





Welcome to

## 3. Security Policies / Confidentiality Policies

KEA Kompetence Computer Systems Security 2020

Henrik Kramselund Jereminsen [hkj@zencurity.com](mailto:hkj@zencurity.com) @kramse  

Slides are available as PDF, [kramse@Github](mailto:kramse@Github)  
3-security-policies.tex in the repo security-courses

# Plan for today



## Subjects

- Security policy
- Discretionary Access Control (DAC)
- Mandatory Access Control (MAC)
- Example Acceptable Use Policies
- Example Academic Computer Security Policy from the book
- Confidentiality Policies Bell-LaPadula Model

## Exercises

- A look at SELinux an example Mandatory Access Control system
- Find your AUP for the ISPs we use, you use, your company uses

# Reading Summary



Bishop chapter 4: Security Policies

Bishop chapter 5: Confidentiality Policies

Appendix G: Example Academic Security Policy

Browse: Campus Network Security: High Level Overview , Network Startup Resource Center

Campus Operations Best Current Practice, Network Startup Resource Center

Mutually Agreed Norms for Routing Security (MANRS)

# Security policy



A security policy defines *secure* for a system or a set of systems.

Matt Bishop, Computer Security 2019

Secure states

Transitions between states, what is allowed

Breach of security - system enters an unauthorized state

Is it possible to return from insecure to a secure state?

Book also defines Confidentiality, Integrity and Availability more precisely

*Origin integrity* authentication

Military security policy (coinfidentiality) vs commercial security policy (integrity)

# Assumptions



Any security policy, mechanism, or procedure is based on assumptions that, if incorrect, destroy the super-structure on which it is built.

Matt Bishop, Computer Security 2019

## Example, vendor patches

### Important points:

- Is patch correct? Example Spectre and heartbleed
- Vendor test environments equal to intended environments
- Installed correctly - including operator skills

# Types of Access Control



**Definition 4-13.** If an individual user can set an access control mechanism to allow or deny access to an object, that mechanism is a *discretionary access control (DAC)*, also called an *identity-based access control (IBAC)*

**Definition 4-14.** When a system mechanism controls access to an object and an individual user cannot alter that access, the control is a *mandatory access control (MAC)*, occasionally called a *rule-based access control*

Quote from Matt Bishop, Computer Security 2019

# Examples from real life systems



Example systems implementing DAC/MAC:

- Unix file permissions - DAC
- SELinux - Mandatory Access Control architecture to the Linux Kernel
- Sun's Trusted Solaris uses a mandatory and system-enforced access control mechanism

See also: [https://en.wikipedia.org/wiki/Discretionary\\_access\\_control](https://en.wikipedia.org/wiki/Discretionary_access_control)

[https://en.wikipedia.org/wiki/Mandatory\\_access\\_control](https://en.wikipedia.org/wiki/Mandatory_access_control)

# 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](https://en.wikipedia.org/wiki/Role-based_access_control)



# Confidentiality Policies Bell-LaPadula Model



The BellLa–Padula Model (BLP) is a state machine model used for enforcing access control in government and military applications.[1] It was developed by David Elliott Bell [2] and Leonard J. LaPadula, subsequent to strong guidance from Roger R. Schell, to formalize the U.S. Department of Defense (DoD) multilevel security (MLS) policy.[3][4][5] The model is a formal state transition model of computer security policy that describes a set of access control rules which use security labels on objects and clearances for subjects. Security labels range from the most sensitive (e.g., "Top Secret"), down to the least sensitive (e.g., "Unclassified" or "Public").

Quote from: [https://en.wikipedia.org/wiki/Bell%E2%80%93LaPadula\\_model](https://en.wikipedia.org/wiki/Bell%E2%80%93LaPadula_model)

Note: models like this have often been used in CISSP certification.

See also Orange Book reference:

[https://en.wikipedia.org/wiki/Trusted\\_Computer\\_System\\_Evaluation\\_Criteria](https://en.wikipedia.org/wiki/Trusted_Computer_System_Evaluation_Criteria)

# Security Enhanced Linux



Security-Enhanced Linux (SELinux) is a Linux kernel security module that provides a mechanism for supporting access control security policies, including mandatory access controls (MAC).

From: [https://en.wikipedia.org/wiki/Security-Enhanced\\_Linux](https://en.wikipedia.org/wiki/Security-Enhanced_Linux)

# Exercise



Now lets do the exercise

## SELinux Introduction up to 60min

which is number **9** in the exercise PDF.

# Policy languages



Our book uses Ponder, here is a Juniper Junos example:

```
system {
  host-name born-core-01;
  time-zone Europe/Copenhagen;
  login {
    class rancid {
      permissions [ access admin firewall interface routing secret security snmp system trace view view-configuration ];
    }
    user rancid {
      uid 2005;
      class rancid;
      authentication {
        encrypted-password "..."; ## SECRET-DATA
      }
    }
  }
}
```



Book mentions Tripwire, an alternative is Aide

Advanced Intrusion Detection Environment

open source host based file and directory integrity checker

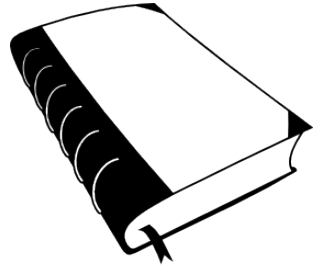
detects changes to files on the local system

Short example available from:

<https://blog.rapid7.com/2017/06/30/how-to-install-and-configure-aide-on-ubuntu-linux/>

[https://en.wikipedia.org/wiki/Advanced\\_Intrusion\\_Detection\\_Environment](https://en.wikipedia.org/wiki/Advanced_Intrusion_Detection_Environment)

# Example Academic Computer Security Policy from the book



Free graphics by Lumen Design Studio

Lets discuss the example from the book, as well as other policies

Campus Network Security: High Level Overview , Network Startup Resource Center

Campus Operations Best Current Practice, Network Startup Resource Center

Mutually Agreed Norms for Routing Security (MANRS)

<https://informationssikkerhed.ku.dk/>

# Exercise



Now lets do the exercise

## Example AUPs up to 30min

which is number **10** in the exercise PDF.

## For Next Time



Think about the subjects from this time, write down questions

Check the plan for chapters to read in the books

Visit web sites and download papers if needed

Retry the exercises to get more confident using the tools