EKMI Policy Guidelines (DRAFT 1.0)

Introduction

In order for an enterprise to build and operate an Enterprise Key Management Infrastructure (EKMI) to serve its business and security needs, it is necessary for implementers and operators of the EKMI to be guided by the enterprise's policies governing the EKMI.

However, EKMI is a nascent field and has the potential to create some levels of confusion amongst policy-makers in the short-term. This document attempts to address that problem by creating guidelines that incorporate best-practices as determined by a group of key-management practitioners across the world. Policy-makers are encouraged to use these guidelines as a starting point towards creating policies that address their specific needs.

Since an EKMI – as agreed upon by the OASIS EKMI TC - is composed of a Public Key Infrastructure (PKI) and a Symmetric Key Management System (SKMS), the policy guidelines must cover both symmetric and asymmetric cryptographic keys used by the enterprise. Consequently, this document will address both PKI and SKMS policies.

Assumptions

Some assumption being made by this TC are as follows:

1. Threat levels on the internet will continue to rise. Consequently, companies want to implement solutions that address current (1-2 years) and medium-term (3-5 years) vulnerabilities at a minimum.

2. Controls deemed effective today will lose their effectiveness as attackers work around them. The "ineffective" controls – such as firewalls, for example – will still be necessary, but attackers are going to factor that into their tactics and use only the open ports of a networked computer to attack the machine.

3. Readers are familiar with the field of cryptography. They do not need to be experts; but this document does not introduce readers to the subjects of cryptography. If unfamiliar with the terms and concepts, readers are advised to review some of the referenced texts at the end of this document before reading this document.

A note on convention: guidelines within the specific type of key-management infrastructure, either PKI or SKMS, will be sequentially numbered. So, PKI guidelines will be named as PKIG-1, PKIG-2, etc., while SKMS guidelines will be SKMSG-1, SKMSG-2, etc. Each guideline will be followed with an explanation so implementers may understand the rationale for the recommendation.

PKI Policy Guidelines

The field of PKI is mostly governed by Request For Comments (RFC) documents published by the Internet Engineering Task Force's (IETF) Public Key Infrastructure X.509 (PKIX) Working Group - http://www.ietf.org/html.charters/pkix-charter.html, so this document will not repeat standards already defined within the RFCs. However, when an RFC provides information only, it leaves room for differences in implementation to accommodate specific needs of implementors. In those situations, we provide additional guidance based on practices honed from years of experience by PKI specialists in the field. While this guidance may not be suitable for all situations or companies, the vast majority of implementers may find them sufficient.

The most important RFC related to PKI policies is RFC 3647 (http://www.ietf.org/rfc/rfc3647.txt). Readers are strongly encouraged to review this RFC and understand the implications of the RFC before establishing a PKI. This EKMI Policy Guidelines document supplements the information in that RFC with recommendations deemed useful for optimal PKI implementations.

PKIG-1

Use hardware-based cryptographic tokens, such as Hardware Security Modules (HSM), smartcards, Trusted Platform Module (TPM), etc., for the generation and storage of asymmetric cryptographic key-pairs.

Public-key cryptography is an extremely powerful security technology that can engender high levels of information assurance when properly implemented. However, the assurance of the integrity or confidentiality of data is ultimately dependent on the protection afforded to the private key of the asymmetric cryptographic key-pair(s). Most software products storing key-pairs (inside a container known as a keystore) generally use Password-Based-Encryption (PBE) to protect the private keys. While PBE can be effective when properly implemented, the strength of the cryptosystem relies on the strength of the password, and the difficulty presented to an attacker in acquiring the keystore and attacking it using dictionary-attack tools. Unfortunately, neither is very difficult to an attacker. Keystore files can generally be copied off from directories of users and transported to external attackers, while backup tapes of corporate servers can provide insiders to keystore files. Using commonly available dictionary-attack tools, an attacker has a greater than 0.9 probability of compromising the keystore within the first 30 minutes and gaining access to the private key(s).

An external, hardware-based cryptographic token eliminates both these vulnerabilities.

By having the private key stored on a specialized chip or device external to the hard-disk and/or computer, the attacker is presented with significant difficulties in trying to gain access to the contents of the chip. Special software is required to access the contents of the cryptographic chip, and each access request is typically authenticated. Even if the attacker managed to gain access to a token – such as a lost/stolen smartcard – most manufacturers of such chips lock the token after 3-5 incorrect password attempts. Unlocking the token requires an elaborate authentication and recovery procedure in most companies thus foiling the attacker's intentions.

PKIG-2

Use M of N custodians to control access to the asymmetric cryptographic key-pairs.

Given that PKIs are hierarchical in nature, access to the private key of a Certificate Authority (CA) has the ability to affect the entire hierarchy of certificates under that CA node. Consequently, access to private key(s) of a CA must be a tightly controlled to ensure there is no single point of vulnerability to the PKI.

Creating a set of N custodians and requiring a minimum subset of those N custodians, M, to be present to activate the private key(s) is strongly recommended. Each of the N custodians would only have one part of the full authentication required to activate the private key(s). A minimum of M of the N custodians must authenticate themselves to the PKI application to activate the private key(s).

Hardware security modules (HSM) used in PKIs generally support M of N access control; using this control ensures that the PKI cannot be compromised by any single individual.

PKIG-3

Use the Online Certificate Status Protocol (OCSP) to ensure "real-time" certificate status checking for online applications.

Since digital certificates can be used when disconnected from a network, the traditional method of determining the current status of a certificate has been with the Certificate Revocation List (CRL). CRLs are issued periodically to keep the overhead of issuing such lists, low. However, today's "always on"

environments require higher levels of assurance when dealing with digital certificate; a periodic update is unacceptable for many environments.

Using an OCSP Responder that provides an up-to-the-minute status of a digital certificate can reduce the risk of accepting a cryptographic message that was created with a key that belonged to a revoked certificate.

PKIG-4

Ensure business continuity by creating a Disaster Recovery plan for a PKI and by testing it frequently.

Business continuity, in general, requires companies to establish a recovery plan for their IT infrastructure. Due to the use of sophisticated cryptographic equipment and security procedures, PKI's add a layer of complexity not typically encountered in standard IT environments.

Having a PKI-specific recovery plan describing how each component of the PKI (Certificate Authority, Registration Authority, OCSP Responder, Publishing Directory, etc.) is to be recovered – securely – is highly recommended. In addition, testing the plan until there are no flaws in the execution and/or deviations from the plan is critical to the recovery of the PKI.

PKIG-5

Key Custodians must understand and acknowledge their responsibility.

While PKIs are very technical environments, like all IT infrastructure, they exist to serve the business. Yet many business people do not understand the nuances of cryptography and the implications of their responsibility within an EKMI.

Individuals who are responsible for the management of cryptographic keys within a PKI – generally known as Key Custodians – must fully understand the implications of their responsibility and acknowledge it. Without such acknowledgment, there can be no accountability.

SKMS Policy Guidelines

While symmetric encryption has existed for decades, the management of symmetric encryption keys across an enterprise – with some exception - has never been a focused discipline. OASIS' EKMI TC has chosen to establish standards and guidelines for an enterprise-scale Symmetric Key Management System (SKMS). While these guidelines may not be suitable for all situations or companies, the vast majority of implementers may find them sufficient.

SKMSG-1

Use strong, industry-standard symmetric key ciphers.

The cryptography community has many choices of ciphers available to it for encryption purposes. However, encryption is a complex arena and remains safest when companies use ciphers and algorithms that have been reviewed by many experts and over which there is consensus on their adequacy and safety for a specific purpose.

Consequently, implementers of an SKMS should either use some of the strongest symmetric key ciphers available to them, or at the minimum implement an infrastructure that allows them to replace ciphers easily without having to change their applications or infrastructure (although a library update and configuration changes will probably be required).

SKMSG-2

Use hardware-based cryptographic tokens, such as Hardware Security Modules (HSM) for generation and use of symmetric cryptographic keys on servers, and use smartcards, Trusted Platform Module (TPM), etc., for authenticating resources requiring access to symmetric keys on clients.

Just as asymmetric keys are protected by the use of external hardware security modules, so are symmetric keys. However, different types of devices are used based on the classification of the computer.

Servers, which may require many symmetric keys – perhaps even in the thousands – will be better served with specialized devices that not only accelerate the generation and use of symmetric keys, but also protect them when in use by "unwrapping" them inside the security module so that tools monitoring the random access memory of computers will not discern the unwrapped encryption key. The same hardware security module can also be used to manage asymmetric keys used by the server when signing and/or encrypting data.

Client machines and devices are better served by individual tokens such as smartcards or Trusted Platform Modules (TPM) resident on some newer computers. The tokens are used to generate and store the asymmetric key-pair, which in turn are used to authenticate the application requesting symmetric key services from an SKMS.

SKMSG-3

Ensure high levels of security in all components and layers of the SKMS.

This might seem to be a redundant guideline, since cryptography and key-management is all about security. However, it is our experience that without information and guidelines, companies tend to implement what is expedient – not necessarily what is secure, and auditors lack the knowledge and guidelines to question such implementations.

In this guideline, we would like to stress that SKMS implementations avoid the following practices:

- Using proprietary algorithms;
- Storing plaintext (unencrypted) encryption keys with data;
- "Hiding" plaintext encryption keys on the machine;
- Compiling encryption keys within the software performing the cryptography;
- Encrypting one symmetric key with another;
- Using a single encryption key across the enterprise;
- Using weak password for Password-Based-Encryption (PBE) algorithms;
- Implementing an SKMS without a key-rotation and/or key-compromise plan;

SKMSG-4

Ensure business continuity by creating a Disaster Recovery plan for the SKMS and by testing it frequently.

While a properly architected SKMS will have redundancy due to replicated servers, it is nevertheless prudent to have a business continuity plan that addresses the unique requirements of the SKMS. However many Symmetric Key Service (SKS) servers exist within an SKMS, it is always possible that all SKS servers may go down simultaneously for different reasons. In such a situation, only a well-thought out Disaster Recovery plan that has been thoroughly tested, will help a company recover from that catastrophe.

SKMSG-5

Ensure all resources that access encryption keys within the SKMS are strongly authenticated.

Companies encrypt data because there is risk associated with that data falling into unauthorized hands. Since access to the decryption key is by definition, access to the plaintext data, it is necessary to ensure that all resources attempting to access a symmetric key are authenticated properly to determine their authorization to access the data.

Given that userid/passwords can be spoofed easily, depending on single-factor, secret-key based authentication to determine the resource's access to symmetric keys, is certain to result in a compromised system. The use of public-key cryptography with two-factor authentication tokens to authenticate requesters is highly recommended to mitigate authentication risks.

SKMSG-6

Use M of N custodians to control access to the tokens which authenticate resources to access symmetric encryption keys.

When a company has deployed an SKMS to manage encryption keys across their enterprise, access to a cryptographic token that requests symmetric keys from the SKS server gives the requester access to the plaintext data. If a single individual is given access to the token, by definition, this allows that individual to access any plaintext data encrypted under keys that are granted access to the credential protected by the token.

Creating a set of N custodians and requiring a minimum subset of those N custodians, M, to be present to activate the token is strongly recommended. Each of the N custodians would only have one part of the full authentication required to activate the token. A minimum of M of the N custodians must authenticate themselves to the application that will access the token.

SKMSG-7

Reduce risk by ensuring that only an optimal amount of data is encrypted per encryption key. The SKMS must support as many symmetric encryption keys as are needed by the company.

If a company were to encrypt all data with a single key, it simplifies some aspects of key-management (key-generation, key-archival, key-retrieval) but complicates others (key-rotation and key-sharing). Key-sharing is particularly risky, since the more people/resources that know the key, the greater the probability of it getting compromised.

Assuming that the company can never know all potential vulnerabilities in the IT infrastructure, it is counter-intuitive to use fewer encryption keys across the enterprise. Consequently, the SKMS should support the management of large numbers of keys so that the company can spread its risk across as many encryption keys as it deems suitable. Additionally, the SKMS must support strong access control and automated archival/retrieval processes so that the key-management function doesn't impose a burden on administrative resources.

SKMSG-8

Rotate encryption keys to mitigate risk of a compromise to the key or ciphertext. Allow for automated key-rotation in the SKMS.

The longer a key is used to encrypt data, the greater the probability of that key being compromised (due to faster brute-force attacks, leakage of some critical information that helps the attackers, insider knowledge, etc.).

Companies must periodically eliminate these risks by changing their encryption keys frequently as their risk-management strategy and constraints permit them. The SKMS must support automatic key-rotation so that manual intervention by operators is unnecessary.

SKMSG-9

Ensure an immutable audit trail exists for the SKMS.

Audit trails are an important mechanism in ensuring accountability for the actions of a company and its resources. In the event of a breach, audit trail data provide evidence to law-enforcement agencies that may allow them to convict the attackers.

Consequently, an SKMS must provide immutable records of all activity within its core infrastructure that show the chronology of events. The log must be periodically reviewed and validated to ensure that keys are being accessed only by properly authorized entities.

It must not be possible for attackers to circumvent the audit log or to modify the logs to "hide their tracks".

SKMSG-10

Key Custodians must understand and acknowledge their responsibility.

SKMS – like PKI - are very complex and technical environments; but like all IT infrastructure, they exist to serve the business. As with PKI, many business people do not understand the nuances of cryptography and the implications of their responsibility within an EKMI.

Individuals who are responsible for the management of cryptographic keys within a SKMS – generally known as Key Custodians – must fully understand the implications of their responsibility and acknowledge it that responsibility, so they may be accountable for their actions in protecting and managing company data.