Welcome to

4. Integrity and Availability Policies

KEA Kompetence Computer Systems Security 2021

Henrik Kramselund Jereminsen hkj@zencurity.com @kramse

Slides are available as PDF, kramse@Github

4-integrity-availability-policies.tex in the repo security-courses

Plan for today

Subjects

- Accuracy vs disclosure
- The Biba Model
- Clark-Wilson Integrity Model
- Trust models
- Deadlocks
- Availability and flooding attacks
- Protection against TCP Synfloods

Exercises

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

Reading Summary

Bishop chapter 6: Integrity Policies

Bishop chapter 7: Availability Policies

TCP Synfloods - an old yet current problem, and improving pf's response to it, Henning Brauer, BSDCan 2017

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-proceedings.pdf`

Separation of duty ns 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

The Biba Model

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

Lipners Integrity Matrix Model

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

Lipners Integrity Matrix Model

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

Lipners Integrity Matrix Model

	OBJECTS							
SUBJECTS	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

Clark-Wilson Integrity Model

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 maintainance of integrity of CDIs.

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

Clark-Wilson rules

Certification rule 1 —When an IVP is executed, it must ensure the CDIs are valid.

Certification rule 2 —For some associated set of CDIs, a TP must transform those CDIs from one valid state to another. Since we must make sure that these TPs are certified to operate on a particular CDI, we must have E1 and E2.

Enforcement rule 1 —System must maintain a list of certified relations and ensure only TPs certified to run on a CDI change that CDI.

Enforcement rule 2 —System must associate a user with each TP and set of CDIs. The TP may access the CDI on behalf of the user if it is "legal".

Enforcement rule 3 -The system must authenticate the identity of each user attempting to execute a TP. This requires keeping track of triples (user, TP, CDIs) called "allowed relations".

Certification rule 3 —Allowed relations must meet the requirements of "separation of duty". We need authentication to keep track of this.

Certification rule 4 —All TPs must append to a log enough information to reconstruct the operation. When information enters the system it need not be trusted or constrained (i.e. can be a UDI). We must deal with this appropriately.

Certification rule 5 —Any TP that takes a UDI as input may only perform valid transactions for all possible values of the UDI. The TP will either accept (convert to CDI) or reject the UDI. Finally, to prevent people from gaining access by changing qualifications of a TP:

Enforcement rule 4 —Only the certifier of a TP may change the list of entities associated with that TP.

Clark-Wilson Integrity Model

> 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`

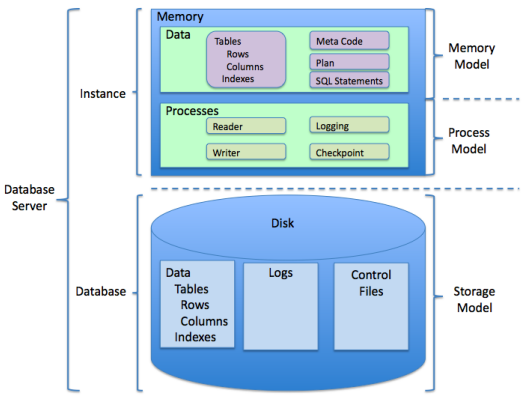
Availability Policies

An availability policy ensures that a resource or service can be accessed in some way in a timely fashion

Often expressed as *quality of service*

Denial of service occurs when this resource or service becomes unavailable

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`

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`

Deadlocks

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 help 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`

Common Discusssion

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

Fairness and starvation

Fairness policy prevents starvation, often rephrased as - process will make progress

If one process gets all resources, memory, cpu, network the others will starve - not have enough resources to progress

Compare to old operating systems Windows 3 / Mac OS 9
Cooperative multitasking vs pre-emptive multitasking

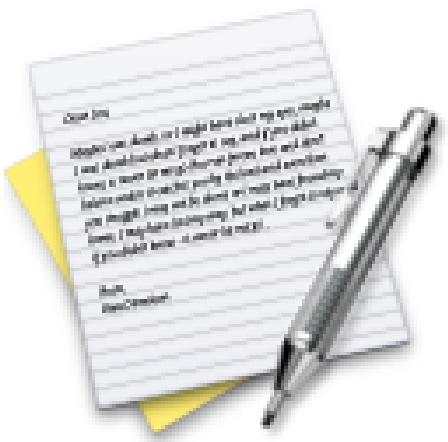
Availability and Network flooding attacks

- Our book spends some time on SYN and other flooding attacks
- SYN flood is the most basic and very common on the internet towards 80/tcp and 443/tcp
- ICMP and UDP flooding are the next targets
- Supporting litterature is TCP Synfloods - an old yet current problem, and improving pf's response to it, Henning Brauer, BSDCan 2017
- All of them try to use up some resources
- Memory space in specific sections of the kernel, TCP state, firewalls state, number of concurrent sessions/connections
- interrupt processing of packets - packets per second
- CPU processing in firewalls, pps
- CPU processing in server software
- Bandwidth - megabits per second mbps

There is a presentation about DDoS protection with low level technical measures to implement at

`https://github.com/kramse/security-courses/tree/master/presentations/network/introduction-ddos-testing`

Exercise

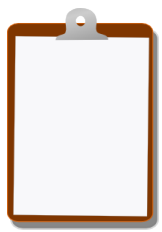


Now lets do the exercise

SYN flooding 101 - 60min

which is number **15** 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