Welcome to

12. Security Design

KEA Kompetence OB2 Software Security 2019

Henrik Lund Kramshøj hlk@zencurity.com @kramse

Slides are available as PDF, kramse@Github

12-security-design.tex in the repo security-courses

Plan for today

Subjects

- How to reach the goal of secure design
- Security Principles for software
- Principle of least privilege, fail-safe defaults, separation of privilege etc.
- Files, objects, users, groups and roles
- Security by Design
- Privacy by Design
- Benchmarking standards
- CIS controls Center for Internet Security

Exercises

- How should software be designed today

Reading Summary

We will use these as a reference

Security by Design Principles

`https://www.owasp.org/index.php/Security_by_Design_Principles`

Selected as OWASP has existed for many years, expect site to survive

Likewise we will use the Wikipedia article about *Privacy by Design*

`https://en.wikipedia.org/wiki/Privacy_by_design`

Other resources used today

- This in danish summarizing the implications of General Data Protection Regulation (GDPR)

  https://www.dubex.dk/aktuelt/nyheder/det-skal-du-vide-om-privacy-by-design-ny-vejledning
- This one which recommends doing Privacy Impact Assessment (PIA)

  https://itb.dk/persondataforordningen/privacy-by-design-default/
- ENISA, EU security office, https://www.enisa.europa.eu/publications/privacy-and-data-protection-by-design and https://www.enisa.europa.eu/publications/big-data-protection

A lot of this is because of GDPR

Goals: Foundation and design

.

- How to reach the goal of secure design by listing Security Principles for software like Principle of least privilege, fail-safe defaults, separation of privilege etc.
- Security by Design and Privacy by Design requires a lot of thought and perhaps Benchmarking standards like CIS controls Center for Internet Security can help
- Security Principles for software
- Some repetition, hope it gives a nice roundup for the course

Picture from Unsplash kyler-trautner-693525-unsplash.jpg

Weak Structural Security

Example design flaws:

- Large Attack surface
- Running a Process at Too High a Privilege Level, dont run everything as root or administrator
- No Defense in Depth, use more controls, make a strong chain
- Not Failing Securely
- Mixing Code and Data
- Misplaced trust in External Systems
- Insecure Defaults
- Missing Audit Logs

Secure Programming for Linux and Unix Howto

More information about systems design and implementation can be found in the free resource:

Secure Programming for Linux and Unix HOWTO, David Wheeler

`https://dwheeler.com/secure-programs/Secure-Programs-HOWTO.pdf`

Chapter 5. Validate All Input details input validation in the context of Unix programs

Chapter 6. Restrict Operations to Buffer Bounds (Avoid Buffer Overflow)

Chapter 7. Design Your Program for Security

Example applications from Microsoft

How to get ahead? - use existing good examples!

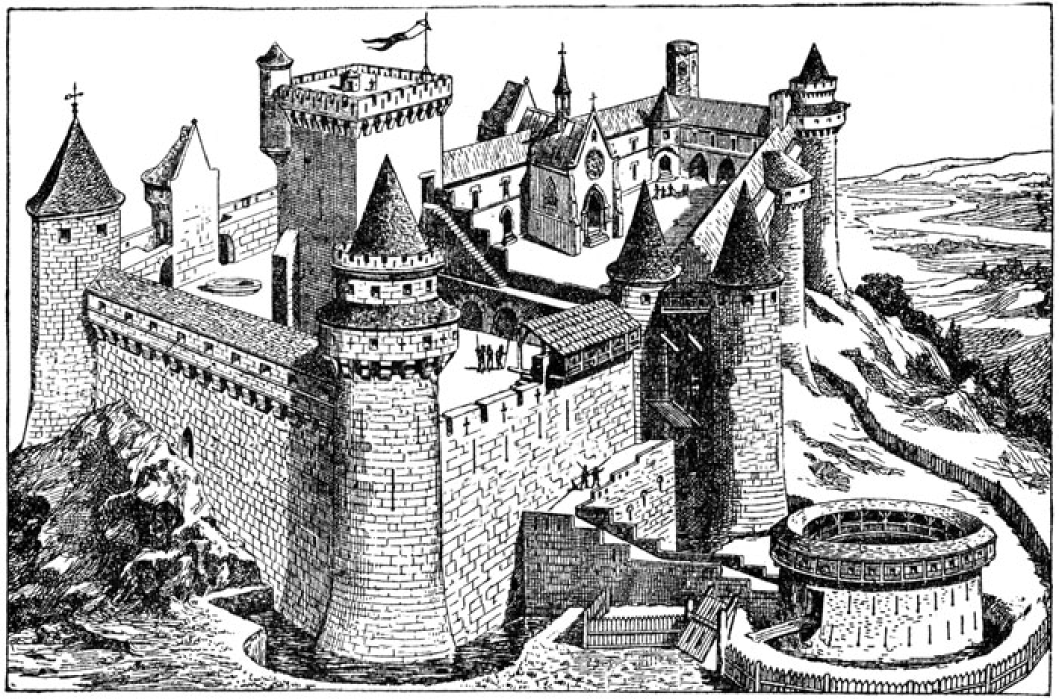
Microsoft has released sample applications.

Secure Development Documentation Learn how to develop and deploy secure applications on Azure with our sample apps, best practices, and guidance.
Get started Develop a secure web application on Azure

Source: `https://docs.microsoft.com/en-us/azure/security/develop/`

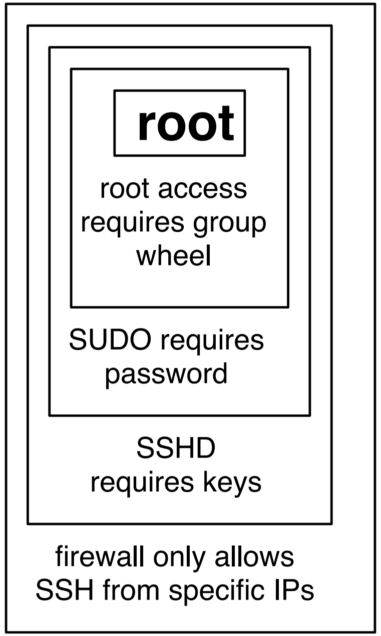
Yes, this describes how to run Alpine Linux on their Azure Cloud.

Defense in depth



Picture originally from: `http://karenswhimsy.com/public-domain-images`

Defense in depth - layered security


root
root access requires group wheel
SUDO requires password
SSHD requires keys
firewall only allows SSH from specific IPs

Multiple layers of security! Isolation!

Principle of Least Privilege

Definition 14-1 The *principle of least privilege* states that a subject should be given only those privileges that it needs in order to complete the task.

Also drop privileges when not needed anymore, relinquish rights immediately

Example, need to read a document - but not write.

Database systems can often provide very fine grained access to data

Principle of Least Authority

Definition 14-2 The *principle of least authority* states that a subject should be given only the authority that it needs in order to complete its task.

Closely related to principle of least privilege

Depend if there is distinction between *permission* and *authority*

Permission - what actions a process can take on objects directly

Authority - as determining what effects a process may have on an object, either directly or indirectly through its interactions with other processes or subsystems

Book uses the example of information flow, passing information to second subject that can write

Principle of Fail-Safe defaults

Definition 14-3 The *principle of fail-safe defaults* states that, unless a subject is given explicit access to an object, it should be denied access to that object.

Default access *none*

In firewalls default-deny - that which is not allowed is prohibited

Newer devices today can come with no administrative users, while older devices often came with default admin/admin users

Real world example, OpenSSH config files that come with `PermitRootLogin no`

Principle of Economy of Mechanism

Definition 14-4 The *principle of economy of mechanism* states that security mechanisms should be as simple as possible.

Simple $->$ fewer complications $->$ fewer security errors

Use WPA passphrase instead of MAC address based authentication

Principle of Complete Mediation

Definition 14-5 The *principle of complete mediation* requires that all accesses to objects be checked to ensure that they are allowed.

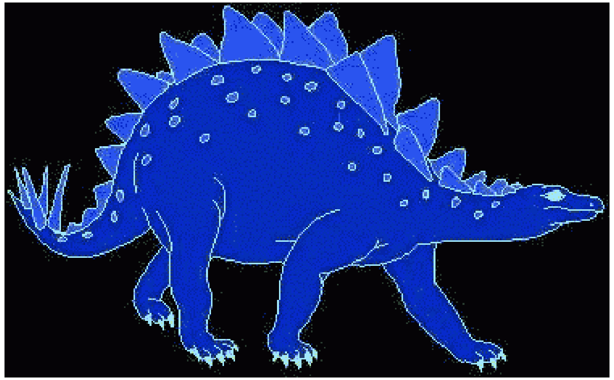
Always perform check

Time of check, time of use

Example Unix file descriptors - access check first, then can be reused in the future

Caching can be bad.

Principle of Open Design



Source: picture from `https://www.cs.cmu.edu/~dst/DeCSS/Gallery/Stego/index.html`

Definition 14-6 The *principle of open design* states that the security of a mechanism should not depend on the secrecy of its design or implementation.

Content Scrambling System (CSS) used on DVD movies

Mobile data encryption A5/1 key - see next page

Files, objects, users, groups and roles

> Establish secure defaults There are many ways to deliver an “out of the box” experience for users. However, by default, the experience should be secure, and it should be up to the user to reduce their security – if they are allowed.
>
> For example, by default, password aging and complexity should be enabled. Users might be allowed to turn these two features off to simplify their use of the application and increase their risk.

Source: OWASP Security by Design Principles
`https://www.owasp.org/index.php/Security_by_Design_Principles`

- Applications dont run in vacuum, they run in existing environments
- Make sure to integrate with existing solutions, dont expect a clean slate
- Make it easy to use by following convention

Fail securely

```
isAdmin = true;
try {
  codeWhichMayFail();
  isAdmin = isUserInRole( "Administrator" );
}
catch (Exception ex) {
  log.write(ex.toString());
}
```

Source: OWASP Security by Design Principles
https://www.owasp.org/index.php/Security_by_Design_Principles

- DO use exception handling!
- Just make sure code does not end up giving more privileges when failing!

Separation of Duties and Function

> **Separation of duties** (SoD; also known as Segregation of Duties) is the concept of having more than one person required to complete a task. In business the separation by sharing of more than one individual in one single task is an internal control intended to prevent fraud and error.

Quote from `https://en.wikipedia.org/wiki/Separation_of_duties`

> **Separation of function**. Developers do not develop new programs on production systems because of the potential threat to production data.

Computer Security, Matt Bishop, 2019

Danish: Funktionsadskillelse

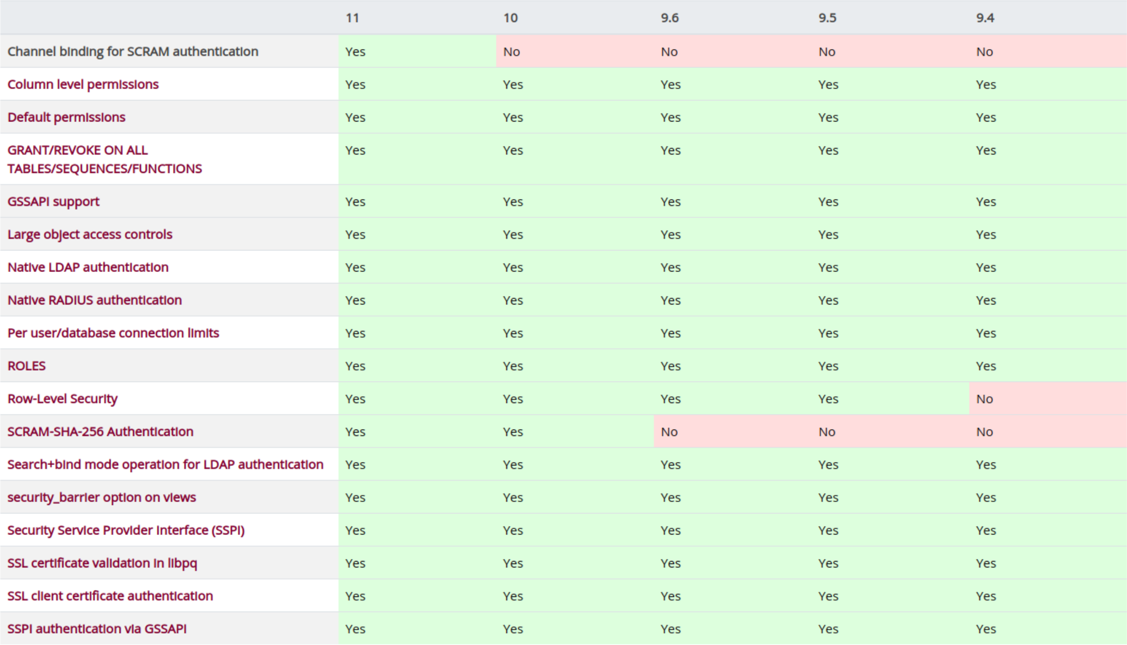
Role-based Access Control (RBAC)

In computer systems security, **role-based access control (RBAC)**[1][2] or role-based security[3] is an approach to restricting system access to unauthorized users. It is used by the majority of enterprises with more than 500 employees,[4] and can implement mandatory access control (MAC) or discretionary access control (DAC).

Role-based access control (RBAC) is a policy-neutral access-control mechanism defined around **roles and privileges**. The components of RBAC such as role-permissions, user-role and role-role relationships make it simple to perform user assignments. A study by NIST has demonstrated that RBAC addresses many needs of commercial and government organizations[citation needed]. RBAC can be used to facilitate administration of security in large organizations with hundreds of users and thousands of permissions. Although RBAC is different from MAC and DAC access control frameworks, it can enforce these policies without any complication.

Quote from `https://en.wikipedia.org/wiki/Role-based_access_control`

Example role based system: PostgreSQL security



	11	10	9.6	9.5	9.4
Channel binding for SCRAM authentication	Yes	No	No	No	No
Column level permissions	Yes	Yes	Yes	Yes	Yes
Default permissions	Yes	Yes	Yes	Yes	Yes
GRANT/REVOKE ON ALL TABLES/SEQUENCES/FUNCTIONS	Yes	Yes	Yes	Yes	Yes
GSSAPI support	Yes	Yes	Yes	Yes	Yes
Large object access controls	Yes	Yes	Yes	Yes	Yes
Native LDAP authentication	Yes	Yes	Yes	Yes	Yes
Native RADIUS authentication	Yes	Yes	Yes	Yes	Yes
Per user/database connection limits	Yes	Yes	Yes	Yes	Yes
ROLES	Yes	Yes	Yes	Yes	Yes
Row-Level Security	Yes	Yes	Yes	Yes	No
SCRAM-SHA-256 Authentication	Yes	Yes	No	No	No
Search+bind mode operation for LDAP authentication	Yes	Yes	Yes	Yes	Yes
security_barrier option on views	Yes	Yes	Yes	Yes	Yes
Security Service Provider Interface (SSPI)	Yes	Yes	Yes	Yes	Yes
SSL certificate validation in libpq	Yes	Yes	Yes	Yes	Yes
SSL client certificate authentication	Yes	Yes	Yes	Yes	Yes
SSPI authentication via GSSAPI	Yes	Yes	Yes	Yes	Yes

Feature overview security features in PostgreSQL

`https://www.postgresql.org/about/featurematrix/#security`

Security by Design

- Doing the above should result in applications which are secure by design
- Adhering to the best security principles
- Implementing security from design to deployment ensure good security

Privacy by Design

> Objectives of the report
>
> This report shall promote the discussion on how privacy by design can be implemented with the help of engineering methods. It provides a basis for better understanding of the current state of the art concerning privacy by design with a focus on the technological side.
>
> “Personal data” means any information relating to an identified or identifiable natural person—for a detailed discussion see [19]. This is related to the term personally identifiable information (PII), as e.g. used in the privacy framework standardised by ISO/IEC [125].

Source: Privacy and Data Protection by Design – from policy to engineering

`https://www.enisa.europa.eu/publications/privacy-and-data-protection-by-design`

- Next, if the application is secure, what about handling and protecting personal data

ENISA: List of Recommendations

- Policy makers need to support the development of new incentive mechanisms for privacy-friendly services and need to promote them.
- The research community needs to further investigate in privacy engineering, especially with a multidisciplinary approach. This process should be supported by research funding agencies. The results of research need to be promoted by policy makers and media.
- Providers of software development tools and the research community need to offer tools that enable the intuitive implementation of privacy properties.
- Especially in publicly co-founded infrastructure projects, privacy-supporting components, such as key servers and anonymising relays, should be included.
- Data protection authorities should play an important role providing independent guidance and assessing modules and tools for privacy engineering.
- Legislators need to promote privacy and data protection in their norms.
- Standardisation bodies need to include privacy considerations in the standardisation process.
- Standards for interoperability of privacy features should be provided by standardization bodies.

Source: ENISA Privacy and Data Protection by Design – from policy to engineering

Privacy Protection Goals

> In ICT security the triad of confidentiality, integrity, and availability has been widely accepted. ... As complement to these security protection goals, three privacy-specific protection goals were pro- posed in 2009 [172], namely unlinkability, transparency, and intervenability.

- *Unlinkability*. Unlinkability ensures that privacy-relevant data cannot be linked across domains that are constituted by a common purpose and context, and that means that processes have to be oper- ated in such a way that the privacy-relevant data are unlinkable to any other set of privacy- relevant data outside of the domain.
- *Transparency.* Transparency ensures that all privacy-relevant data processing including the legal, technical and organisational setting can be understood and reconstructed at any time. The information has to be available before, during and after the processing takes place. Thus, transparency has to cover not only the actual processing, but also the planned processing (ex- ante transparency) and the time after the processing has taken place to know what exactly happened (ex-post transparency).
- *Intervenability.* Intervenability ensures intervention is possible concerning all ongoing or planned privacy-relevant data processing, in particular by those persons whose data are pro- cessed. The objective of intervenability is the application of corrective measures and counter- balances where necessary.
- Source: ENISA Privacy and Data Protection by Design – from policy to engineering

Data oriented strategies

- Strategy #1: MINIMISE
  The most basic privacy design strategy is MINIMISE, which states that the amount of personal data that is processed 27 should be restricted to the minimal amount possible.
- Strategy #2: HIDE The second design strategy, HIDE, states that any personal data, and their interrelationships, should be hidden from plain view.
- Strategy #3: SEPARATE The third design strategy, SEPARATE, states that personal data should be processed in a distributed fashion, in separate compartments whenever possible.
- Strategy #4: AGGREGATE The fourth design pattern, AGGREGATE, states that Personal data should be processed at the highest level of aggregation and with the least possible detail in which it is (still) useful.

Privacy Impact Assessment (PIA)

> Privacy-by-design er en tilgang, der sikrer, at virksomheden indarbejder databeskyttelse som en integreret del af virksomhedens forretningsprocesser, værdikæde og produktlivscyklus. Lige fra produktionsfasen til at produktet havner hos slutbrugeren. ...
> Udarbejd en konsekvensanalyse (PIA)
> En god tilgang til at sikre privacy-by-design/default er, at udarbejde en konsekvensanalyse også kaldet en PIA (Privacy Impact Assessment), der tydeliggør konsekvenserne af virksomhedens databehandling.

En PIA skal som minimum indeholde følgende:

- En beskrivelse af de påtænkte databehandlinger og deres formål.
- En vurdering af behandlingens nødvendighed og proportionalitet.
- En vurdering af risikoen for de personer, hvis persondata bliver behandlet.
- Foranstaltninger til at imødegå risici og demonstrere overholdelse af persondataforordningen.

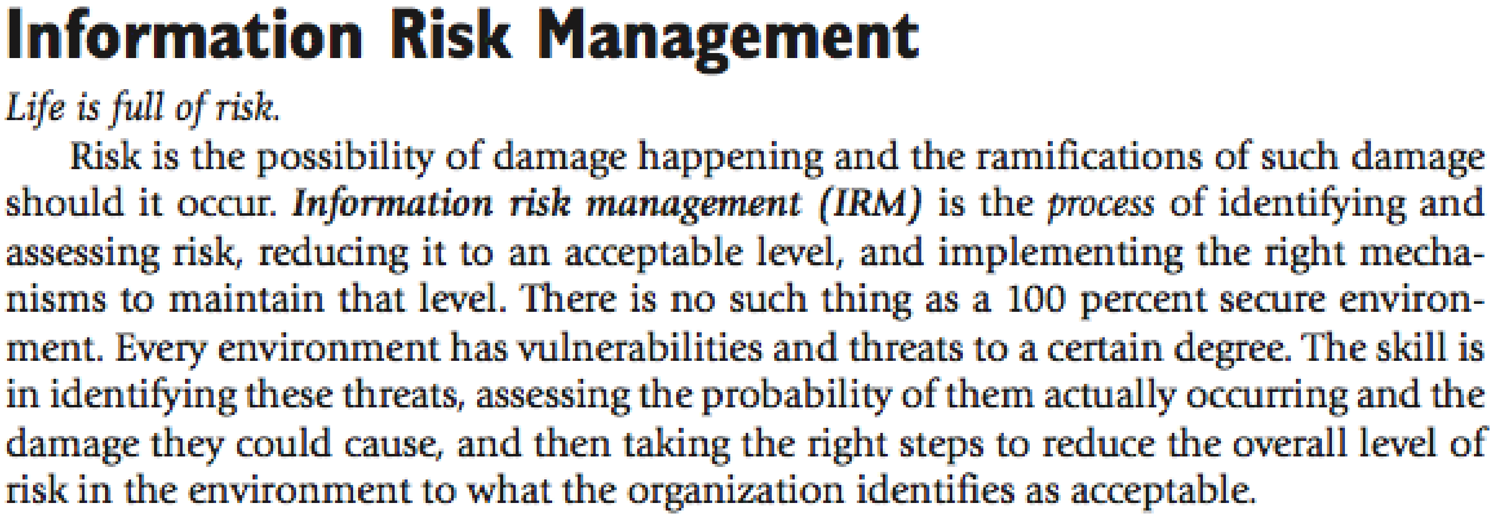
Source: IT-Branchen: Privacy-by-design/default
`https://itb.dk/persondataforordningen/privacy-by-design-default/`

Security Controls and Frameworks

Multiple exist

- CIS controls Center for Internet Security (CIS) `https://www.cisecurity.org`
- PCI Best Practices for Maintaining PCI DSS Compliance v2.0 Jan 2019
- NIST Cybersecurity Framework (CSF)
  Framework for Improving Critical Infrastructure Cybersecurity
  `https://www.nist.gov/cyberframework`
  `http://csrc.nist.gov/publications/PubsSPs.html`
- National Security Agency (NSA)
  `http://www.nsa.gov/research/publications/index.shtml`
- NSA security configuration guides
  `http://www.nsa.gov/ia/guidance/security_configuration_guides/index.shtml`
- Information Systems Audit and Control Association (ISACA)
  `http://www.isaca.org/Knowledge-Center/Risk-IT-IT-Risk-Management/Pages/default.aspx`

Risk management defined



Information Risk Management

Life is full of risk.

Risk is the possibility of damage happening and the ramifications of such damage should it occur. ***Information risk management (IRM)*** is the *process* of identifying and assessing risk, reducing it to an acceptable level, and implementing the right mechanisms to maintain that level. There is no such thing as a 100 percent secure environment. Every environment has vulnerabilities and threats to a certain degree. The skill is in identifying these threats, assessing the probability of them actually occurring and the damage they could cause, and then taking the right steps to reduce the overall level of risk in the environment to what the organization identifies as acceptable.

Source: Shon Harris *CISSP All-in-One Exam Guide*

Center for Internet Security CIS Controls

> The CIS ControlsTM are a prioritized set of actions that collectively form a defense-in-depth set of best practices that mitigate the most common attacks against systems and networks. The CIS Controls are developed by a community of IT experts who apply their first-hand experience as cyber defenders to create these globally accepted security best practices. The experts who develop the CIS Controls come from a wide range of sectors including retail, manufacturing, healthcare, education, government, defense, and others.

Source: `https://www.cisecurity.org/` CIS-Controls-Version-7-1.pdf

Center for Internet Security CIS Controls 7.1

- The five critical tenets of an effective cyber defense system as reflected in the CIS Controls are:
- **Offense informs defense:** Use knowledge of actual attacks that have compromised systems to provide the foundation to continually learn from these events to build effective, practical defenses. Include only those controls that can be shown to stop known real-world attacks.
- **Prioritization:** Invest first in Controls that will provide the greatest risk reduction and protection against the most dangerous threat actors and that can be feasibly implemented in your computing environment. The CIS Implementation Groups discussed below are a great place for organizations to start identifying relevant Sub-Controls.
- **Measurements and Metrics:** Establish common metrics to provide a shared language for executives, IT specialists, auditors, and security officials to measure the effectiveness of security measures within an organization so that required adjustments can be identified and implemented quickly.
- **Continuous diagnostics and mitigation:** Carry out continuous measurement to test and validate the effectiveness of current security measures and to help drive the priority of next steps.
- **Automation:** Automate defenses so that organizations can achieve reliable, scalable, and continuous measurements of their adherence to the Controls and related metrics. Source:

Application Software Security

CIS Control 18:
Application Software Security
Manage the security life cycle of all in-house developed and acquired software in order to prevent, detect, and correct security weaknesses.

Source: Center for Internet Security CIS Controls 7.1 CIS-Controls-Version-7-1.pdf CIS-Controls-Version-7-1.pdf

Inventory and Control of Hardware Assets

CIS controls 1-6 are Basic, everyone must do them.

> CIS Control 1:
> Inventory and Control of Hardware Assets
> Actively manage (inventory, track, and correct) all hardware devices on the network so that only authorized devices are given access, and unauthorized and unmanaged devices are found and prevented from gaining access.

What is connected to our networks?

What firmware do we need to install on hardware?

Where IS the hardware we own?

What hardware is still supported by vendor?

Source: Center for Internet Security CIS Controls 7.1 CIS-Controls-Version-7-1.pdf

Inventory and Control of Software Assets

CIS Control 2:
Inventory and Control of Software Assets
Actively manage (inventory, track, and correct) all software on the network so that only authorized software is installed and can execute, and that all unauthorized and unmanaged software is found and prevented from installation or execution.

What licenses do we have? Paying too much?

What versions of software do we depend on?

What software needs to be phased out, upgraded?

What software do our employees need to support?

Source: Center for Internet Security CIS Controls 7.1 CIS-Controls-Version-7-1.pdf

Continuous Vulnerability Management

CIS Control 3:
Continuous Vulnerability Management
Continuously acquire, assess, and take action on new information in order to identify vulnerabilities, remediate, and minimize the window of opportunity for attackers.

Source: Center for Internet Security CIS Controls 7.1 CIS-Controls-Version-7-1.pdf

Controlled Use of Administrative Privileges

CIS Control 4:
Controlled Use of Administrative Privileges
The processes and tools used to track/control/prevent/correct the use, assignment, and configuration of administrative privileges on computers, networks, and applications.

Source: Center for Internet Security CIS Controls 7.1 CIS-Controls-Version-7-1.pdf

Secure Configuration for Hardware and Software

CIS Control 5:
Secure Configuration for Hardware and Software on Mobile Devices, Laptops, Workstations and Servers
Establish, implement, and actively manage (track, report on, correct) the security configuration of mobile devices, laptops, servers, and workstations using a rigorous configuration management and change control process in order to prevent attackers from exploiting vulnerable services and settings.

Source: Center for Internet Security CIS Controls 7.1 CIS-Controls-Version-7-1.pdf

Maintenance, Monitoring and Analysis of Audit Logs

> CIS Control 6:
> Maintenance, Monitoring and Analysis of Audit Logs
> Collect, manage, and analyze audit logs of events that could help detect, understand, or recover from an attack.

... and present it, use it daily, report it to management!

Source: Center for Internet Security CIS Controls 7.1 CIS-Controls-Version-7-1.pdf

Application Software Security

CIS Control 18:
Application Software Security
Manage the security life cycle of all in-house developed and acquired software in order to prevent, detect, and correct security weaknesses.

Source: Center for Internet Security CIS Controls 7.1 CIS-Controls-Version-7-1.pdf

Exercise: Open Mike and Software Security

- What have we learnt in this course?
- Do you have tools for improving software security in your organisation?
- What are we missing?

For Next Time

Think about the subjects from this time, write down questions

Check the plan for chapters to read in the books
Most days have less than 100 pages, but some days may have more!

Visit web sites and download papers if needed

Retry the exercises to get more confident using the tools