Special Topics in Security
ECE 5698

Engin Kirda
ek@ccs.neu.edu

Northeastern University

Admin News

- Course information on:
  - https://course.iseclab.org/website/ece5698
  - Account information was sent out
    - Let me know if haven't received it or cannot login
  - Labs will begin next week
- Today, we'll look at some of the basic concepts needed for Challenge 1

TA Times -- Office Hours

- TA times have been fixed for one TA
  - I'll be sending out an email
- My office hours:
  - Thursdays 4-5pm, 617 ISEC
  - Outside of that, happy to meet if you send me an email and fix a time

Where we left off last week…

- Social engineering
  - Sound did not work, so let's try again

A Collection of Hats

The Good, the Bad and the Ugly

- White Hat
  - The good guys
  - Ethical hackers, penetration testers, security researchers
  - Try to protect systems, promote and move security forward

- Black Hat
  - The bad guys
  - Break into systems, write malware, steal data, etc.

The Good, the Bad and the Ugly

- Gray Hat
  - Somewhere in between, often good intent, but legal trouble
  - Will attack first, then make an offer you can't refuse to your company
- Blue Hat
  - Microsoft term for hackers who pen test systems before launch
- Red Hat?

Hacktivism

- We all love newly coined terms:
  - Hacking + (political) activism = Hacktivism
- Step 1: Hack a system or website
- Step 2: Gain media attention, spread a message
  - Relay a political message to the public
  - Back in the day, most hackers were looking for fame or attention → hacktivism similar to “old school” hacking
  - Profit-driven cybercrime / financial incentives → no hacktivism, simply crime
  - Typical: Defacement, DOS, Email campaigns, ...
  - Sometimes black hat, sometimes not

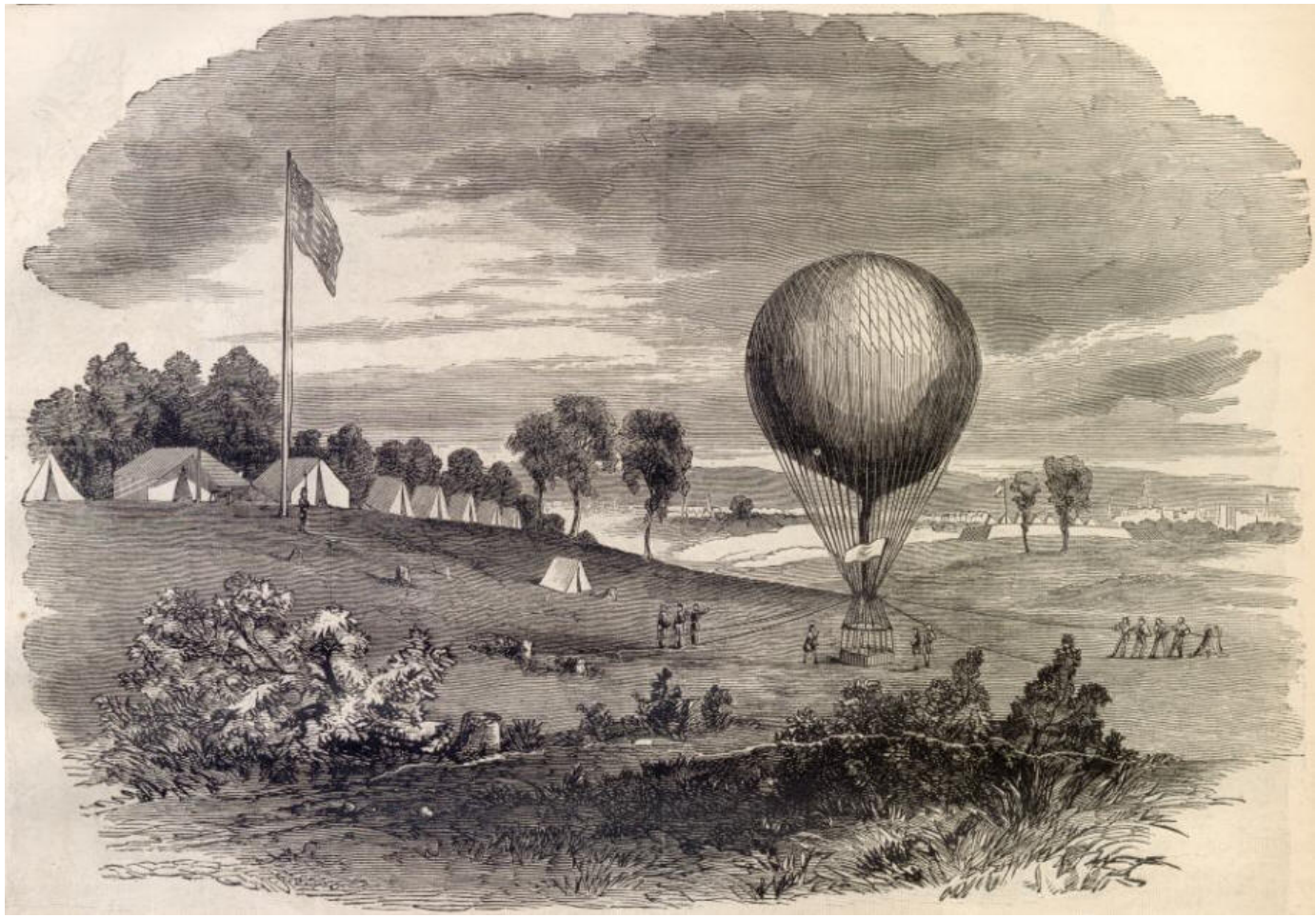
Hacktivism Example

Defacements

- Original source (2009)
  - http://www.zdnet.com/blog/security/X-hackers-deface-the-X-consulate-in-shanghai/2641
- Translates to
  - “Xinvaded our territory to kill people X. Hack done for the Xcrew of controversy! X must be punished! ! ! Hacked BY: Y”
- Defacement archive: http://www.zone-h.org
- Famous hacktivism groups
  - Cult of the Dead Cow, Anonymous, CCC (Chaos Computer Club), ...

Some general technical issues and concepts as a quick preparation for first lab…

Reconnaissance



Reconnaissance

- Definition:
- "*An inspection or exploration of an area, especially one made to gather military information.*"
- May start with just one piece of information
  - Company name, IP address, name of employee, email address, URL, …
- Objective: Mapping of target to a set of IP addresses
  - *Relevant: must be associated to target*
  - *Accessible: must be able to attack it*

Methodology

- Reconnaissance usually encompasses these steps
  - Intelligence gathering
    - Find out more about structure of target (as organization)
  - Footprinting
    - Find out individual computers of target (IP addresses, DNS names)
  - Verification
    - Check if target candidates are actually alive and reachable

Reconnaissance Steps

- Intelligence Gathering
- Footprinting
- Verification

Intelligence Gathering

- Learn about target's structure and organization
  - Involves relatively high amount of manual work
  - Extends scope of security analysis, may reveal new parts of target
  - Less technical information, but important
  - If you make mistakes in this phase, you might never see vulnerable parts of the target
- Output
  - Should reflect entire target
  - DNS domain names

Intelligence Gathering Sources

- Search engines
- Employees!
- Target's web site
- Social Networks & Co.
  - Casual: Facebook, Twitter / Professional: LinkedIn
- Press reports
- Infos on parent or sister companies
- Query business databases
- Many freely available
- Collect DNS domain names along the way

Databases

- Netcraft (www.netcraft.com)
  - Use “SearchDNS“ feature
  - Much information, try to click on a net block entry
  - Not reliable (no authoritative source!)
- Whois
  - Often limited infos
  - Some providers allow wildcard searches
  - Others do not give any infos
  - Also not reliable, but often good starting point

Reconnaissance Steps

- Intelligence Gathering
- Footprinting
- Verification

Footprinting

- We obtained a list of DNS names from previous steps
- Now get IP / Host name pairs
- Locality can be an advantage (close IP addresses might belong together)
- Get as many *plausible candidates as possible*
  - False positives will be removed later
- NS (nameserver) / MX (mail exchange) records as a starting point
- Guess some well-known names
    - www, mail, firewall, proxy, …

Getting DNS Records

- DNS
  - Directory of human readable / computer addresses
- DNS lookup
  - DNS Name -> IP address
- Reverse lookup
  - IP address -> DNS name
- Name server
- Computer (or program) that stores mapping and answers DNS requests
- Extract Domain Records
  - Unix tools: *host, dig*, …
  - host –t ns → Get name servers

DNS

- Zone Transfer
  - Original purpose: Replicate DNS data to a different DNS server
  - Misconfigured DNS servers, backups, etc.
  - “host –l” is your friend
  - Yields a list of name / IP mappings
  - More likely to work in a local analysis testing scenario
- Forward / reverse DNS queries
  - Answered by different databases that are maintained by different entitites
  - Domain name owner: Forward DNS database
  - IP subnet owner: Reverse DNS database
  - If entries equal: assume subnet owner = domain name owner

Zone transfer
DEMO

Reconnaissance Steps

- Intelligence Gathering
- Footprinting
- Verification

Verification

- After previous phases, we have many (DNS) names
- Aim of verification: Check if results are plausible → test them
- Use different web resources, network tools, etc. than those used for the collection
- Internet Registries
  - ARIN (US), RIPE (Europe), APNIC (Asia), AFRINIC (Africa), LACNIC (Latin America)
- Whois infos (use tools or web interface)
  - Contain real-world addresses, contact names, phone numbers, IP addresses and ranges

Scanning

Scanning Phase

- Big picture
  - We start with IP addresses of systems that belong to the target (results from reconnaissance phase)
- Aim of scanning: Find out details about these systems
  - Type of system?
  - What does it do (offered services)?
  - What is its role in the target organization (system component)?
- Scanning usually involves active techniques, might interfere with running systems
- To understand scanning, you need basic network skills
- To really understand scanning, you need excellent network skills :)

Ping Sweep

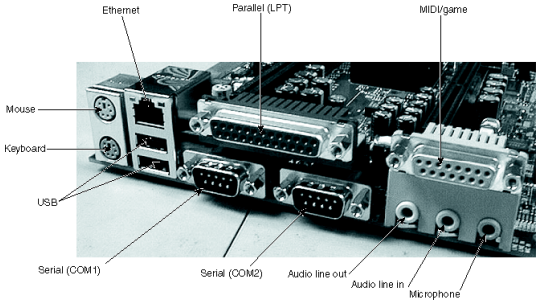
- Starting point for scanning
- Aim: Check an IP range for “live“ systems
- ICMP (Internet Control Message Protocol) ping
  - ICMP designed for network diagnosis
  - However: Often blocked due to abuse
- TCP ping (ACK ping)
  - Send a TCP ACK packet
  - RFC mandates a TCP RST packet (Reset) to be returned
- ARP “ping“ on local ethernet
- Many tools: ping, hping or *nmap*

Port Scanning

- Ports used by applications to communicate
  - TCP / UDP ports
  - Endpoints of communication
    - Think door number (port) in house (IP) on street (network)
- 16 bit unsigned integer value
  - 0-1023 (“well known“)
  - 1024-49151 (“registered“)
  - 49152-65535 (“dynamic“)
- http://www.iana.org/assignments/port-numbers

Port Scanning

- Core principle: Send a packet, then check response
  - Infer from behavior if port is open (= accepts communication)
- TCP port scan:
  - SYN scan: send SYN packet, check response:
    - SYN/ACK → port is open
    - RST → closed
  - TCP handshake is not finished → often prevents logging of connection
- Other TCP flags: URG, PSH, FIN
- Different variations of the TCP flags
  - Lead to different response behavior, targeted at services, firewalls, ...
  - Considered different scans
  - Xmas Scan (Christmas tree) → Set URG, PSH, FIN
  - Null Scan → Set no flags (if no response, port is open!)
  - Balance between raising anomaly alerts / better scanning results



Port Scanning

- UDP scan
  - Send an “empty“ UDP packet (header) to target port
  - Return type: UDP -> Open, No answer -> open / filtered, Port unreachable -> Closed

- Banner grabbing
  - Often, port scanning is not possible
  - Initial “Welcome“ message of services often leaks infos
  - Example: Connect to service manually via *telnet*

Banner Grabbing
DEMO

Stealthy Scanning

- Stealth considerations for security analysis?
  - IDS, lockouts → you want to continue
  - Avoid inadvertent DOS attacks
    - Bandwidth, # of connections
- SYN scan better than establishing full connections
- FTP bounce scans
  - Use FTP server's PORT command to proxy requests
- Zombie scan
  - Perform scan over a third party (Zombie) via spoofed SYN packets

OS Fingerprinting

- Aim: Find out which operating system runs on a given system
- Check combinations of open ports
  - Running services often indicate specific OS versions / families
- Timing behavior might also reveal infos
- Different network stack behavior
- Passive fingerprinting
  - Do not send anything to the network, and causes no indirect traffic (lookups)
  - P0f: http://lcamtuf.coredump.cx/p0f.shtml
- Active fingerprinting
  - Send packets to the network
  - nmap -O

Application Fingerprinting

- Same principle as OS fingerprinting
- Aim: Get information on specific applications and version numbers
- Analyze banner information
- Send application specific requests
  - check small deviations in response behavior
  - or response syntax
- Tool: nmap
  - Check for the switches yourself ;)
- Interesting read on OS and app fingerprinting
  - http://www.sans.org/reading_room/whitepapers/protocols/os-application-fingerprinting-techniques_1891

Scanning Tools

- You may already have noticed that tools are essential for port scanning and finding services
  - Tools capture knowledge of networking devices, software, applications, ...
  - Ideally, you could spend as much effort as needed on each target and do manual testing
  - Constraints: Time / money / IDS …
- *Effective* scanning is crucial in practice
  - Concentrate efforts on systems that are responsive or that appear valuable
  - Use automated tools, write scripts yourself
  - Telnet is always your friend…

nmap

- http://nmap.org/
- Powerful network scanning tool
  - Can be used for many general-purpose network probing / scanning tasks
  - We recommend that you use nmap for Challenge 1 (“Swiss Army knife“ for security professionals)
  - Useful in both, recon and scanning phases
  - Available for many systems (even smart phones)
- Security people *have* to be familiar with this tool
  - Many options, nmap captures the network behavior of many different systems and applications → expert knowledge
  - Even useful with user access permissions

More Tools

- tcpdump
  - WinDump for Windows
  - Read network packets, apply filters → sniff traffic
- traceroute
  - Tells you how the IP routing looks like for a given IP address
  - Increase TCP TTL (Time to Live) for each hop, record sources of "Destination Unreachable" senders until final target reached → results in "route"
- amap
  - http://freeworld.thc.org/thc-amap/
  - Application protocol detection
- telnet
  - Precursor of SSH, no encryption → useful for manual testing

Questions?

Let’s continue with more concepts…

Design and Architectural Principles

Overview

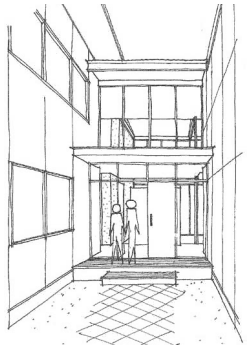
- Security issues at various stages of application life-cycle
  - mistakes, vulnerabilities, and exploits
  - avoidance, detection, and defense
- Architecture
  - security considerations when designing the application
- Implementation
  - security considerations when writing the application
- Operation
  - security considerations when the application is in use

Architecture – A Quick Recap

- Software architecture

  A representation of an engineered software system, and the process and discipline for effectively implementing the design(s) for such a system

- Representation
  - architecture concerned with components and their relationships
- Process
  - steps are provided that describe how to change design within set of constraints
- Discipline
  - set of principles how to design system within constraints



Architecture – A Quick Recap

- Software architecture has emerged as crucial part of design process
  - much work was done in the early 90s
  - today, there are research issues such as product family architectures, architectural description languages, flexibility, fault tolerance, etc.

- Software architecture encompasses the structures of large software systems
  - architectural view is abstracted
  - mostly concerned with interface descriptions (behavior)
  - distills details of implementation (such as algorithmic aspects and data representation)

Security Architecture

- What is security architecture?

  A *body* of high-level *design principles* and decisions that allow a programmer to say "Yes" with confidence and "No" with certainty

  A *framework* for *secure design*, which embodies the four classic stages of information security: *protect*, *deter*, *detect*, and *react*

- Security is a measure of the architecture's ability to resist unauthorized usage
  - at the same time, services need to be provided to legitimate users

What happens if architecture is flawed?

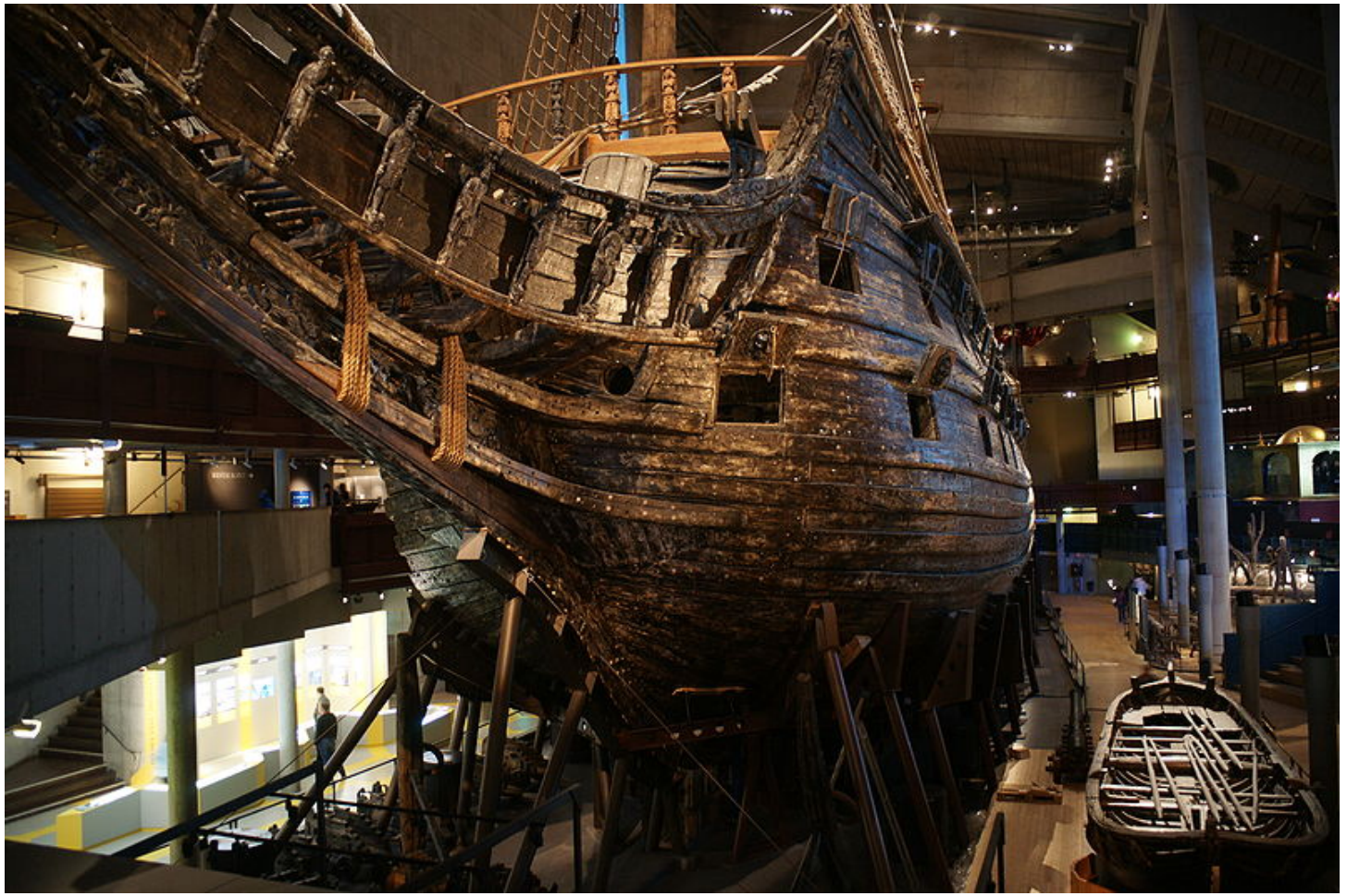
- Some history: The Swedish warship Vasa
  - now in Stockholm, Vasa Museum
  - a solemn reminder for engineers
  - the ship was built well, but its architecture was *flawed*
  - on its first voyage, it fired its guns to salute the port and…

- So what does Vasa have to do with security?
  - your code might be engineered well, but if your architecture

    is bad from a security point of view, your system may be

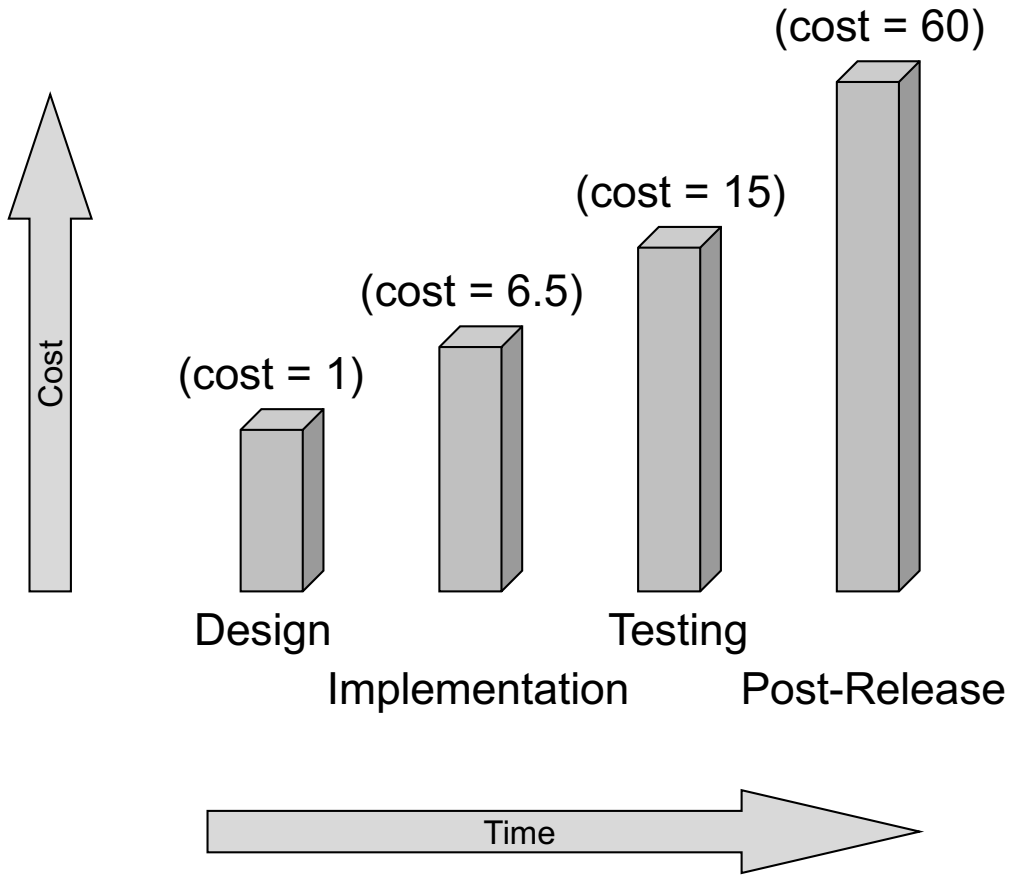
    broken by attacker
  - e.g., peer-to-peer systems

The Vasa (Stockholm, Sweden)



Architecture is Important

Cost of fixing security flaws during different development phases



Security as an Architectural Goal

- Software architects are often charged with the goal of making future-proof architecture design decisions
  - i.e., ability to accommodate change
  - Adding security to the architecture often has the negative impact of collapsing the levels of abstraction
    - Elevating low-level design decisions to a higher and often wrong level
  - Poor architecture has caused many of these partial myths to crop up, e.g.
    - Security causes huge performance problems
    - Security increases system management complexity
    - Security for legacy systems is too costly