WP8 D8.2 PKI Challenge (pkiC) Best Practice for PKI Users

Authors: Chris Gilbert and Kate Hodgson

A two-year project

fully-funded by the European Commission and
the Swiss Government

run by EEMA as part of a 13-strong management consortium
to solve the current interoperability problems associated with the deployment of Public Key Infrastructure (PKI) technology

www.eema.org/pki-challenge

The pkiC Management Consortium: -

Baltimore + Belgacom + Consignia + EEMA + GlobalSign + KPMG + Makra
Security & Standards + SmartTrust + University of Leuven + University of Salford + Utimaco + WISeKey

An EEMA Project

The European Forum for Electronic Business

EEMA

Funded by

European Commission

Bundesamt für Bildung und Wissenschaft
Office fédéral de l'éducation et de la science
Ufficio federale dell'educazione e della scienza
Uffizi federal da scolaziun e scienza
Federal Office for Education and Science

Swiss Government

Contents

1. Introduction

This paper is one of papers that have emerged from the analysis of the results of the EU and Swiss Government sponsored pkiC project. The purpose of the papers is to offer advice and guidance that will foster the deployment of interoperable PKIs and thus encourage the widespread adoption of PKI-supported trust. The other two papers make recommendations to PKI vendors and lists the challenges that the PKI industry (standards bodies, European Commission, users groups with an interest in this area, PKI Forum and other participants) must overcome. This paper is aimed at the purchasers and users of PKI products.

All PKI products are highly configurable. This paper aims to provide guidance to those organisations that wish to benefit from the services they can deliver but who also wish to deploy and use PKI in a manner that maximizes the chances of interoperability with other PKIs. It gives advice on avoiding configurations that, if applied at the initialization stage, could lead to interoperability problems with other CA operators at a later date. The paper also describes techniques for mitigating the known weaknesses in currently available PKI products.

The paper does not give advice on specific products. The issues covered here are general and widespread. Not all of the PKI products that you can buy will exhibit all of the problems that the advice offered in this paper is designed to mitigate.

This paper does not explicitly address the requirements of those end users who are buying in PKI services from a third party. The issues raised here are nonetheless worth considering and discussing with your provider. There may be some flexibility in your Service Level Agreement that will permit a degree of local configuration in line with the recommendations listed here.

It should be noted also that the PKI Challenge was essentially a technical exercise, concerned exclusively with the interoperability of software. PKI deployment, however, brings with it supplementary human and legal processes which may also have interoperability problems. The project and this guide do not address these issues.

2. Background

If, as a PKI user, your organisation does not plan to communicate with other organisations that are also PKI users then it would be easy to assume that the interoperability issues discussed in this paper do not apply to you. It is worthwhile, however, at least considering the points covered here. Many of them concern what amounts to no more than good PKI practice and may make PKI life easier in the long term, especially when upgrading or buying supplementary PKI-aware products at a later date. Making your PKI easy to work with may also open up otherwise unforeseen business opportunities and potential improvements in efficiency.

It is important to stress that the decision to support interoperability should be made before the start of the design and build of the PKI as it can significantly affect the way it is built. Also, while detailed technological understanding of how the Internet and PKI works is not a pre-requisite, a modicum of technical knowledge and a willingness to learn new concepts will definitely help.

3. Best Practice Issues

The remainder of this document describes topics and features of PKIs that are usually under the control of the purchaser of the PKI product and thus can be configured locally. In each case the rationale for the recommendations is also given.

3.1 Designing and deploying the supporting Directory

Most PKIs come with an optional, tightly coupled, X.500-based directory system to which PKI objects can be published and from which third parties can retrieve them as and when they are needed. Deployment of the directory is not usually mandatory but if you wish your user community to be able to look up the public keys of other users in the organisation and to be able to check the revocation status of certificates on-line and in real-time then a directory is essential.

A directory is an object oriented, hierarchical database. It is built using a predefined set of directory objects based on the X.500 model. The user, or a vendor agent program, creates the directory structure and the PKI software populates the directory objects with data.

Directories that accompany PKIs are often third-party products. As such they are general-purpose devices, not tailored to the specific needs of the PKI they come with. They support a far greater number of different directory structure objects than the PKI, and more importantly your organisation's implementation of the PKI, probably requires. The PKI vendor usually supplies some form of configuration tool that is executed against the empty directory to create within it the structures the PKI requires but there is usually at least some choice available to the user as to which attributes the directory will use to hold the information in their PKI. For the sake of interoperability it is safest to use only those attributes that are most widely supported, particularly for the fields in the Distinguished Name[1] in your certificates.

Recommendations

Use simple directory structures

We have recommended to the vendors that they should support at least the following Directory attributes. They are thus likely to be widely understood. Your PKI vendor may also mandate others:

C (country)
L (locality)
O (organisation)
OU (organisational unit)
CN (common name)
DC (domain component)

It is not usual practice to deploy non-PKI data in the directory if is to be made available for enquiry by third party organisations.

PKIs often also come with a tightly coupled relational database that is used as a repository for the information that supports the PKI components. Deployment of the supporting database is usually mandatory and should be a black box component of the PKI that is not available for configuration by the user. The directory is not the same as the database.

[1] The place in the directory where objects with that Distinguished Name can be found

3.2 Certificate Profile

The currently accepted structure for PKI digital certificates is described by the Internet Engineering Task Force (IETF) document RFC3280. This document has also been accepted by the International Telecommunications Union (ITU) as an update to its Recommendation X.509. X.509 is currently at version 3. RFC3280 is available on the IETF website[2] and while detailed knowledge of this document is in no way necessary to the successful deployment of a PKI it is certainly worthwhile getting familiar with as it gives a tremendous insight into how PKI works and thus how the certificates issued by your PKI are likely to be treated by other people's PKIs.

RFC3280 permits a large number of valid configurations of the certificates that a CA can publish. How you structure your certificates, which fields you choose to include and how you implement them can have an enormous impact on how the rest of the PKI-enabled world receives them and works with them. Understanding how the structure of your certificates can prevent other organisations from working with them will help you create software-friendly, interoperable certificates that will be well received by relying parties and encourage other organisations to do e-Business with you.

On reading RFC3280 you may well decide that many of the fields available to you are irrelevant. Their exclusion will **not** usually be a handicap in interoperability.

In general it is good practice to use as few fields as possible in your certificates as every field you add increases the risk that someone outside your organisation will have problems working with its contents. The remainder of this section describes the important structural components of certificates as well as some essential concepts about how the fields in the certificate are used.

Recommendation

Do not make your certificates any more complex than is absolutely necessary.

There are two field types in a certificate. There are a small number of mandatory fields that occur in the top of the certificate. These are followed by a number of extra, optional fields called Extensions.

3.2.1 Mandatory Certificate fields

The mandatory certificate fields are as follows. They cannot usually be removed from the certificate definition that your PKI uses and a certificate that contains only these fields is valid:

Version
The version of X.509 with which this certificate complies. You will probably not be able to change this.

Serial Number
An Integer, unique within the issuing CA and which identifies the certificate. The Serial Number is generated automatically by the CA when the certificate is created.

Signature
Tells the program using the certificate which Hash and Signature algorithms to apply to the user data that accompanies the certificate.

[2] http://www.ietf.org

Issuer
The Distinguished Name for the CA that signed this certificate. Other information about the Issuing CA may also be available at this location.

Validity from
The date and time from which the certificate is valid. This permits the pre-issuance of certificates before they are required.

Validity to
The date and time that this certificate ceases to be valid.

Subject
The Distinguished Name of the certificate holder's entry in the directory. Other information about the certificate holder, the subject, may also be available at this location.

Subject Public Key Information
The certificate holder's public key and the name of the signature algorithm it was generated for.

3.2.2. Certificate Extensions

There are a number of PKI features that the set of Mandatory fields does not enable. These features require the inclusion of optional, supplementary fields called Extensions before they can be used. The following table, Commonly Used Certificate Extensions', contains a list of commonly used Extensions. This list is not exhaustive and does not cover all of the Extensions available to the PKI implementer. It covers the Extensions that may be valuable to your correspondents if implemented correctly. We recommend that you implement all of these Extensions.

The **Impact** column in the table should be interpreted as follows:

Low	Exclusion of this extension is unlikely to cause any significant problems in the present PKI environment but as PKI becomes more widespread it may become standard practice to include it.
Medium	This extension is used by some PKI programs if it is present. Its presence definitely enhances the interoperability of your certificates while its absence may reduce the value that a relying party will place upon your certificates under certain circumstances.
High	This extension is becoming very important to PKI programs in supporting the trust that they should place in your certificates. Excluding it would be a serious mistake. You may find that your PKI software already includes it by default anyway.

Commonly Used Certificate Extensions

Extension Name	What is this Extension used for?	How should I Deploy it?	What happens if I Don't include it?	Impact
Key Usage (keyUsage)	Identifies what cryptographic operations the Private Key can be used for.	Specify with values of only Digital Signing or Encryption unless you have a specific business need for others and mark it as **CRITICAL.**	Crypto-operations might be performed using the incorrect key pair members. Signatures will not verify or encryptions decrypt.	**High**
Extended Key Usage (ExtKeyUsage)	Also identifies what cryptographic operations the Private Key can be used for. Used by Microsoft.	Use in the same way as the Key Usage field But do **not** mark as critical.	A Microsoft crypto-enabled application will not be able to work with your certificates.	**Medium**
Certificate Policies (certifcatePolicies)	Links the certificate to a real-world policy.	Register an OID[3] for your policy and place the OID in this field.	Not much at present. Most UAs[4] would not know what to do with it but it might future-proof your certificates.	**Low**
CRL Distribution Point (CRLDistributionPoints, CDP)	Describes a location on the Internet from where the CRL for your CA can be retrieved.	Most PKIs will use by default LDAP to access your directory. We recommend that you create certificates with an additional CDP that points to an HTTP location somewhere on your company portal. Your CRL should be copied to this location.	Relying parties that use CRLs to validate certificates will not be able validate yours and may not want to work with you.	**Medium /High**
Subject Key Identifier (SubjectKeyIdentifier, SKI)	See next row	See next row	See next row	**Medium**
Authority Key Identifier (authorityKeyIdentifier, AKI)	SKI and AKI are used by some UAs to link the certificate to a trusted root.	Ensure that your PKI creates 160-bit SKI and AKI values.	UAs may have problems linking your certificates to a trusted root.	**Medium**
Authority Information Access (authorityInfoAccess, AIA)	Many things. Most commonly for OCSP (See 3.4.2).	As per RFC3280 and RFC 2560.	Not much but giving a relying party as much information as possible will increase the trust they place in you.	**Low**
Subject Alternative Name (subjectAltName)	Defines an alternative name for the subject.	Insert a valid email address for your end user.	Microsoft crypto-enabled applications will not be able to associate your end users identity with the certificate and will thus not be able to use it. You will not be able to create S/MIME email.	**Medium**

[3] http://www.alvestrand.no/objectid/index.html

[4] User Agent. A standard term for an End User program. Usually runs locally on the client.

3.3 Other certificate profile related issues

3.3.1 Using Suitable crypto algorithms

Asymmetric cryptography, the basis for PKI, has been around for about 20 years now and has been in a state of continuous improvement. There is broad agreement that some of the earlier cryptographic algorithms have been superseded by better ones and that using the older algorithms, which are now capable of compromise, may be risky. It is always good practice to keep up with the market with regard to algorithms. There is a short-term risk, however, in immediate adoption of every new algorithm that comes along. It takes time for a new algorithm to propagate through the market. In using a new algorithm there is a chance that it may not yet be supported in the relying party's client software.

The best practice is always to use the most widely supported, recent and hardest to compromise algorithms.

The IETF document RFC3279 lists a number of suitable algorithms. It is important to note that support for any or all of these algorithms by PKI vendors is not mandatory but their presence in RFC3279 implies that there should be wide support for them within the PKI community;

Recommendation

Use the following algorithms:

- Hash functions – SHA-1
- Signature algorithms- RSA, DSA

3.3.2 Key Lengths

Support continues in the PKI industry for a wide range of key lengths. This is due in part to a rapid increase in the processing power available at the desktop. As computing power has increased it has been possible to increase security by using longer key lengths but there has been a lag in their universal adoption as the new technologies penetrate the marketplace. It is also due in part to the way the US government for many years controlled the cryptographic capability of exported software. Since the controls were relaxed in 1999 there has been a steady migration to longer key lengths.

In general it is best practice to use the largest key lengths available. This will deliver both the highest security and the longest life for your PKI objects. Your PKI will associate the algorithms available in its cryptographic engine with key lengths (e.g. RSA-1024 is the RSA signature algorithm applied to a 1024-bit key) and may give more than one key length option for each of the algorithms supported.

Longer keys require more processing time, however, and the lower the power of the machine hosting the User Agent (UA) then the longer it will take to perform cryptographic operations. Large key lengths on low power machines may take a considerable time to process. You should always take into consideration the age of the equipment that your user base has been supplied with and is expected to use. There is also always a risk that somebody somewhere will not be able to handle large keys but given the general high-awareness these days of Internet security issues it should be presumed that support for the largest key lengths available is widespread.

Recommendation

Use the longest key lengths that are associated in your PKI with the previously mentioned recommended algorithms.

3.3.3 Private Extensions

X.509 is **so** flexible that it allows the PKI owner to define his or her own extensions. Such 'private' extensions might be used to carry organisation specific information in the certificate. As long as private extensions are encoded correctly relying software will be able to understand the structure of the certificate that carries them.

3.3.4 Criticality

When including an extension in your certificate profile it is possible to mark it as Critical. This may seriously affect the interoperability that your PKI will enjoy with third-party organisations.

If a relying party receives a certificate that includes an extension that is marked as critical then the client software that they are using **must** be able to process that extension otherwise it must reject the certificate. It may be that your organisation uses one of the standard fields as part of an internal application and as such the criticality is required to enforce some form of local policy. Some PKIs may not be able to understand and process all of the extensions described by RFC3280 because their own architecture does not use some of them. Marking an extension as critical in your own PKI and then using your certificates with external PKI communities runs the risk that at some point in its life your certificate will be rejected by a relying party.

In general, extensions should only be marked as critical under exceptional circumstances.

Recommendation

Avoid using critical extensions if possible.

3.3.5 Private extensions and Criticality

A third party PKI will not understand a private extension. Marking a private extension as critical thus guarantees that nobody else in the world outside your PKI community will be able to work with your certificates. They will always be rejected.

Recommendation

Never use critical private extensions.

3.3.6 Basic Constraints (basicConstraints)

Basic Constraints is actually an Extension and as such could be found in any certificate. The reason why it merits an entry in this section rather than in the earlier table is that there is some disagreement between PKI vendors about how it should be included in certificates. This has arisen because there is a conflict between the various standards that describe how Basic Constraints should be implemented. This means that a valid certificate from one vendor's PKI could be rejected by a relying party that is using PKI software from a different vendor.

Current industry best practice, set to prevail in the next generation of standards, is as follows: If your PKI encodes Basic Constraints in an End Entity certificate and that encoding includes the CA field then that field should carry no value.

If your PKI does not work in this fashion then you should approach the vendor with a view to getting the feature fixed to bring it in line with the agreed best practice

Recommendation

Check with your PKI vendor that if your PKI encodes Basic Constraints in End Entity certificates that it has a null value in the CA field. If it does not then pressure them to fix it so that it does.

CA certificates should always contain the basicConstraints field with CA=TRUE.

Mark basicConstraints as **CRITICAL** in all cases.

3.4 Certificate Status Checking

It is insufficient for a relying party to be able to confirm the integrity of the contents of a signed or encrypted object using the information carried in the accompanying public certificate. The relying party should confirm the status of the certificate itself. In addition to checking Valid From and Valid To dates in the certificate a relying party should also recover, if possible, the Certificate Revocation List (CRL) from the CA that signed the certificate and look to see whether the serial number of the certificate is present in it. This will tell them whether or not the CA has removed its sponsorship of the key pair the certificate represents.

As implied by the above, the main vehicle for supporting certificate validity checking is the CRL but there is more than one way in which a CRL might be deployed.

3.4.1 Using CRLs directly

The table ‘Commonly Used Certificate Extensions’ stated that the CRL Distribution Point (CDP) extension should be added to the certificate to support validity checking. The CDP is a pointer to a location on the Internet from where the CRL can be retrieved. For the sake of interoperability it makes sense that this location should be a publicly accessible point of your network; if not in the directory then perhaps on the company website server. If posted on a company website then the CRL can be retrieved using HTTP rather than LDAP.

You should also ensure that your PKI issues CRLs that conform to X.509 v2, the most recently agreed structure for CRLs, and that they are Consolidated. There are two agreed formats for CRLs: Consolidated and Partitioned. Support for Consolidated CRLs is far more widespread than Partitioned CRLs and should be preferred.

Recommendation

Deploy Consolidated CRLs to a location accessible using HTTP. Add the CDP to your certificate profile.

3.4.2 Online Certificate Status Protocol (OCSP)

OCSP serves revocation requests by holding CRLs from one or more CAs at a single point: the OCSP server. The server responds to requests from OCSP-enabled client software to confirm the revocation status of individual certificates. An OCSP server can provide a revocation status service for many different CAs but many PKIs come with their own OCSP service which can be dedicated to a single organisation, satisfying requests for revocation both from inside and outside the organisation.

There are three possible configurations for OCSP:

1) Responses can be signed by the same CA that issued both the certificate being queried and the CRL list that might contain it.
2) Responses can be signed by a Validation Authority (VA) that has been delegated the authority to sign responses on behalf of 1).
3) A internal arrangement based on the trust structure of the local PKI.

Of these, Option 2) is preferred.

Option 3) is definitely **not** preferred; local strategies do not scale to sustainable internet-wide proportions. Option 1) has not actually been implemented by some PKI vendors. Option 2) is widely supported and introduces the least complexity into the trust model.

This issue has also been covered by WP8 D8.3 PKI Best Practice Paper - Guide for Vendors.

The location on the internet of your OCSP server is contained in the Authority Information Access (AIA) field of the certificate.

Recommendation

If using OCSP make sure that the value of AIA in your certificates complies with RFC 3280 and that the OCSP signing certificate used by the OCSP responder is issued by the same CA that issued the CRLs that the responder is servicing.

CDPs and OCSP are not mutually exclusive. It is not possible to know what method relying party software might use to recover revocation information from your PKI so if possible it should support both.

Recommendation

If possible your PKI should support both CDPs and OCSP.

3.5 Infrastructure issues

Deploying a PKI for use across an open LAN/WAN within the organisation presents few problems. Supporting interoperability with external relying parties, however, introduces an extra level of complexity. Making your PKI available to external parties conflicts with the requirements to protect the remainder of the organisational infrastructure from hacking and other meddling.

Most, if not all, organisations use a firewall to control the risk and this needs to be configured to work with the PKI. The configuration of the firewall to support the PKI requires careful consideration.

3.5.1 Firewall Configuration

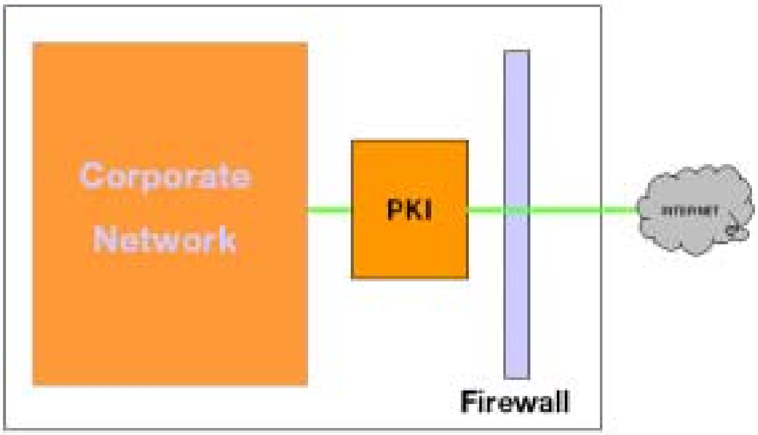
The components of the PKI that need to be accessible from both within and outside the organisation (CA, Directory, OCSP server and CDP) should be placed behind firewalls which only have those ports open that support the services available on the machines behind the firewall. If a protected network area only contains machines that support the PKI then the firewalls should only have open the ports that PKI exploits. In general, firewalls should never expose ports that the services they protect do not require.

Recommendation

Ensure that the following port numbers are available in your firewall to support your PKI. If the PKI is the only service in the network area protected by the firewall then close all other ports;

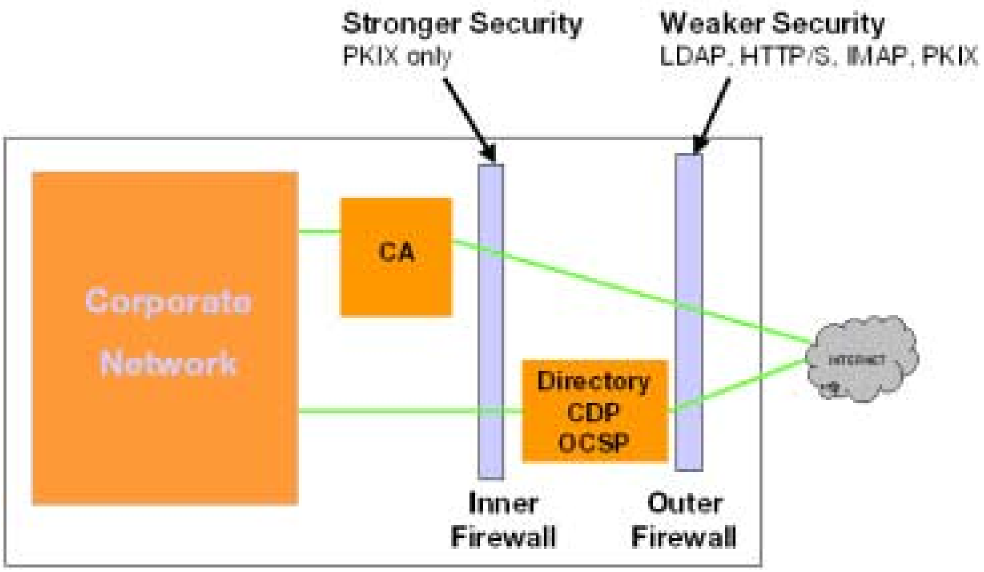
Port Number	Service	Use
(Definitely)		
829	PKIX	CA/RA communication
389	LDAP	Enquiring on-line LDAP directories
80	HTTP	For recovering CRLs from a public location
443	HTTPS	SSL enabled version of the above
(Desirable)		
636	LDAP/S	SSL-enabled version of the LDAP
143	IMAP	Mail transport for CMC communication
220	IMAP-3	Mail transport for CMC communication
585	IMAP/S	SSL enabled version of IMAP

Beware also that there may be some vendor-specific Ports used to support services within the PKI. Consult the product documentation and open any other ports specified. Do not open the ports for system administration programs in the outer firewall.



Example of PKI and Corporate Network protected by a single firewall

Consider using a layered firewall architecture that exposes machines that support the more common protocols like HTTP in a less restrictive uppermost layer while less common protocols like PKIX are serviced by machines that sit behind another firewall layer that exposes only those protocols.



Example of PKI and Corporate Network protected by a layered firewall

3.5.2 ISP Packet Filtering

The data stream beyond the organisation may be subject to monitoring and control. ISPs often monitor and filter out data packets for protocols commonly used by hackers to detect and exploit weaknesses in web services. It is important to talk to your ISP and make sure that the TCP/IP packets that support the protocols described in the previous section are not filtered. This applies to both the organisation owning and operating a PKI and Relying Party organisations.

Recommendation

Ask your ISP to not filter out the data packets that are used by the services described in the previous section.

3.6 Client issues

PKI-enabled clients often come with built-in presumptions about the structure and operations of the PKI from which they obtain their information. This is because PKI vendors often publish complimentary client software as part of their PKI offering. Tight integration with their architecture is inevitable and this may conflict with other PKI vendors' architectures leading to interoperability problems between heterogeneous PKIs and clients. Following the guidelines in this document will avoid many client-end problems. Certificate profiling has already been covered and is touched on again here along with other client-specific issues.

The operations that a crypto-enabled (e.g. S/MIME capable email) client program would be expected to perform can be grouped into three categories:

- Cryptographic operations
- Identity management
- Support for the Trust framework

Each category will be covered in turn.

3.6.1 Cryptographic operations

This is the application of the algorithms specified in the end user's certificate to the data that the certificate accompanies. The algorithms create hashes of the data, digital signatures and encrypt the data.

For the sake of interoperability it is best to follow the guidelines detailed earlier in this paper with regard to certificate profiling. i.e.

- Make the certificate structure no more complex than absolutely necessary
- Use the most widely supported algorithms
- Use the most widely supported and secure key lengths

3.6.2 Identity management

Support for the association of a person's real world identity with their electronic identity is a central feature of PKI. This only works seamlessly, however, within individual vendor's product sets. Interoperability, therefore, means coming to grips with the mechanisms for importing and exporting certificates into and from the client software and using them to associate certificates with correspondents.

Address book management
Most client applications come with some form of address book for managing correspondent's details. If the application is crypto-enabled and the correspondent's digital certificates can be associated with their address book entry then this should be encouraged in your end user community. Correctly integrating third-party certificates with the local address book increases the chances that the certificates will be used in a seamless fashion by your PKI software. Mail servers like Microsoft Exchange should have the features enabled that support Internet certificates.

Recommendation

Use the features available in the client application address book to associate correspondent's identities with their Digital Certificates. Enable PKI object handling in your mail servers.

Export file format
Within vendor product sets there is still a lot of support for proprietary data formats and communication protocols. Nearly all products, however, support the internationally recommended data formats for PKI objects. Within a product, however, there is usually support for a variety of formats and this can lead to some confusion.

Recommendation

If creating a certificate file for transfer to a third party then create the file in one of PKCS#7 or DER (binary) format. DER files may also be given a .CER or .CRT extension in some systems.

If creating a certificate and private key file for transfer to a third party then create the file in PKCS#12 format. The file may be given a .PFX extension in some systems.

3.6.3 Support for the Trust framework

There is a significant design assumption in PKI that it should be possible to retrieve from a directory the certificates published by a CA. This may be either to allow two users to contact each other securely by email or to support the construction of a trust path to prove the validity or otherwise of a certificate being used by relying software. Such Directory querying works well within a single PKI vendor strategy. It is compromised, however, in a heterogeneous environment by the failure by PKI vendors to support third-party certificate look up and by the use of proprietary directory structures.

This section describes techniques that may be used to mitigate this weakness and to reduce the risk of failure by a relying party to recover a certificate which might prevent them from creating the required trust relationship.

Trust path construction

Most client applications will not let you use a certificate unless it is valid within a known trust context i.e. it is within date, it has not been revoked and it is sponsored by a known trusted root.

Checking certificate expiry dates usually comes as a built-in feature. If the default configuration is ‘No checking’ then the feature should be enabled.

Recommendation

If certificate expiry checking is an optional feature in your software then enable it.

Revocation checking rarely comes enabled and this should be switched on. Consult the application user guide. Be aware, however, that recovery of a third-party CRL may not be possible, as it may not have been made publicly available by the Issuer. Switching on CRL checking in your client applications does not, therefore, guarantee revocation checking. You should check that you are happy with the behaviour of your client applications with CRL checking enabled before their deployment to your user base.

Recommendation

If revocation checking is an optional feature in your software then enable it.

Trusted Roots

When users in the same PKI community communicate securely with each other they do so within a safe trust context as they share a common Trusted Root. The certificates that each party use will have been issued by the same CA and the Root CA certificate for the community will usually be installed in the Trusted Root CA Store on both clients. Difficulties arise when working with certificates coming from other PKI communities that have been signed by a CA outside the local trust context. The Root CA Certificates from these third-party CAs will not be present in the local Trusted Root CA Store so trust for certificates from members of those communities by local users will not be automatic.

Management of the local Trusted Root CA Store is a key activity in making trust work in this case. It is essential that your users have access to all of the root CA certificates that your organisation wishes them to. There must also be a strategy in place to grow the Trusted Root CA Store in a controlled manner to extend the trust used by your organisation when it is required.

Some organisations grant their end users local administrator rights to their clients. When new CA certificates are encountered the end user can add them to the client. More security sensitive organisations that do not accept the legal implications of this do not permit their end users to add new CA certificates. In the absence of local control there must be a centralized update mechanism whereby the client is regularly updated with new Trusted Root CA certificates.

It is also important to note that the Microsoft desktop comes prepackaged with a large number of Trusted Root CA certificates and thus, by default, trusts certificates within those communities. You should consider whether you wish to inherit these trusts automatically. If not then you should cull the unwanted root CA certificates from the desktop prior to deploying it to your user base.

Recommendation

Deploy your desktop to your users with the Root certificates of the CAs that you wish to trust already installed;

Either
Give your users Administrator rights to the Trusted Root CA store so that they can add in new CA certificates when they encounter them.
Or
Have a support strategy that updates the deployed Root CA store regularly to include new CA certificates.

.

Trust path inclusion
To support the propagation of your trust infrastructure beyond the boundaries of your organisation it is good practice to include the trust chain of public certificates with the outgoing emails. This will allow your correspondents to import your trust context and improve their chances of trusting signed objects received by them from users in your PKI community.

Recommendation

Configure your email clients to always include the whole trust path in outgoing emails.