Implementing Core IT Security Services

Steve Purser

Steve Purser is Director ICSD Cross-Border Security Design and Administration at Clearstream Services, Luxembourg. Steve is also a founder Member of the "Club de Sécurité des Systèmes Informatiques au Luxembourg (CLUSSIL)" and author of "A Practical Guide to Managing Information Security" (Artech House, 2004).

1. Introduction

This paper describes an approach for implementing core IT Security services within a modern, highly distributed, IT infrastructure. It is supposed that such an IT infrastructure will have the following characteristics:

- A high degree of complexity.

  Usually, the infrastructure will be comprised of several generations of IT systems including mainframe, client server and sometimes distributed object architectures.

- A lack of centralised administration facilities.

  Such facilities do exist, both for general IT administration tasks and specifically for IT Security administration.

- Little or no documentation of legacy systems.

  A common problem with legacy systems is the lack of accurate documentation reflecting the current state of the system. This is exacerbated by the fact that the personnel involved in putting these systems into place are often no longer with the enterprise.

- A poorly defined network perimeter.

  Usually, the Internet Gateway will be well protected using modern perimeter defences, such as Firewalls, but dial in and legacy connections may be insufficiently protected.

- Inappropriate procedures and working practices

  Many of the procedures and working practices associated with modern IT infrastructures have gradually evolved from the procedures that were put in place for the legacy systems. Such procedures rarely take account of the increased complexity of the IT environment and do not scale well.

In an environment of this complexity, it is necessary to implement security services using a structured approach. The approach described in this paper is based on the assumption that most applications potentially require a standard set of security services, known as 'core' security services, and that these services can be implemented in the form of an IT Security Architecture. This approach has the following advantages:

- Applications do not have to provide core security services themselves. They rely on the security services provided by the underlying architecture.

- As core IT Security services are implemented in the form of a Security Architecture, these services are standardised.
- New applications can be developed to take advantage of the existing architecture, therefore decreasing time to market.

2. IT security layers

In order to successfully implement an IT Security architecture, it is necessary to understand how software components interact from a security point of view.

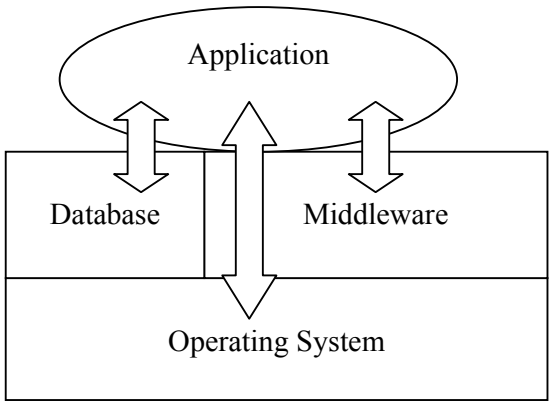
The most important layer of software is known as the Operating System (OS). This is the most important because all tasks performed by the system involve the participation of the operating system. For this reason, the security system associated with the OS usually determines the level of security of the entire system. An exception to this is provided by the use of cryptographic smart cards, where the cryptographic services placed on the card can, under certain conditions, increase the level of security offered by the operating system itself. It should be noted however, that applications interface with the smart card through the operating system.

Other support software, such as relational database software and 'middleware' (e.g. CORBA) is implemented on top of the OS and uses the services offered by the OS. As a consequence, the security of this type of software is dependent upon the security of the OS. In particular, if the OS is penetrated, it is usually trivial to bypass the security mechanisms associated with these packages. A concrete example is provided by relational database packages running on mid-range systems. If the superuser account associated with the OS is obtained, access to the database administrator account is usually trivial.

Application layer security is generally weaker than OS and support software security for several reasons:

- Application layer security relies on OS and support software security.
- Application layer security is often designed and implemented by engineers who are unaware of the relevant concepts (e.g. locally designed cryptographic algorithms for storing passwords).

These ideas are summarised in the following diagram:



Because the operating systems provide the most fundamental security services to the IT infrastructure, the IT Security architecture should be designed to reinforce and to complement the services provided by the operating systems themselves.

3. Modelling the IT infrastructure

The first step in the proposed approach is to model the existing IT infrastructure. The purpose of this exercise is to reduce the complexity of the problem by concentrating on the issues, which are important from an IT Security viewpoint.

One way to achieve this is to classify systems according to simple criteria, such as:

- Internal or perimeter system
- Type of operating system (mainframe, mid-range, PC, PDA)
- Type of application (sensitive or non-sensitive)
- Networking equipment

And to show how these systems are interconnected using a simplified network diagram. This results in a simplified graphical representation of the IT infrastructure illustrating the essential classes of systems and the way in which they are interconnected.

The risk analysis is carried out against this simplified architecture and will result in the addition of 'risk labels' to each component on the diagram.

4. Risk Analysis

The purpose of core IT Security services is to reduce risk related to the IT infrastructure. A risk analysis is performed against the model of the IT infrastructure in order to identify the key risks and how they are distributed throughout the infrastructure. The aim is to identify those risks, which can be reduced using an architectural approach, rather than those associated with a particular application.

Many tools and methodologies exist for performing risk analyses and defining plans to reduce the identified risks to an acceptable level. However, most of these formal methodologies are too broad in scope and too slow for the purposes of a first cut risk analysis. A simple, but effective approach, is to use risk analysis tables.

For each type of system identified by the model, two risk tables are produced. The first table lists the major risks in the absence of any protection mechanism and has the following columns:

- Identifier for this risk
- Description of the threat
- Probability of occurrence (low, medium or high)
- Impact
- Comments

The second table describes the measures to be put in place to reduce the risk, together with the estimated residual risk and has the following columns:

- Identifier for this risk
- Description of security service and mechanisms
- Residual risk (low, medium, high)
- Comments

The residual risk should be reported to higher management and signed off as part of the design process.

5. Identifying architectural components

For each component of the IT infrastructure (network segments are considered as components), the risk analysis process has identified the risks and the IT Security services to be put into place to reduce these risks to an acceptable level.

The next step consists in analysing the distribution of these services, to ensure that they are deployed in the most efficient manner. The output of this step is an updated model of the IT infrastructure, with the following information:

- Logical components providing well-defined security services.
- Flows of security related information between logical components and target systems.

In order to do this, it is necessary to have a reasonable knowledge of how such services are offered by commercial packages, as the final architecture will be essentially an integrated set of commercial packages. As an example, security scanners are available as both a 'network centric' product or as a 'host based' product. Network based tools are very convenient in that they centralise information from a number of distributed platforms, however security routers and Firewalls often prove to be important obstacles. Host based scanners are more decentralised and it is often necessary to centralise the resultant information, but this can be done with standard protocols (such as ftp), which is less of a problem for network barriers.

The following are guidelines for defining architectural components:

- Make maximum use of the native security services offered by the operating systems. This is the last line of defence and arguably the most important (see 'Security Layers').
- Centralise services wherever possible in order to facilitate administration. A good example is provided by access control, where a centralised database of access control information greatly facilitates the administration process.
- Do not include security services specific to particular applications (e.g. non-repudiation). There is little advantage in implementing these as architectural services.
- Prioritise services and discard those services offering protection against minor risks.
- Consider placing security servers on protected networks.

Examples of logical components, which may be used to construct an IT Security architecture, are given below. Most of these components are available as commercial off the shelf (COTS) products.

- Perimeter defence components.

  Typical perimeter defence components include Firewalls, authentication servers and associated encryption software. Such components need to be selected with care if they are to inter-operate in an optimal fashion. For example, integrating Virtual Private Network (VPN) software with Firewalls allows encryption and proxy services to work together. RADIUS or TACACS+ servers are often used for authentication of dial-in connections.

- Intrusion Detection systems.

Intrusion detection systems essentially come in two flavours – Network Intrusion Detection Systems (NIDS) and Host Intrusion Detection Systems (HIDS). The former tend to concentrate on analysing protocol information, whereas the latter analyse information produced by the host itself (such as audit trail information). NIDS and HIDS systems provide complementary views of the same events.

- Vulnerability scanners.

  These are used for detecting operating system and networking software level vulnerabilities. As with intrusion detection software, vulnerability scanners can be host-centric or network-centric. Network scanners probe vulnerabilities by detecting and probing open ports and are capable of providing a broad view of vulnerabilities across a network. Host-based scanners are more limited in scope, but are able to probe deeper into the local configuration. Accompanying procedures are necessary to ensure that identified vulnerabilities are corrected without affecting the stability of the applications.

  It is also worth mentioning the existence of telephone scanners in this context. Telephone scanning software is the commercial equivalent of war dialling software and is used to identify security weaknesses that are accessible through the telephone system.

- External security management systems for mainframe environments.

  These systems are the de facto solution to mainframe security. They provide basic security services, such as authentication and access control. Examples include RACF, TSS and ACF2.

- A framework for handling malicious code.

  Handling malicious code normally requires a multi-layer approach, with different tools at the network perimeter, the server infrastructure and the workstation. A malicious code protection framework can therefore be viewed as an architecture within an architecture. Examples of tools that can be used to construct such a framework include traditional anti-virus software, content scanners and NIDS systems.

- Privilege managers.

  Privilege managers are used to provide users with the ability to execute a restricted subset of commands using a privileged profile. These are often used to prevent sharing of the superuser account (or ‘root’ account on UNIX systems).

- Public Key Infrastructure (PKI).

  PKI is used to support the notion of ‘trust’ in the IT domain. The infrastructure is used to manage electronic certificates. This infrastructure enables and supports the use of advanced cryptographic techniques throughout the architecture and is a strong enabling technology.

- Secure storage devices.

  Secure storage devices are most frequently used to protect cryptographic secrets. At the server side, cryptographic keys tend to be stored in specialised devices known as Hardware Security Modules (HSM). At the client side, a number of solutions are currently available, including smart cards, dongles and specialised hardware tokens.

- Unique Administration Interface (UAI).

  The UAI should be used as the central point for all IT security administration tasks. Currently available products tend to concentrate on access control services, but this is a rapidly evolving product area.

- Secure Middleware
  In some areas, notably CORBA, secure middleware products are available. Highly distributed object environments present many special challenges from a security point of view. A secure ORB greatly simplifies the task of securing such environments.

This list is by no means exclusive and the components present in a given architecture will necessarily reflect the needs and infrastructure of the enterprise concerned.

6. Implementation

At the end of the design phase, a logical architecture has been specified, but no products have been identified. The implementation phase involves the following activities:

- Identification of priorities and production of a phased deployment plan.
- Selection of commercial packages.
- Implementation and integration of packages following the plan.
- Negotiation of Service Level Agreements (SLA) and support contracts.
- Possible in-house development to improve integration.
- Training of administrators and support personnel.
- Adapting current procedures and introducing new ones.

For a project of this complexity, a phased approach is recommended. Components are prioritised and introduced into the IT infrastructure based on the importance of the security service they are offering. Acquisition of commercial packages is to be favoured over in-house development as few organisations have the necessary expertise to develop and maintain security software. Although in-house development should be minimised, in some cases it cannot be avoided (e.g. using an API to allow control of application access rights from a UAI). Selection of commercial packages is carried out using standard methods such as RFPs and/or comparative studies. As security software is often considered to be critical (problems can often block the production environment), sufficient attention needs to be paid to support contracts and Service Level Agreements.

It is to be expected that the introduction of an IT Security Architecture will result in major changes to current procedures and practices. It is important to challenge concepts, which are related to older, simpler IT architectures. For example, distributed IT architectures can easily produce hundreds of log files. It is not possible to review the vast majority of these files proactively. A better solution is to increase the proactive security measures, introduce real-time alerts and refine escalation procedures. If this is achieved, it should be possible to restrict systematic log analysis to the most important log files.

Throughout the implementation process, it is important to talk to other organisations, which have experience in implementing the packages selected.

7. Using security baselines

Many of the components, which will be implemented as part of an IT Security Architecture, operate by allowing the administrator to define and manage a security baseline. In this context, a

security baseline is a technical version of the IT Security policy in a particular technical domain. The baseline specifies what the security configuration should look like. The associated component will either signal or manage any detected exceptions to the baseline.

Translating IT Security policy into a technical baseline is a relatively complex, but important activity and is usually carried out by experts in the underlying technology. For example, the configuration of the policy of a Unix security scanner will usually be done by a Unix security administrator.

Whereas some components will automatically manage exceptions to the security baseline (such as virus scanners, which will usually clean up detected viruses), many will simply alert the administrator to the presence of a problem. It is extremely important to define procedures for the correction of such problems and it is to be expected that such procedures will involve people outside the IT Security department. In the case of security scanners, it is often necessary to alert suppliers of commercial packages to possible security problems and to follow the resolution of these problems together with the supplier.

Typical problems in this area include over ambitious targets, inappropriate procedures and insufficient tracking mechanisms.

8. Putting It All Together

This paper has presented a simple, practical method for designing and implementing core IT Security services in the form of an IT Security Architecture. The reasoning behind this approach is that the architecture will supply standard services to the applications, thus removing the need to implement these services within the applications themselves. From an architectural standpoint, the overall complexity of the IT infrastructure will be reduced, which will make it easier to understand and manage security incidents.

The following comments apply to the whole process:

- A business case should be developed to support the initiative. The business case should explain what risks are being addressed and how they will be reduced, any cost savings due to increased efficiency should be signalled. It may also be possible to show that certain enabling technologies are a necessary precursor to business projects (e.g. E-Commerce) and that the architecture will reduce time to market for future applications.
- It is important to develop the practical side and the theoretical side of the project together. In particular, whilst it is important to follow a structured and well-documented approach, it is important to implement the first components quickly in order to demonstrate progress.
- Increasing complexity is a sign of problems. The architecture should be simple and easy to understand.
- Today's IT environments are characterised by rapid change. For this reason, a 'modular' approach is to be preferred. This will allow for components to be replaced on as as-required basis.
- It is extremely important to define realistic security baselines. The residual risk should be signalled to management and signed off.
- Procedures are as important as the technological solution and should be developed as components are introduced.