





Welcome to

3. Security Policies / Confidentiality Policies

KEA Kompetence Computer Systems Security 2019

Henrik Lund Kramshøj hk@zencurity.com @kramse  

Slides are available as PDF, 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 <https://www.debian.org/doc/manuals/debian-handbook/sect.selinux.en.html>
- 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 superstructure 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/Mandatory_access_control

Role-based access control

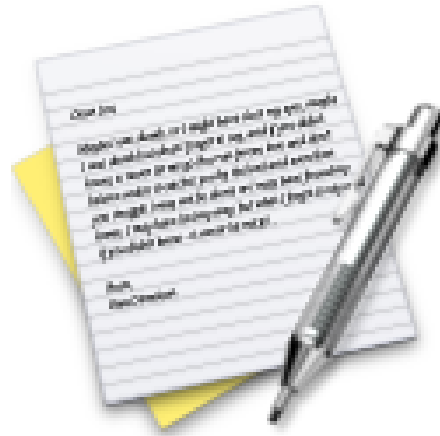


In computer systems security, **role-based access control (RBAC)**[1][2] or role-based security[3] is an approach to restricting system access to unauthorized users. It is used by the majority of enterprises with more than 500 employees,[4] and can implement mandatory access control (MAC) or discretionary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

Exercise



Now lets do the exercise

SELinux Introduction

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

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)



Exercise



Now lets do the exercise

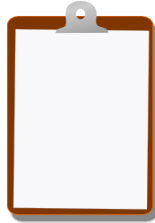
Example AUPs

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

Confidentiality Policies Bell-LaPadula Model



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