Security Challenges For Future Systems

Steve Purser
Head of Technical Competence Department
June 2011

Agenda

- Introduction to ENISA
- The Trends
- Scope & Requirements
- Design Considerations
- Deploying Secure Systems
- ENISA's contribution

I. Introduction to ENISA

Who are we?

- The European Network & Information Security Agency (ENISA) was formed in 2004.
- The Agency is a *Centre of Expertise* that supports the Commission and the EU Member States in the area of information security.
- We facilitate the exchange of information between EU institutions, the public sector and the private sector.

Activities

- The Agency's principal activities are as follows:
  - Advising and assisting the Commission and the Member States on information security.
  - Collecting and analysing data on security practices in Europe and emerging risks.
  - Promoting risk assessment and risk management methods.
  - Awareness-raising and co-operation between different actors in the information security field.

II. The Trends

Several generations of Architecture

- Computer architectures have changed enormously in the past 20 years:
  - Mainframe environments.
  - Simple networked environments.
  - Client-server and three tier architectures.
  - Highly distributed architectures.

- These architectures are secured according to different principles.
- Many companies run heterogeneous environments involving several generations of technology.
- Boundaries between technologies are weak points.

De-centralisation

- In the mainframe era, all processing essentially took place within a centralised host.
- In client-server and three-tier architectures we separate data access, application and presentation functionality.
- In highly distributed environments the data is encapsulated in objects...
  - ...which can be anywhere.
  - In some architectures, hiding the location of the data from the user is a design principle.

Globalisation

- The Internet has resulted in global connectivity.
- Most IT architectures are embedded in a global environment even if they were not designed that way.
- A fine line separates extranets from the Internet.
- Similarly, users regularly switch context between Internet and intranet sessions.
  - This is prone to store and forward attacks.
- In global environments:
  - Who do you turn to when things go wrong?
  - How do you know who you're dealing with?

Authentication ≠ Trust

Empowerment of the End User

- As systems become increasingly sophisticated, they are offering more choice to the end user.
  - The concept of Power Users illustrates this trend.
  - Where security is concerned, the move towards browser-based applications is important.
- It is clear that many users are not rising to this challenge at present:
  - Botnets are built largely from compromised PCs.
  - Existing security controls are not being used effectively.

The Need For Speed

- The Internet year has now become reality.
- In order to remain competitive, organisations strive to be first to the market with products and services.
- Where software development is concerned, this has resulted in shorter development lifecycles.
- As a result, we can expect to see more products being released in a less mature state.

III.
Scope and Requirements

Getting The Scope Right

- An operational system is not just technology.
- System = Software + People + Procedures.
- The environment in which a system operates will have a major influence on the security of the overall system.
- Secure software will not function correctly if we make unrealistic assumptions about how people behave.
- This makes developing systems for mass markets more difficult than developing turnkey systems.

The Key Challenge

The key challenge to developing secure systems is understanding and responding to the limitations of the target environment(s).

This is the difficult part when developing software for different communities.

Understanding The Threats/Risks

Risk = Threat + Probability + Impact

For mass market applications, there are many possible target environments and we must argue in terms of threats.

For bespoke systems, the target environment (and hence the impact) is known and we can analyze risks.

Requirements are derived from the threat/risk analysis and should evolve as threats evolve....

Functional Requirements

- We can distinguish between functional and non-functional security requirements.
- Traditional security functional requirements are reasonably well understood:
  - Confidentiality.
  - Data and session integrity.
  - Availability.
  - Accountability.....
- Certain requirements are more difficult:
  - Which data is considered as private?
  - There are competing requirements – security vs. privacy.

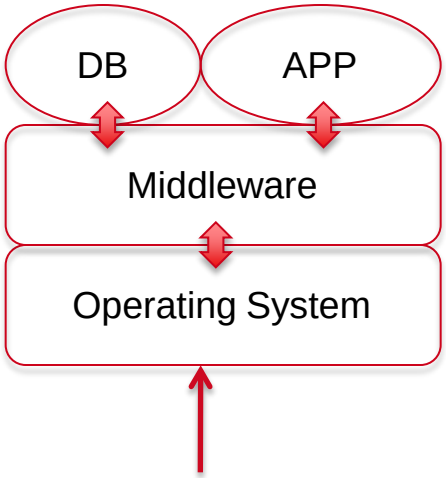
Non-Functional Requirements

- Many real-life security issues arise out of a poor definition of the non-functional requirements.
- Key questions to ask in this area are:
  - Are assumptions on the operational constraints reasonable?
  - Is the system sufficiently scalable to cope with expected and unexpected growth?
  - Does the system exhibit reasonable flexibility – can it be extended to include new or modified functionality?
  - Usability – Does the system make any unreasonable demands on the users..

IV.
Some Design Considerations

Software Layers

- Different software layers perform different security functions.
- This has led to a difference between infrastructure services and application layer services.
- In the future we should strive for a closer integration.
- We need secure services, not secure applications or secure infrastructure.



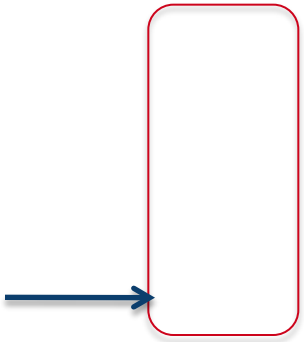
The OS is the key to everything. Nearly always, all higher layer security depends on it.
root is king.

Evolving Security Models

- Different IT architectures require different security models.
  - This applies not only to the technology, but to the associated procedures.
- Established techniques are being pushed to their limits as we try to make old models respond to new demands.
- In the next few slides, we will look at (one) example of this, before looking toward the future.

Example - Authentication

Mainframe Architecture

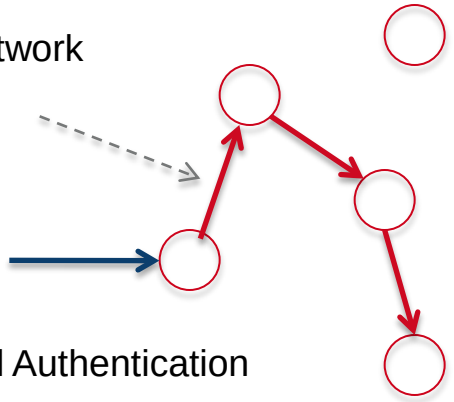


Initial Authentication

We authenticate to the OS
And we stay within the ‘box’.

Highly Distributed Architecture

This is a network connection



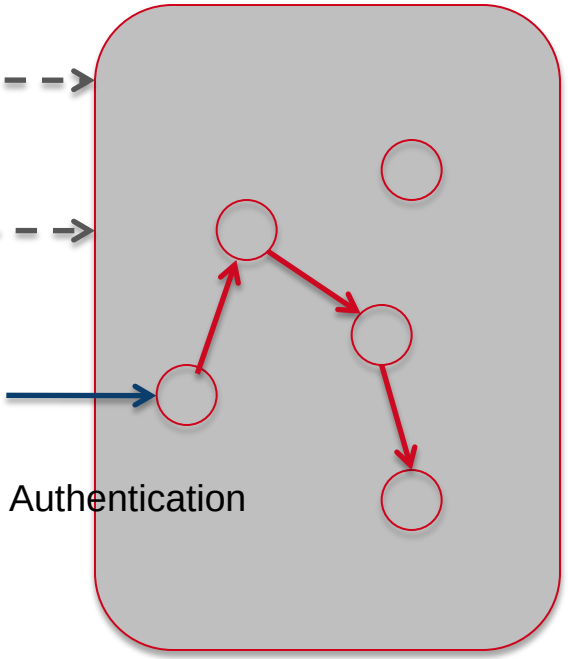
Initial Authentication

Here we need to re-authenticate
- Relay user authentication?
- Object-to-object?
- How easy is this to deploy?
- How scalable is the solution?

Authentication - Solutions

Objects are constrained to
Operate in a secure domain

All communications in the domain
run-over pre-authenticated secure
Sessions.



Note however, that we must be able
To stop invocations breaking out of the
Secure domain.

New Security Models

- As the trend towards de-centralisation continues, we will need to consider new security models.
- Peer-to-peer networks have no central point of control by definition.
  - Such networks operate on the basis of distributed trust.
  - Models based on reputation with peer review are likely.
- Cloud computing puts new demands on the scalability of applications.
  - Predicted scalability is feasible.
  - On-demand scalability will be challenging for secure applications (e.g. Key management).

Some Design Considerations

- Make realistic assumptions about the environment.
- Place sufficient emphasis on non-functional requirements – scalability, flexibility, usability.
- Think in terms of architecture not individual applications.
- End-to-End Security is more important than single system security.
- Use the principle of Defence in Depth. Relying on a single control gives no fallback solution.
- Use Compensating Controls to cover weak areas.

Methodologies

- At present, methodologies tend to be specific to a particular community.
  - Development methodologies and security methodologies are only partially integrated.
- Many current methodologies are essentially linear.
  - An iterative approach may be more appropriate.
- There is a risk of not seeing the forest for the trees.
  - Developers should be able to relate security mechanisms back to risks.

Need For a Cross-Discipline Approach

- There are many considerations in securing global distributed systems :
  - Business considerations.
  - Technical considerations.
  - Legal considerations.
  - Cultural considerations
- Different communities should be involved from the start.
  - They see different aspects of the problem.
  - This is equally true *within* areas of expertise.

I. ENISA's Contribution

Community Building

- Many of the challenges that have been mentioned require different communities to work together effectively.
- Designers, developers, implementers, administrators and users must all work together in order to achieve an effective security solution.
- A key role of ENISA is to break down barriers between different communities and to foster information exchange and cooperation.

Policy Alignment

- In order to be competitive in the software development market, policy and regulations need to be aligned with market reality.
- ENISA works closely with the Commission and Member States in a number of policy areas in order to ensure that the EU approach to information security is economically efficient.
- Examples include:
  - Privacy and Data Protection.
  - New technologies (e.g. Cloud Computing)
  - Resilience and CIIP….

Technical Work

- ENISA carries out technical work in a number of areas.
  - We are careful to avoid overlap with other institutions.
  - We focus on helping Member States to achieve concrete results at the operational level.
  - ….we are not a research institution.
- Examples include:
  - Cloud Computing
  - Secure software development
  - Mobile platforms (e.g. smart phones)
  - Mechanisms employed within the protocol stack…

Conclusions

- *Trends in system development include increasing decentralisation, global connectivity, more empowerment of the end user and shorter development lifecycles.*
- *The key challenge to developing secure systems is understanding and responding to the limitations of the target environment(s).*
- *Sufficient weight should be given to non-functional security requirements – scalability, flexibility, usability.*
- *Security design should be based on architectural principles – functionality in different software layers should be complementary.*
- *Traditional centralised security models are being pushed to their limits – new models are emerging.*
- *End-to-end security is more important than single system security for distributed environments.*