might exploit. There is also a risk that an attacker might compromise the communication between the RA and the CA and gain the ability to issue certificates without any validation checks being performed.

The extent to which RAs introduce or mitigate sources of risk depends on technical and practices issues that are not currently visible to external observers. It is not even possible for an outside observer to distinguish an RA from a reseller.

Legal and Practices Considerations

The purpose of a Public Key Infrastructure (PKI) is to control or mitigate risk. Should the PKI fail, there is a possibility of litigation. The legal and technical aspects of PKI are thus tightly coupled.

The Internet is a global infrastructure and Internet services are global in scope. Internet law is inherently complex as a plaintiff in a civil matter may take a dispute to any court willing to accept jurisdiction and criminal prosecutors may seek to apply national law on an international scale. A legal strategy based on exploiting specific privileges granted in one jurisdiction is almost certain to fail in another. Equally a strategy that ignores the existence of local legislation will also fail.

The law relating to Public Key Infrastructure (PKI) is even more complex than general Internet law since it is a new technology with little precedent and also because many governments recognized PKI as a key infrastructure necessary to serve electronic commerce at a very early stage and passed legislation that was intended to stake a claim on the new digital frontier. Much of that legislation has been overtaken by subsequent developments but the laws are still in force and the requirements imposed on the use of electronic signatures and the operation of Certification Authorities, remain.

Despite the legislative interest, PKI has given rise to remarkably little case law in approaching two decades of deployment. There are few precedents to guide courts in common law jurisdictions. This leads to considerable legal uncertainty as to how courts might rule should a case be brought.

Certification Policies

At a minimum, an SSL/TLS certificate provides a binding between a public key and the domain name of the subject. Certificates issued under a policy that only requires this minimum level of authentication are generally known as *Domain Validation* (DV) certificates.

While verifying the domain name is a necessary security objective, it is hardly sufficient to establish a strong trust relationship with an unknown party. DNS domains are easy to obtain, mere holdership of a DNS domain name does not establish the accountability of the subject. The limited trust offered by DV certificates was realized by the earliest Certification Authorities who initially required verification checks on the organization making a certificate request. These certificates are known as *Organizational Validation* (OV) certificates.

Although OV certificates are intended to provide a higher degree of assurance than DV certificates, each Certification Authority defines their own criteria leaving the assessment of the additional assurance offered to Relying Parties that are rarely capable of doing so. This deficiency led to an industry effort of CAs and Browser providers to develop criteria for a form of OV certificate that met a specific standard for establishing accountability. These new certificates are known as *Extended Validation* (EV) certificates.

Is DV Enough?

Even though verification of the domain name alone is insufficient to establish trust in the general case, there are two particular exceptions when just knowing the domain name is sufficient:

- The domain name holder is a global brand that has a well known domain name.
- The domain name holder does not need to establish trust beyond its existing community.

These two cases represent the very largest and the very smallest organizations that use the Internet which has led many to assert that DV certificates are sufficient for all purposes. This is a fallacious argument as the reason that one group finds DV certificates sufficient is that they are already highly trusted while the other group finds DV certificates sufficient because their application does not require a trust relationship at all. Between these two extremes lie the vast majority of Internet based businesses whose primary concerns are how to reach out to new potential customers to generate sales leads and how to convert those leads into sales.

Should ICANN require accountability?

One approach to making DV certificates sufficient is to change ICANN rules to require all DNS domain name holders to establish their accountability. Proposals to this effect have been made with increasing regularity over the past decade as successive Internet crime problems have demonstrated a lack of accountability in Internet use.

Such proposals face the challenge that procedures for registration of DNS names are designed to make names easy to obtain and a large industry has been built based on that position. The openness of the Internet itself depends on the ability of any Internet user to establish themselves as a first class digital citizen by acquiring a domain name of their own.

A business that uses business@freemail.com as their contact address is investing in a brand and a contact address that they do not own and may change their terms of service at any time. A consumer who uses the ‘free’ email service bundled by their cable ISP provider has a high switching cost should they decide to move to another ISP and thus little leverage in negotiations. Changes to the registration

process that made DNS names harder to obtain would impose a transaction cost on the openness of the Internet.

ICANN may not have the power to make such a change even if the organization was inclined to do so. Control over the management of communications infrastructures remains a top priority for many governments. Several governments have proposed that 'unfair' US influence over ICANN decisions be eliminated by transferring responsibility for running the DNS root system to the International Telecommunications Union (ITU). It is highly doubtful that ICANN could attempt any contentious change in the operation of the DNS system without at least some governments using the controversy as a pretext to establish a rival organization.

Liability

Default by an Internet Service may result in an actual (or alleged) loss by the Relying Party. This might result in a lawsuit against one or more of the other parties involved in the trust relationship. Moreover since the Internet is global in scope, the Relying Party may have suffered a loss in any national jurisdiction. In the long run regulators will almost inevitably take the side of consumers over the creators and operators of an infrastructure the consumers use.

It was in large part the realization that regulators would insist on the consumer being protected from the consequences of fraud that led to the need for Application Providers to create Certification Authorities as separate entities in the early design of SSL. One of the major design constraints of the SSL model was that the Application Providers were anxious to transfer as much liability as possible onto other parties. This required liability to be concentrated at the Certification Authority.

Anti-Trust and Competition Law

Selection of Certification Authorities by Application Providers with a dominant market position may raise Anti-Trust concerns in the US or their equivalents in other jurisdictions.

Whether such concerns are valid or not, an Application Provider that refuses to extend trust to a CA may face an expensive legal challenge. This constraint has led some critics to claim that this leads to a failure in the existing Internet Trust Infrastructure as Application Providers may be coerced into extending trust to Certification Authorities that they would not recognize if the only criteria were the strength of the validation processes.

Coercion

The Internet Trust Infrastructure was originally designed to mitigate and control commercial risks arising from Internet transactions. In particular the original design criteria that SSL and Code Signing were designed to meet was to make Internet commerce as safe as traditional 'bricks and mortar' commerce.

Recent years have seen a dramatic rise in the use of the Internet as a political medium and in particular as a medium of political protest and political engagement in countries with authoritarian regimes. Attempts to limit or control Internet use have increased proportionately (8).

This change means that the Internet Trust Infrastructure must address threat of attack by state actors and groups acting under state direction. Such actors have very different objectives and capabilities to those of ordinary Internet criminals and the consequences of a successful attack go beyond a mere financial loss.

One important capability that is (in practice) unique to state actors is the ability to coerce a CA to default. The threat of coercion by professional criminals can be rendered negligible through standard separation of duties controls. It is highly unlikely that a professional criminal will attempt to coerce a CA if doing so is going to be less likely to succeed and offer less in reward than robbing a bricks and mortar bank.

A state actor has more to gain from a CA default and can bypass separation of duties controls by threatening all the employees involved with jail or other sanctions.

Usability

Usability is a key concern in the design of all Internet applications. One of the principal reasons that the World Wide Web was more successful than other network information systems was the ease of use and the high quality of the early Web browsers relative to the client applications for competing schemes.

Although usability is a primary concern for browser providers, the requirement of security usability is to make the user safe. This requirement is frequently in conflict with the traditional usability concern of making a system easy to use.

Speed

Browser providers place great emphasis on the time taken for their products to connect to a site and start showing information to the user. Browsers that are perceived as being slow are quickly abandoned in favor of faster alternatives.

Connection Robustness

A Web browser must be capable of connecting to a Web site in any environment where the user has at least some form of Internet connection.

A majority of Browser providers have consistently refused to implement security features in a way that would cause Web sites to work from some Internet connections but fail in others. One consequence of this refusal is that most Web sites will accept a certificate regardless of the revocation status if an attempt to retrieve the revocation status fails.

User Interpretation of Security Signal

Most browsers show a padlock icon when a connection is secured using SSL/TLS. The padlock icon tells the user that they are safe, an assurance that is usually unjustified.

Attempting to teach the typical user the distinction between safety and the mere use of cryptography is an exercise in futility. Any indication that tells the user that a connection is encrypted is going to be

interpreted as a statement about the trustworthiness of the certificate subject regardless of any disclaimers in the instruction manual.

The use of an EV certificate establishes a degree of accountability, a condition that most Web browsers recognize with some form of 'green bar' security signal. But it is still left to the user to know if they should expect a padlock, a green bar or nothing at all when visiting each site.

Audit and Transparency

For a security policy to be effective it must (1) be capable of preventing or mitigating the unintended outcome and (2) be followed. The primary purpose of audits and transparency is to determine if practice complies with policy.

An auditor is expected to be a 'watchdog not a bloodhound'. While an audit report is frequently required for compliance purposes, auditors are hired by the businesses they report on and their primary role is to advise their customer rather than third parties. While most auditors certainly take their responsibility to third parties very seriously, audit practices have been developed to detect malpractice by low ranking employees who are corrupt or lazy. There is little an auditor can do when their customer intentionally deceives them.

No audit is complete. Financial auditors examine a representative selection of invoices and payments in detail and the processes used to verify them rather than looking at every single receipt. A statistical approach is much less effective when the concern is security and a single failure or a small number of failures can create a vulnerability.

Even the existence of an audit does not mean that that it is relevant to the system or the uses at issue. The Diginotar CA at the center of the recent incident giving most concern had passed the audit required by the Dutch government for issue of certificates within their system but that audit did not cover the commercial CA that was breached (9).

Verifying some aspects of a security policy requires access to information and systems that are not otherwise public. It is not practical, nor is it desirable for every Internet user to personally verify aspects of the CA operation that rely on physical access or confidential information. It is not possible for every Relying Party to make an on-site visit to inspect the physical cabling of the firewall. The role of an auditor is thus a trusted third party role and we are thus forced to rely on one trusted third party to verify the compliance of another.

A security policy requirement is made *Transparent* when compliance with that requirement may be verified without recourse to non-public sources or an additional trusted party. It is not possible for every aspect of the operation of a Certificate Authority to be made transparent but achieving transparency is likely to instill a greater degree of confidence in the operation of the CA amongst Relying Parties and Subjects.

Another concern that arose in the DigiNotar breach incident is that the Certificate Authority lost control of both the Certificate Signing Keys and the audit logs recording certificate issue. This condition was not visible to Relying Parties despite the fact that the Certificate Authority was answering queries for the status of certificates that it had no record of having ever issued.

Technical Transparency Controls are typically designed to alert concerned parties to the possibility of a breach rather than the exact nature of the breach. In the ideal case a technical transparency control is based on a cryptographic technique that provides a virtually irrefutable level of proof but this ideal is only sometimes achievable and may only cover some aspects of the operation of the system under examination.

Procedural Transparency Controls are enforced through contractual agreements and regulation. Various jurisdictions have enacted data breach notification laws that require prompt notification to persons whose personal data has been improperly disclosed. Fines and other penalties imposed for failure to comply provide a high degree of compliance even in the case that the breach might otherwise have remained undiscovered.

One of the key controls that safeguards the integrity of an audit is a Procedural Transparency Control. Even though a customer could deceive their auditor, to do so may constitute an act of fraud. Knowingly using an audit obtained in this manner to secure business almost certainly would.

'Whistleblower' statutes in the US and other jurisdictions provide a form of Procedural Transparency Control. Such statutes may apply if a party has agreed to a voluntary disclosure requirement and is intentionally concealing the relevant information.

Security Analysis

The Internet Trust Infrastructure represents a balance between security, commercial and practical considerations. The principal evidence that at least some of the design decisions taken were correct is that SSL/TLS and code signing are the only Internet security solutions that have achieved widespread use.

SSH and IPSEC have become established as *intranet* security solutions, but neither protocol is in widespread use in an open community of users. Nor have any of the end to end email security solutions proposed achieved widespread use despite the ready availability of clients.

Downgrade Attack

The biggest security problem in the SSL/TLS infrastructure is that security is optional and every use of a security enhancement is subject to downgrade attack.

Most sites advertise the use of TLS by means of a HTTP redirect that references a HTTPS URL. An attacker with the ability to perform a man-in-the-middle attack can defeat the attempt to upgrade to TLS by intercepting the redirect request and acting as a proxy.

At present the responsibility for defeating such attacks rests with the user who is expected to look for a padlock icon placed in a position of the browser developer's choice. It should not take a usability study to understand that this approach is doomed to fail.

Mis-issue

A certificate mis-issue occurs when a CA issues a certificate that does not meet the requirements of the stated Certification Policy.

A mis-issue may occur as a result of:

- Subject issues
    - The Subject made a material misrepresentation during the certificate request and validation process
    - One or more material representations made during the certificate request and validation process are no longer true
    - The Subject is breached resulting in disclosure of the private key.
- A breach at a Registration Authority
- A breach at a Certification Authority

Mis-reliance

For a mis-issue event to result in mis-reliance, two further failures are necessary:

- The attacker must cause the relying party to connect to the fraudulent site.
- The relying party application must fail to discover that the mis-issue event has occurred.

Mis-connecting traffic

Mis-connection of Internet traffic may be effected by :

- A network layer man in the middle attack.
- Redirection of the Internet routing layer (i.e. a BGP attack)
- Redirection at the Internet naming layer (i.e. a DNS attack).

Revocation failure

A revocation failure may occur because:

- The mis-issue has not been detected and the certificate status is reported as valid.
- The CA does not provide revocation data
    - The CA never provides revocation data
    - The CA Database has been corrupted and the CA is unable to determine that a certificate was issued.
- The application did not check revocation data.
    - The application never verifies revocation data.
    - The application usually verifies revocation data but will accept certificates when the revocation data is unavailable.

Detection and Reporting

A breach event may not be reported because:

- There were no controls in place to detect possible breach activity
- The breach detection controls failed to distinguish the breach activity from normal operation
- The breach detection controls detected the breach activity but the notification of the event was ignored
- The breach detection controls detected the breach activity but the records of the event were deleted by the attacker

Deployment Challenges

Proposals to change the Internet Trust Infrastructure face two challenges. The first challenge is to design a technical proposal that offers clear advantages over the existing system. The second challenge is to deploy the new proposal.

Experience suggests that the second problem is considerably harder than the first. DNSSEC and IPv6 were both in development for over a decade and it is highly unlikely that either deployment will be complete by 2016 which mark over twenty years since the first proposals were circulated.

Deployment must be a major consideration in the design of any proposal to change the Internet infrastructure because it is the biggest challenge.

Downgrade attack

As previously discussed, the default state for the Internet is 'insecure'. Security enhancements can only improve security if the relying parties know to expect them.

Vulnerability to downgrade attack need not disqualify a particular solution if it is capable of being employed in conjunction with a mechanism that does provide the necessary protection against downgrade attack. We should not however accept a proposal that provides protection against specific forms of downgrade attack that is itself vulnerable to a downgrade attack.

Joint Action

The Internet has over two billion users and many times that number of applications and devices. The feasibility of a proposal depends in large measure on the number of stakeholders that are required to act in order for a benefit to be realized.

Many Web browsers have adopted some form of automatic update feature. A proposal that only requires changes to be made at the Web browser can be deployed very quickly to up to half the installed base. Proposals that provide an immediate benefit to the user of a Web browser can thus deploy very quickly.

There is however a 'long tail' issue: Proposals that rely on ubiquitous adoption of a particular feature take much longer to become viable since a significant number of Web users keep using the same Web

browser for ten years or more. The number of Internet users without broadband is still large and is likely growing in terms of numbers of users even if it is falling as a proportion of users.

Changes to Web Server infrastructure take considerably longer to deploy. A Web Server is typically a single component in a system with many software modules and complex interdependencies. Web sites and Web Services frequently employ custom code that has a complex relationship to the platform software. Deploying and testing updates to such systems can be an expensive and time consuming task.

Proposals that require changes to both browsers and Web servers face an even more difficult challenge. The Web browser providers are liable to wait for server providers to deploy changes that make their changes useful and vice versa. Such joint-action problems are essentially a form of deadlock.

DNSSEC

DNSSEC(10) is a powerful security enhancement to the existing DNS infrastructure. It is not however a panacea. In particular DNSSEC suffers from two important drawbacks.

The first drawback is that DNS clients traditionally make use of the closest DNS resolver, even if that resolver is not trusted. While the latest revisions of the DNS specifications require DNS resolvers to pass DNSSEC records to a DNS client, a significant proportion of deployed resolvers do not. In addition DNSSEC records may be modified by DNS proxies at firewalls and NAT boxes.

As a consequence, a DNSSEC client is that does not have access to a trustworthy resolver is able to distinguish a downgrade attack from lack of support for DNSSEC at the local resolver.

The second drawback is that the root of trust in DNSSEC is the DNS root controlled by ICANN and thus raises concerns about concentration of risk and control as the PEM scheme did. Unlike the PEM hierarchy, the DNS is a deployed infrastructure that is already a focus of international political concern.

While the current arrangements for administering the DNS root affords the US a uniquely privileged and capable role it is a limited one. While the US government can in theory pass a law that requires ICANN to act in pa particular way, doing so would almost certainly result in a ‘fork’ of the DNS root and an end to the privileged US position. Deployment of the current DNSSEC architecture based on a single (US controlled) root of trust changes the nature of US influence on the DNS root from being a uniquely privileged one to absolute control.

US control of the DNS system is arguably a more potent security threat to other countries than the earlier US monopoly on GPS navigation systems. Russia, India, China, Japan and the European Union have all developed alternative navigation infrastructures to avoid dependency on the US system(11). Japan and the European Union are investing billions of dollars to build a system whose sole purpose is to avoid dependency on an ally.

The US government has already made use of its de facto control over the DNS infrastructure to seize DNS domains(12) through procedures that provide for little if any effective judicial oversight. It is only to be expected that when the US attempts to make use of its privileged position with respect to control of

the DNS to pursue what it sees as the US national interest that other countries will attempt to do likewise.

Despite these drawbacks, the DNS is the authoritative infrastructure for assertions about DNS names. It is thus the best and most appropriate infrastructure through which a DNS name holder may publish assertions about a DNS name even if it may not be possible for every DNS client to access that assertion directly.

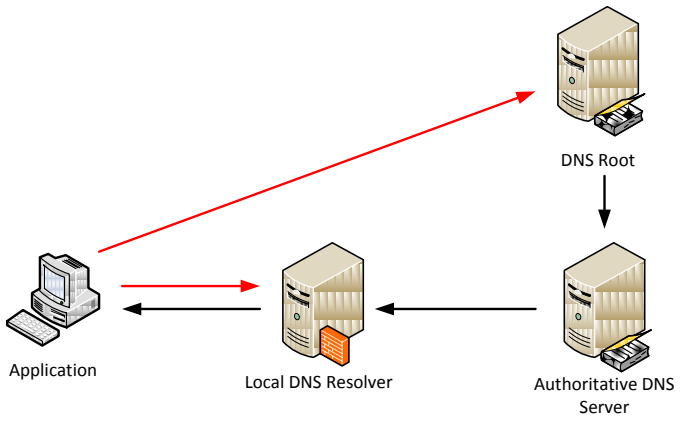

Figure 11: DNS Trust Relationships

These trust issues are depicted in Figure 11. The application receives authenticated data from the Authoritative DNS server (black arrows) but relies on the DNS Resolver and DNS Root to construct a valid trust relationship (red arrows).

The trust concern that arises from the concentration of control at the DNS root is similar to the concern that arose in the design of the US Federal Government PKI. The relying parties do not have any immediate expectation of default but are put in a position where they have no exit strategy should a default occur in the distant future.

One approach to addressing this concern is to abandon the somewhat peculiar convention whereby Internet hosts take DNS service from the nearest DNS resolution service and instead require that this trusted role be performed by a chosen, trusted DNS resolution service. This service would then be responsible for curating all aspects of DNS trust, including the delegation of trust to the DNS root (Figure 12).

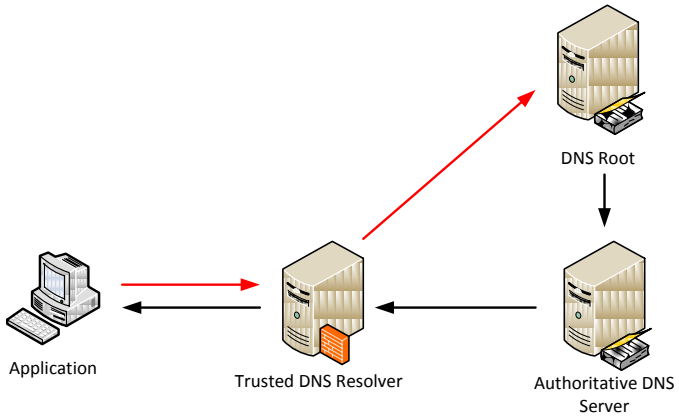

Figure 12: Trusted DNS Resolver

This approach is analogous to the way that each Federal Bridge member agency regards itself as its own ultimate root of trust but delegates the task of de-facto root to the Bridge CA *for a renewable, time-limited period*.

Education

End users have many priorities and reading instruction manuals is certainly not one of them. Proposals that rely on end users investing learning new technical skills that are not directly relevant to their concerns are certain to fail.

Proposals

The past 24 months has seen a renewed interest in new PKI architecture and a series of proposals designed to improve or supplement the established CA based ITI, in some cases replacing it entirely.

Security Policy

As previously discussed, the threat of downgrade attack exists because Internet Applications have no trusted means of determining that an Internet service offers any security enhancement(s) or the configuration of the enhancement(s) offered. Security Policy mechanisms provide a means of informing a client that a service offers a particular security enhancement. A wide variety of Security Policy proposals have been submitted to the IETF WEBSEC Working Group (13), Ritter (14) provides a more detailed survey.

Security Policy proposals range from addressing one particular aspect of security to advertising a specific security policy to generalized frameworks capable of expressing arbitrary security policy statements. The principle aspects for which security policy have been proposed for TLS include:

Use of TLS

: A Security Policy that the ability to connect via TLS is always offered as an option or that use of TLS is required.

Strict Transport Security (15)

A HTML document retrieved via HTTPS may contain URI references that do not carry a security requirement. Depending on the nature of the reference and the context in which it is used, this may or may not introduce a security vulnerability and may or may not be detected and subjected to appropriate handling by the Web client. A strict security policy mitigates these and similar. HTTP Strict Transport Security (HSTS) is a security policy specific to Web servers that declares that complying user agents (such as a web browser) are to interact with it using secure connections only (such as HTTPS).

Certificate Restriction (including CA Pinning) (16)

A Security Policy that states that only particular certificates or certificates issued by a specific issuer should be accepted as valid TLS certificates for that site, even if there is an otherwise valid trust chain to a trusted root certificate.

Certification Authority Authorization (CAA) (17)

CAA is similar to Certificate Restriction except that the Security Policy is directed at and applies to the actions of certificate issuers rather than client applications. An Internet Service specifies the certificate issuers that are authorized to issue certificates for a domain.

Note that a security policy mechanism need only address the limited range of security policy concerns that are not already controlled in the TLS protocol. Downgrade attacks such as cipher substitution and version downgrade (at least for recent versions) are already addressed by controls within the TLS protocol and other security protocols.

Various mechanisms for distribution of security policy have been proposed/employed:

Preconfigured Application Data

Security Policy data is frequently embedded into the application itself as static data. For example, most Web Browsers have code to ensure rejection of a limited number of certificates known to have been mis-issued as the result of an attack. This approach generally requires that the application at least supports a mechanism for periodic updates.

Server assertions

Distributing security policy by means of application headers avoids dependency on any other infrastructure but has the considerable disadvantage of only offering security for *subsequent* connections. The visitor to example.com can be told to expect and apply a certain security policy to future connections but cannot obtain authenticated security policy for the current connections.

DNS Records

The DNS is the Internet infrastructure for making authoritative assertions relating to DNS names. It should therefore be the infrastructure for making authoritative statements about security policy bound to a DNS name.

Other Online Service

While the DNS is the authoritative infrastructure for originating statements about a DNS name, the DNS protocol is not necessarily ideal for distribution of other aspects of security policy, in particular third party assertions relating to a domain. There is thus a valid case to be made for the use of other online services.

DNS-based Authentication of Named Entities (DANE)

The IETF DANE Working Group (18) is in the process of developing a mechanism that uses the DNS as the basis for binding keys to DNS names and declaring a certificate restriction security policy.

In the current DANE proposal, the two functions are linked. The use of one by necessity implies the other. To be useful for either purpose, DANE records must be validated using DNSSEC.

DANE is thus positioned to provide a replacement for the existing CA infrastructure as opposed to a supplement. The risk of CA default is replaced by the risk of default by the Registrar, Registry or Root manager.

While the DANE documents assume that the only possible Root manager is ICANN and the possibility of default by ICANN is small, both assumptions may be reasonably regarded as naïve. Russia and China have established an international treaty (the Shanghai Cooperation Organization or SCO) pledging both countries to joint action to prevent the US extending its 'dominance' in cyberspace. DANE anticipates ICANN acting as the defacto root of the global PKI. It is hard to see how such a proposition would be acceptable to the SCO member countries.

Another concern is that DANE anticipates ICANN and the TLD registries acting in the role of CAs for a global PKI without any indication that either is willing to accept such a role and the liabilities it would entail.

Perspectives

In the Perspectives proposal (19) the relying party selects at least one 'notary server'. Each notary server is connected to the Internet and continuously monitors the deployment of SSL certificates at Web sites. When connecting to a Web site, a client checks the certificate presented by the site for consistency with the prior observations by their selected notary servers.

While the notaries play a trusted role, the role of the notaries is constrained by the use of an append-only notary protocol. Regular checkpoint events are inserted into the notary log and cryptographically bound to all prior notarization events so to prevent insertion of notary events without this becoming apparent and to ensure that all valid notarization events may be verified, even in the case that the notary ceases to function.

Even though the Perspectives notary has a highly restricted trust role, it is nevertheless a trusted party in that it may refuse or be unable to provide service. A key weakness in the Perspectives approach is the lack of an apparent business model for the providers of the notary services on which the system depends.

While open source software has proven to be an effective and successful resource, the marginal cost of the product in question is almost exactly zero. The marginal cost of supporting software as a service is small but rises in direct proportion to the number of users. Since the SSL infrastructure serves a user base of two billion users and millions of certificate holders, it is only likely to succeed if an existing provider of software as a service decided to do so for commercial reasons of its own.

Convergence

Convergence (20) improves on the Perspectives approach by addressing a practical weakness in the use of multiple notaries. Instead of requiring the client to perform separate communications with each of the notaries selected for use, these communications are all directed through a single notary chosen to act as a gateway to the others.

Although this approach reduces the latency introduced by making a notary query, the Convergence approach is still slower than the traditional approach. Nor is the question of a business model for the notary services addressed.

The principle reason that Convergence cannot be seriously considered at present however is that almost a year after the initial announcement there is neither a substantive description nor a protocol specification. It appears that other developers are expected to reverse engineer the prototype code.

Sovereign Keys

Sovereign Keys (21) provides a form of certificate pinning reinforced by strong cryptography. A Sovereign Key is created by adding a digest of the key data to a verifiably append-only notary structure.

The principal advantage of sovereign keys over basic certificate pinning schemes is that it provides a defense against downgrade attack since the declaration of the pinning security policy is authenticated by tamper-resistant cryptography.

A major defect is that a certificate subject who loses access to their private key may have no means of establishing a new one. In effect Sovereign Keys ‘solves’ the problem of ensuring that a PKI is trustworthy by ignoring the practical difficulties that make the problem hard.

Certificate Transparency

Certificate Transparency is an emerging proposal made by Ben Laurie et. al. (22). While Certificate Transparency has the potential to provide a client enforcement capability, the objective is to improve rather than replace the existing certificate issue infrastructure by making issue more transparent.

In particular Certificate Transparency is designed to address the situation discovered in the DigiNotar incident where the attacker successfully deleted records of the mis-issued certificates being created. Due to a defect in the OCSP protocol, the Certificate Authority was not even required to distinguish between the status of a certificate being unknown and there being no record of the certificate being issued.

A Hybrid Approach

In contrast to the existing proposals, the goal of the Omnibroker protocol is not to provide 'a' solution to 'the' problem of establishing and maintaining Internet Trust but is instead designed to provide a platform that allows multiple approaches to be deployed, tested and used.

The approach that is most appropriate for a senior who began using the Internet at seventy is not likely to be the same as the approach most appropriate for a longtime Internet resident. The security needs of a consumer doing Internet banking are not the same as those of a dissident protesting their government which are in turn very different from those of a 12 year old.

Furthermore the prospects for deployment of any new protocol scheme will depend on there being a compelling deployment proposition for each party that is required to take action for deployment to occur. Changes to code are easy, changes to behavior are much harder to achieve.

Perspectives, Convergence, Sovereign Keys and Certificate Transparency each try to constrain the behavior of the Certification Authority through the introduction of notaries whose activities are in turn cryptographically constrained to prevent default.

Instead of constraining the Certification Authority in a specific fashion, the Ombnibroker protocol introduces a new party that is chosen by the Relying Party to constrain the trust provider according to their specific requirements.

A Four Corners Trust Model

Like a Certificate Authority, an OmniBroker is a trusted party. Unlike a Certificate Authority however, the OmniBroker is chosen by the Relying Party rather than the Site being accessed.

The introduction of the Omnibroker addresses one of the principal criticisms of the current trust infrastructure; that the trust providers are insufficiently accountable to the Relying Party. The Omni Broker is chosen by the Relying Party who may change their selection at any time. The Omni Broker is thus accountable to the Relying Party and will suffer consequences if they do not perform in an acceptable fashion.

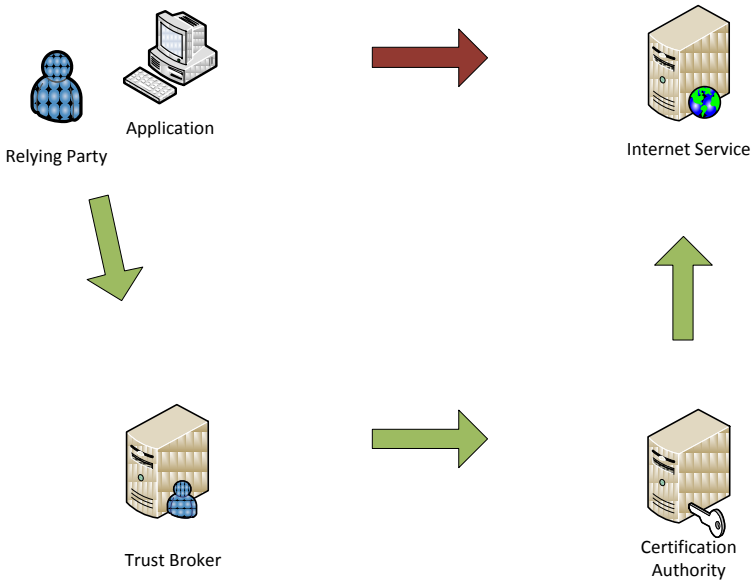

Figure 13: Omni Broker Four Corners Model

This trust model is similar to the 'four corner' model used in the payments industry. In the four corner model of financial transactions the customer and the merchant are each represented in the transaction by a bank that they have selected. The customer's bank is known as the 'issuing bank' and the merchant's as the 'acquirer' (because the bank acquires the customer's credit card information during the transaction).

It is natural to expect that any communication system requiring management of trust will evolve into some form of Four Corners model since a communication must involve at least one originator and at least one receiver. When both parties are represented by an agent, a Four Corner model emerges.

A Four Corners model is powerful but faces a considerable deployment challenge, particularly when a completely novel infrastructure is being deployed. While the credit card payment system is a four corner model, only three of those corners are involved in the Internet payments infrastructure; the customer merchant and acquiring bank. The issuing bank is not normally represented and attempts to introduce the issuing bank through smart tokens, security codes and other technologies have proved unsuccessful.

While Four Corner models have emerged in practically every communication system that has been operated for an extended period of time, each of the major four corner models has emerged from an earlier three corner model.

The early telephone system was a three corner model in which subscribers could only talk to other subscribers on the same system. The four corner model emerged from the need to connect independent telephone networks. While the early operators frequently avoided such interconnection within towns, the obvious benefits of reaching different cities and different countries made this inevitable.

The first card payment systems were two party systems. Customers could use their store card only in the department store that issued it. The first modern credit card was invented by Dinners Club as a

means of allowing restaurants to offer their business customers a convenient payment method. The Dinners Club model was three cornered, the fourth corner did not emerge until banks started to issue credit cards and recognized that they needed to adopt a federated model if they were to successfully challenge the market leaders.

Protocol Scope

Having determined the need for a fourth party, what should be its scope of operation? Should the scope be limited to 'trust' decisions or be broader?

Traditionally protocol designers are advised to develop protocols that are as narrowly focused as possible. It is asserted that the broader the scope of a project, the lower the chances of success. Such advice flies in the face of the empirical evidence. While most broadly scoped projects fail, this is no less true of those that are narrowly scoped.

The most successful Internet infrastructures are those like the Web, XML and SSL, specifications that are designed to address broad, ambitious goals. Such proposals can attract the broad base of support required to change the Internet infrastructure because they are attractive to a wide constituency. Narrowly focused, timid proposals that only offer benefit to a few are much more likely to fail.

Rather than imposing arbitrary limits determined by the current problem we wish to solve, the protocol scope should be determined by first identifying the interaction pattern appropriate for the role that the broker is intended to play and then enumerating all the possible use cases that might be met by such a party.

Interaction Pattern

The Relying Party may obtain information from their fourth party broker directly or indirectly through either the service or their CA. The choice of interaction pattern must meet both the trust and the performance needs of the Relying Party.

A core requirement for the fourth party role is that the Relying Party be able to choose any broker they wish without constraint. If the communication between the relying party and their broker is gated by another party such as the service or the CA, that party has the ability to exercise a veto power. It is thus essential that the Relying Party be able to communicate with their chosen broker directly.

A key constraint imposed by a major browser provider is that any new protocol should not incur any more latency than existing HTTP and HTTPS transactions in the typical case. Since this baseline for comparison does not include waiting for an OCSP transaction in either case it is only possible to introduce a fourth party trust broker if that functionality can be combined with an existing service. Since the only other service to which queries are directed is the DNS, it follows that a fourth party trust broker service must at minimum provide both name and security services.

Special Scope

Although resolution of names and keys are supported by independent infrastructures, both are trusted and both should be delivered through trustworthy channels.

In the traditional Internet architecture the client is instructed to visit an Internet address (e.g. example.com) using a specific protocol (e.g. http) and security properties (e.g. ssl chained to an embedded root). Combining the name and security lookup permits the client to hand over all the responsibility for forming an Internet connection to the broker, optionally retaining ability to verify the result.

In addition to being simpler to implement on the client, combining the name and security lookups allows the Omnibroker to apply additional information to protect the client:

- Block access to DNS names that are believed to be controlled by parties considered to be malicious
- Block access to IP addresses that are believed to be part of a botnet or otherwise likely to be malicious
- Instruct the client to require use of security enhancements (e.g. SSL)
- Instruct the client which cert chain restrictions are operative (e.g. must be key X or Y, must be issued by Comodo Inc.)
- Instruct the client that a particular client authentication mechanism should be employed (OAUTH, DIGEST, etc).

The special Omnibroker protocol answers the following query:

How should party X establish a trustworthy connection to Internet service Y?

The response consists of an IP address, a protocol and a set of security constraints.

General Scope

The traditional Internet architecture is a client-server architecture. Clients use the DNS to resolve servers. This approach was adequate when the party responding to a user connection request was a machine or service. It is not adequate when a user is attempting to connect to another user who may be using any one of a number of machines. DNS names bind to Internet services and not Internet users.

While several 'presence' infrastructures have emerged to fill this need, there is no single authoritative infrastructure as there is for DNS names and client applications are required to address these needs in piecemeal fashion.

The broker is logically well placed to provide a gateway to this presence infrastructure in the same way that it provides a gateway to the name and trust infrastructures. The general Omnibroker protocol answers the following query:

How should party X establish a trustworthy connection to party Y?

In addition, the general Omnibroker protocol allows an Internet Service to assert a claim to offer an Internet service for a specified party.

Implementation of the general Omnibroker protocol is slightly more complex than the first as the broker becomes a supplier of information and not just a consumer.

Additional Scope

Having extended the role of the Omnibroker to include brokering inbound connections as well as outbound, a question occurs: Where should the scope of the protocol end?

For efficiency and security it is highly desirable for the connection between a client and an Omnibroker service be protected by means of mutual authentication. The Omnibroker is thus excellently placed to provide a gateway to an authentication service such as SAML or OpenID. Moreover, since client authentication criteria are one of the security constraints that an Omnibroker might report it is natural to expect that the Omnibroker might facilitate meeting them.

An Omnibroker might even provide a hosted decryption key capability (with appropriate security safeguards) that would allow S/MIME encrypted email to be delivered in a form that could be read on a wide number of devices without releasing the key itself to those devices.

Business Model

Any proposal to introduce a new party into a model must provide a viable business model for providing that service. While virtually all Internet Service Providers provide facilities such as DNS, email and Web hosting as part of their service offering, these are all services that have been regarded as basic Internet service for over 15 years. Almost no ISPs have offered new services as part of their standard service and virtually no new services have been offered until after demand is proven.

As with any Internet Service, the two principal business models are supported by advertising and as a paid service. Precedents for both business models exist.

There is an existing market for advertising supported DNS name services. The Omni Broker service provides existing providers with an opportunity to enhance and extend their product.

There is also an existing market providing paid security services to consumers. While these are known in the trade as ‘Anti-Virus’ products consumers regard them as security products and expect to be protected against all threats rather than specific types of threat. When non-Virus based mal-ware (e.g. spy-ware) began to circulate, consumers demanded that their Anti-Virus solution eliminate all unwanted code on their machine, including code that the authors asserted to be legitimate.

The Omni Broker protocol provides a new opportunity for ‘Anti-Virus’ providers to provide protection to their customers that is complimentary to and a logical extension of their existing product. As distribution of email borne viruses becomes more problematic, attackers are increasingly turning to the Web to deliver the malware payload. The Omni Broker protocol provides an opportunity for Anti-Virus providers to advise their customers that they should not visit Web sites at suspect IP addresses, DNS names or with compromised certificates.

Data Sources

An Omnibroker provides a gateway to a variety of data sources providing name data, certificates, certificate status and security watchlists. For each of those data sources the information provided may be authoritative statements that are explicitly asserted or empirical data that is inferred from observations.

Both types of data are necessary and neither should be considered intrinsically superior. It is preferable for data to be accurate, trustworthy and authoritative but this is not always possible. An authoritative DNS entry may still be malicious or erroneous. At present there is no infrastructure for expressing security policy data and so the only means by which such data may be obtained is by observation.

An Omnibroker makes use of both types of data applying empirical observation to curate authoritative assertions.

At present there are no standards for expressing Internet security policies, only proposals to deliver such policies through HTTP headers or DNS records. Once Ombinbrokers are deployed and used, there will be a constituency capable of making use of this information and hence a much greater incentive to assert authoritative security policies for sites.

Until that point is reached however, Omnibrokers will be obliged to rely on inferred security policy data. If a Web site has been observed to redirect HTTP connections to a HTTPS connection, for an extended period of time, it may be reasonable to infer that browsers should attempt to connect using SSL first. If the site is also known to be a likely target of phishing attacks it may be desirable to infer that this should be mandatory.

Client Interpretation

An Omnibroker is an advisor, not a dictator. The Omnibroker can advise the client that a Web site is not safe to visit or even refuse to provide the information necessary to connect but the user retains the ability to choose a different Omnibroker. A technical challenge for the application designer is how to ensure that the user's consent to use a particular Omnibroker is informed consent.

Discovery vs. Reliance

Depending on the security needs of the user, the client may rely on the Omnibroker as being entirely trustworthy or merely use the Omnibroker to collect data which shall be re-checked before use. These two modes correspond to the 'path discovery' and 'path validation' modes of SCVP.

While the 'path discovery' approach permits the client more control it does not necessarily afford the user more security. The client application has less information on which to base a decision and can only apply processing rules that are pre-programmed and thus static. The path validation model allows the Omnibroker to address the user's security needs dynamically.

While a client could in theory delegate all trust decisions to the Omnibroker, in the normal case the role of the Omnibroker is limited to negative trust assertions; reporting that a certificate has expired or that the site is suspected of malicious activity. In an idealized model, the Certificate Authorities and DNS

registries make positive trust assertions that are subject to veto by the Omnibroker. In this model, the roles of the positive and negative trust broker are distinct, complimentary and encourage mutual accountability.

In practice the natural role of the Omnibroker includes making statements that are traditionally considered to be 'positive' trust assertions. When site offers a 'self-signed' certificate for use with SSL, this is considered to be a positive trust assertion, even though it is more appropriately considered to be neutral. A similar problem arises with the use of positive trust assertions that are presented in a form that the client is unable to process: Should an Omnibroker be able to tell the client to trust a certificate asserted by means of a DANE assertion?

Multiple Brokers

The exceptionally security conscious user might even employ multiple brokers, issuing queries to both and cross-checking the responses against each other. Alternatively an Omnibroker may act as a proxy for one or more other Omnibrokers, making use of the data provided as a basis for their own decisions.

Enterprise case

An individual surfing the Internet from their home has the right to use the security. Should an employee have the same right at work or should the Enterprise decide the security policy for their own network?

As with many security questions, this answer depends on circumstances, policy and opinion. There is no right or wrong answer. While the Omnibroker protocol is not designed to facilitate a particular solution to this use case, it is not designed to prohibit one either.

Employees are paid to work rather than surf the net and so Internet access at work is a privilege, not a right. Therefore an employer may assert a right to determine the security policy under which the employee gains access to the Internet, if this is permitted at all. Such a security policy may limit or even deny outright the right to communicate in confidence.

What is never acceptable in any such circumstance is a covert intercept capability. The use of covert intercept capabilities is reserved for use by law enforcement in virtually every jurisdiction. Attempting to deploy such a capability without the informed consent of employees is likely to expose the employer to civil liability and possibly criminal charges.

Client Response

After receiving direction from the Omnibroker, a client may respond as follows:

Connect to site with accountability accreditation (aka Green Bar)

> A site should only display positive accreditation for a site if it has verified that all the conditions for that accreditation are met. Although an Extended Validation accreditation is presented through means of a certificates, it is the accountability of the subject that is worthy of note rather than the use of an encrypted transport.

Connect to site with no accreditation

In the normal case, the client should connect to a site using the best available transport and this should not necessitate any further comment on the part of the client.

Connect to site with warning

A client may connect to a site but warn the user that doing so may be unsafe. While this approach may appear to be 'safety conscious', the value of doing so is likely negligible to none. Users quickly learn to disregard warnings that become commonplace.

Connect to site with warning and reduced capabilities

Rather than connecting to a site with a merely warning, it may be more appropriate to limit the capabilities of that site, disregarding embedded objects or content downloads.

Refuse connection to site

If a site is determined to be malicious, the user probably does not wish to visit it at all.

Report security policy violation

Should a client discover that the security practice at a site differs from that described by the Omnibroker, the discrepancy should be reported and further directions requested.

Report intercept capability

In the case that a client permits the use of an SSL interception capability, the user should be advised that this capability is in use prominently and repeatedly.

It should be noted that the display of the 'SSL security signal (aka padlock icon) is not listed. The purpose of the padlock icon is to enable the user to check that a Web session is encrypted. This security signal is highly problematic as the interpretation justified by the protocol (transport is encrypted) is very different to the meaning that the user is told to infer (I am safe). The Omnibroker protocol enables the use of Security Policy to determine whether a connection should employ SSL or not and may therefore eliminate the need for the padlock icon entirely.

Protocol Implementation

Work is on implementation of the Omnibroker protocol has begun. This section describes the current approach.

User Experience

Since the normal operation of the Omnibroker should be transparent to the end user, the user experience has visible three parts: First the user decides to establish an account with a broker 'subscription', next the user decides to make use of that service on one or more devices. Finally the user may modify the parameters of their service from time to time.

Subscription and Account Management [Out of Scope]

The user establishes an account with an Omnibroker and is issued an account name that follows the traditional email format (e.g. hallam@omnibroker.com).

In the typical case the subscription interaction will be supported through a traditional Web interface that allows the user to choose subscription options. An Omnibroker may offer subscribers a range of security offerings with different service options and different devices and/or accounts within a subscription may have different options applied. For example a parent might decide that a child was permitted to access educational web sites but not those that involve interactive gaming Once the initial security settings are established, the subscriber may modify them from time to time.

The security options offered and the means of selection by the user are outside the scope of the protocol. The only aspects of the subscription mechanism that are specifically relevant to the protocol are that there a subscription comprises:

- An account identifier (represented by an email address)
- A set of policy identifiers (represented by text labels)
- A set of device identifiers (represented by text labels)

The broker must also establish some means of communication with their subscriber in order to permit future device binding requests to be supported. These may include:

- An email address
- A one-time use passcode
- A password

Device Binding

Having established a subscription, a user must bind one or more devices to that subscription. This process may be performed by a 'plug in' application or supported by the operating system.

To bind a device the user must supply their account identifier. The binding request is then verified by one of the following means:

- Account password
- One-time use password obtained from the subscription interface
- Email callback confirmation to the email account associated with the subscription
- Confirmation made by accessing the subscription interface from a previously bound device

It may be desirable to allow a device to be bound for a limited period of time. For example a user at an Internet café might wish to have access to their usual security policy settings without making them available to other users at the café after they leave.

Automatic Subscription

It may be desirable to support a single step process that allows the first time user to achieve subscription and their first device binding in a single step.

In this approach the user would provide the DNS name of the Omnibroker they wish to use, a requested account name and (optionally) an email contact address and/or account password.

If supported, this operation would be an extension of the device binding operation.

Operations

Bind Device

Device binding is requires mutual authentication. The user must be assured that they have indeed bound to the Omnibroker service that they intend and if the Omnibroker is to support features that require account level authentication, the client must also be authenticated.

For obvious reasons, the client cannot rely on the Omnibroker service for name or trust services during a device binding request. Thus the client must rely on the traditional mechanisms of DNS queries to the locally provided DNS resolver and embedded roots of trust. This in turn means that it is most appropriate for the device binding operation to be supported through a HTTP REST Web Service over an SSL transport that is separate from the Session and Transaction operations.

When first requesting a device binding operation the client supplies the following information:

- Account identifier (required)
- Device/Account public key (required)
- Device recognition assistance (optional)
    - Image
    - Nickname
    - String

The broker service then determines the mode of authentication to be used. If in-band authentication is offered, the service supplies parameters to support an appropriately secure password verification mechanism. Otherwise the client is told that it should wait until the request has been verified out of band. If out-of-band authentication is used the Device recognition assistance data is presented to the user to assist them in verifying the origin of the particular request.

In the second phase the client requests completion of the device binding request. If password authentication is selected, the client provides the proof of knowledge of the password. The request is authenticated under the device public in either case.

At the completion of the Bind Device operation the client has established the following persistent state data:

- Device/Account public key (self generated)
- Session parameters
    - Creation date-time
    - Expiry date-time
    - Session authentication key, Session encryption key
    - Security policy name
    - Transaction service connection information

Session

Having achieved device binding, a device may be required to perform periodic exchanges with the Omnibroker service to refresh the session authentication key and associated contact information. This 'session refresh' is performed automatically on either the expiry of the old session information or at the request of the broker in a transaction response.

Transaction

The following transactions are currently being defined:

- Connect to service
- Advertise service
- Credential request
- Property
- Certificate status request (OCSP tunneling or status conclusion)

Transport

The Omnibroker transport protocol must need two conflicting requirements:

- Latency must be as good or better than existing protocol in almost all circumstances
- Availability must be as good or better than existing protocol in all circumstances.

When considering the user experience, the key figure of merit is the proportion of unacceptable or poor connections rather than the average or typical response. A browser that usually responds within a nanosecond for 99% of cases but takes 10 seconds to respond to the remaining 1% has the same speed of response as a browser that always responds in exactly 0.1 second but is far more frustrating to use.

Examining the transport options it becomes clear that no single option can meet both criteria. To meet the latency requirement, a UDP based transport is required but many NAT boxes and firewalls block UDP transport other than DNS which is frequently limited in other ways. To meet these

UDP Protocol

: A new protocol in which transactions consist of a single request and at least one response packet. This transport is capable of efficiently supporting clients that require evidence or proof but can only be used in circumstances where the firewall or NAT configuration allows.

DNS TXT record

: Requests are mapped to DNS TXT record queries and responses to DNS TXT record data segments. Since DNS responses are limited to a single packet (and often to 512 bytes), this transport can only be used where the client fully trusts the Omnibroker and does not require evidence or proof to be returned. It is also limited to use on networks where the firewall or NAT configuration permits queries to arbitrary DNS resolvers or the local DNS resolver returns TXT responses.

Web Service

The Web Service transport is a REST style Web Service layered on HTTP. Requests are mapped to HTTP URIs and responses returned in the same syntax as for the UDP protocol. Since the response messages may be of arbitrary size, this protocol supports the use of evidence or proof. As the transport is layered on HTTP, it may be used in any local network where the firewall configuration permits general Internet access and is not explicitly configured to block these particular queries.

Syntax

There are four options for the response syntax:

- ASN.1

  Sufficiently efficient but tiresome to use and has poor buy-in.
- XML

  Insufficiently efficient unless binary encoding is used which has poor buy in.
- SSL Encoding

  Sufficiently efficient, well defined but no toolset.
- JSON

  May be sufficiently efficient.

At present it appears that use of the JSON format is likely to prove most satisfactory.

Conclusions

The Internet trust infrastructure has been very successful at solving the problems it was originally intended to solve. Despite the recent security incidents, the security record of Certification Authorities is considerably better than that of software providers which in turn are considerably better than users. While the Internet trust infrastructure is in need of urgent review, this should be focused on the problem of solving the many problems that are not addressed by the current infrastructure rather than merely attempting to replace what already works.

The Omnibroker protocol is an Internet service that serves as a replacement for and an enhancement of traditional DNS resolution services. Combining name and PKI resolution services in one protocol permits greater protection for the end user than the traditional approach.

Bibliography

1. TBS* DNS standard Reference.

2. **Cooper, D. et. al.** *Internet X.509 Public Key Infrastructure Certificate.* s.l. : IETF. RFC 5280.

3. **CCITT.** Recommendation X.509, The Directory-Authentication Framework, Blue Book. Melborne : International Telecommunications Union, Geneva Switzerland, 1988, Vol. Fasicle VIII.8: Data Communications Networks: Directory, pp. p127-144.

4. **Rescorla, T. Dierks and E.** *The Transport Layer Security (TLS) Protocol.* s.l. : IETF. RFC 5246.

5. **Kent, S.** *Privacy Enhancement for Internet Electronic Mail: Part II: Certificate-Based Key Management.* s.l. : IETF, February 1993. RFC 1422.

6. **Authority, Federal Bridge Certification.** *X.509 Certificate Policy For the Federal Bridge Certification Authority (FBCA).* December 9, 2011.

7. **al., M. Myers et.** *X.509 Internet Public Key Infrastructure.* s.l. : IETF, June 1999. RFC 2560.

8. **Edited by Ronald J. Deibert, John G. Palfrey, Rafal Rohozinski and Jonathan Zittrain.** *Access Denied: The Practice and Policy of Global Internet Filtering.* Cambridge MA. : MIT Press, 2008.

9.

10.

11.

12.

13. **(WEBSEC), IETF Web Security Working Group.** Charter. [Online] http://tools.ietf.org/wg/websec/charters.

14. **Ritter, Tom.** New Standards for Browser-Based Trust. [Online] http://ritter.vg/p/2012-TLS-Survey.pdf.

15. **J. Hodges, C. Jackson, A. Barth.** HTTP Strict Transport Security (HSTS). [Online] http://tools.ietf.org/html/draft-ietf-websec-strict-transport-sec-06. draft-ietf-websec-strict-transport-sec-06.

16. **C. Evans, C. Palmer.** Public Key Pinning Extension for HTTP. [Online] http://tools.ietf.org/html/draft-ietf-websec-key-pinning-01. draft-ietf-websec-key-pinning-01.

17. **P. Hallam-Baker, R. Stradling.** DNS Certification Authority Authorization (CAA) Resource Record. [Online] March 7, 2012. http://tools.ietf.org/html/draft-ietf-pkix-caa-05. draft-ietf-pkix-caa-05.

18. **Group, DANE Working.** Charter. [Online] http://tools.ietf.org/wg/dane/.

19. **Perrig, Dan Wendlandt David G. Andersen Adrian.** Perspectives: Improving SSH-style Host Authentication with. [Online] http://perspectivessecurity.files.wordpress.com/2011/07/perspectives_usenix08.pdf.

20. **Labs, Thoughtcrime.** Convergence Beta. [Online] http://convergence.io/.

21. **Eckersley, Peter.** The Sovereign Keys Project. [Online] https://www.eff.org/sovereign-keys.

22. **B. Laurie, A. Langley, E. Kasper.** Certificate Transparency. [Online] http://www.links.org/files/sunlight.html.

23. [TBS] ** Link to Diginotar break report.