


Welcome to

3. User Accounts

KEA Kompetence Computer Systems Security 2023

Henrik Kramselund he/him han/ham xhek@kea.dk @kramse

Slides are available as PDF, kramse@Github 
3-user-accounts.tex in the repo security-courses

Goals for today



Today's goals:

- Talk about user accounts in general

Photo by Thomas Galler on Unsplash

Plan for today

Subjects

- What are user accounts – user ID
- Securing Administrative User Accounts
- Securing Normal User Accounts
- Databases: RDBMS, PostgreSQL, Deadlocks

Exercises

- Databases - discussion about Relational Database Management System RDBMS Model and NoSQL

MLSH SectionI: Setting up a Secure Linux System

Chapter 1: Running Linux in a Virtual Environment

Chapter 2: Securing Administrative User Accounts

Chapter 3: Securing Normal User Accounts

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

Accuracy vs disclosure

Lipner five commercial requirements:

1. Users will not write their own programs, but use existing production software.
2. Programmers develop and test applications on a nonproduction system, possibly using contrived data.
3. Moving applications from development to production requires a special process.
4. This process must be controlled and audited.
5. Managers and auditors must have access to system state and system logs

Available from

<https://csrc.nist.gov/CSRC/media/Publications/conference-paper/1982/05/24/proceedings-5th-seminar-dod-computer-security-initiative/documents/1982-5th-seminar.pdf>

Ken Biba (1977) proposed three different integrity access control policies.

- 1 The Low Water Mark Integrity Policy
 - 2 The Ring Policy
 - 3 Strict Integrity
- All assume that we associate integrity labels with subjects and objects, analogous to clearance levels in BLP.
 - Only Strict Integrity had much continuing influence. It is the one typically referred to as the “Biba Model” or “Biba Integrity.”

Example page 178 mentions that this was implemented in FreeBSD

Lipner provides two security levels, in the following order (higher to lower):

- Audit Manager (AM): system audit and management functions are at this level.
- System Low (SL): any process can read information at this level.

He similarly defined five categories:

- Development (D): production programs under development and testing, but not yet in production use
- Production Code (PC): production processes and programs
- Production Data (PD): data covered by the integrity policy
- System Development (SD): system programs under development, but not yet in production use
- Software Tools (T): programs provided on the production system not related to the sensitive or protected data

Non-Discretionary Controls for Commercial Applications, Steven B. Lipner, IEEE Symposium on Security and Privacy, and Fifth Seminar on the DoD Computer Security Initiative, 1982

Figure 7.

OBJECTS SUBJECTS	Prod. Data	Prod. Code	Dev. App. Prgm.	Dev. Sys. Prgm.	Tools	Sys. Prg.	Audit Trail
System Mgt. & Audit	R	R	R	R	R	R	RW
Production Users	RW	R				R	W
Application Programmers			RW		R	R	W
System Programmers				RW	R	R	W
System Control	RW	RW	RW	RW	RW	RW	W

Figure 7. Effects of the Commercial Lattice

Non-Discretionary Controls for Commercial Applications, Steven B. Lipner, IEEE Symposium on Security and Privacy, and Fifth Seminar on the DoD Computer Security Initiative, 1982

SUBJECTS	OBJECTS							
	Production Data	Production Code	Develop. Code & Test Data	Develop. Sys. Prog.	S/W Tools	Sys. Prog.	Re-pair Code	Audit Data
System Mgr.	R	R	R	R	R	R	R	RW
Prod. User	RW	R				R		W
App'n. Prog.			RW		R	R		W
Sys. Program				RW	R	R		W
Sys. Control	RW	RW	RW	RW	RW	RW	RW	W
Repair	RW	R				R	R	W

Figure 12. Effects of Commercial Lattice Model with Integrity

Non-Discretionary Controls for Commercial Applications, Steven B. Lipner, IEEE Symposium on Security and Privacy, and Fifth Seminar on the DoD Computer Security Initiative, 1982

A **well-formed transaction** from one consistent state to another consistent state.

- Constrained Data Items: CDIs are the objects whose integrity is protected
- Unconstrained Data Items: UDIs are objects not covered by the integrity policy
- Transformation Procedures: TPs are the only procedures allowed to modify CDIs, or take arbitrary user input and create new CDIs. Designed to take the system from one valid state to another.
- Integrity Verification Procedures: IVPs are procedures meant to verify maintenance of integrity of CDIs.

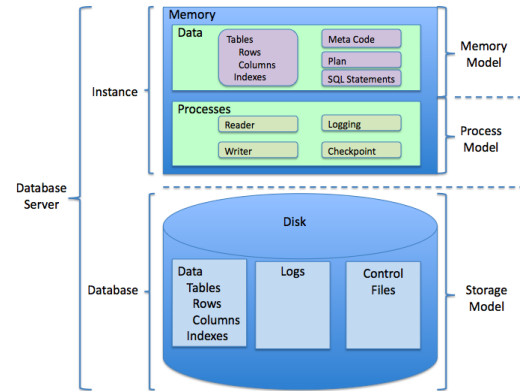
A Comparison of Commercial and Military Computer Security Policies, David D. Clark and David R. Wilson, 1987

The model uses a three-part relationship of subject/program/object (where program is interchangeable with transaction) known as a triple or an access control triple. Within this relationship, subjects do not have direct access to objects. Objects can only be accessed through programs

A Comparison of Commercial and Military Computer Security Policies, David D. Clark and David R. Wilson, 1987

See also https://en.wikipedia.org/wiki/Clark%E2%80%93Wilson_model

Relational Database Management System RDBMS



Relational Database Management System RDBMS is a common database architecture

Common examples MS-SQL, MySQL/MariaDB, PostgreSQL

Picture: By Scifipete - Own work, CC BY-SA 3.0,

<https://commons.wikimedia.org/w/index.php?curid=11506013>

https://en.wikipedia.org/wiki/Relational_database#RDBMS

	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>

Definition 7-1 A *deadlock* is a state in which some set of processes block, each waiting for another process in the set to take some action.

1. The resource is not shared (mutual exclusion)
2. An entity must hold the resource and block, waiting until another resource becomes available (hold and wait)
3. A resource being held cannot be released (no preemption)
4. A set of entities must be holding resources such that each entity is waiting for a resource held by another entity in the set (circular wait)

Often found in Relational Database Systems, if two processes want to update two tables, and each one has a write lock on one table, waiting for the write lock on the other

See also <https://en.wikipedia.org/wiki/Deadlock>

Databases - discussion about Relational Database Management System RDBMS Model and NoSQL databases, which ones do you and your company use?



Now lets do the exercise

⚠ Configure a Database - 20 min

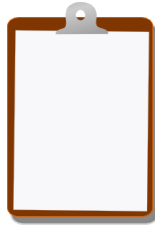
which is number **19** in the exercise PDF.



Now lets do the exercise

⚠ RBAC Access permissions on GitHub 30-45min

which is number **20** in the exercise PDF.



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