Welcome to

4. Security Policies

KEA Kompetence Computer Systems Security 2025

Henrik Kramselund he/him han/ham hlk@zencurity.com @kramse

Slides are available as PDF, kramse@Codeberg

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

Goals

- Policies in all forms
- Databases: RDBMS, PostgreSQL, Deadlocks
- Discuss the need
- See examples

Photo by Thomas Galler on Unsplash

Plan

Subjects

- Security policies what are they
- Example Acceptable Use Policies
- Example Academic Computer Security Policy
- 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

DSH chapter 3: Policies

DSH chapter 4: Standards and Procedures

DSH chapter 5: User Education

MLSH chapter 4: Securing Your Server with a Firewall – Part 1 - NOT firewalld part!

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)

Reading Summary

MLSH SectionI: Setting up a Secure Linux System

DSH chapter 3: Policies

DSH chapter 4: Standards and Procedures

DSH chapter 5: User Education

MLSH chapter 4: Securing Your Server with a Firewall – Part 1 - NOT firewalld part!

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 cale 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

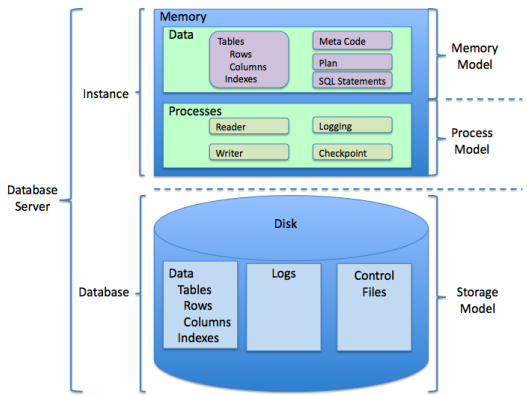
Role-based Access Control (RBAC)

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

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?

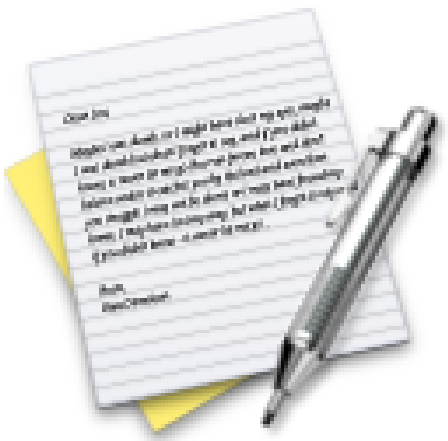
Exercise

Now lets do the exercise

⚠ Configure a Database - 20 min

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

Exercise



Now lets do the exercise

⚠ RBAC Access permissions on GitHub 30-45min

which is number 20 in the exercise PDF.

DSH chapter 3: Policies

Policies are one of the less glamorous areas of information security. They are, however, very useful and can be used to form the cornerstone of security improvement work in your organization. In this chapter we will discuss why writing policies is a good idea, what they should contain, and the choice of language to use.

- Consistency
- Distribution of knowledge
- Setting expectations
- Regulatory compliance and audit
- Sets the tone
- Management endorsement

Source: *Defensive Security Handbook*, Lee Brotherston, Amanda Berlin ISBN: 978-1-491-96038-7

Building Your Own Program

> Building a mature and strategic program from the ground up is achievable with executive support and cultural alignment. ... At the ~~end of this chapter~~ *Appendix A. User Education Templates* you will find a template slideshow for a security awareness program.

- Establish Objectives
- Establish Baselines
- Scope and Create Program Rules and Guidelines
- Implement and Document Program Infrastructure
- **Positive** Reinforcement
- Gamification
- **Define Incident Response Processes**

Source: *Defensive Security Handbook*, Lee Brotherston, Amanda Berlin ISBN: 978-1-491-96038-7

Template Policies

For ease of reading, updating, and overall management it is probably easier to produce a set of policy documents rather than a single monolithic document. SANS, for example, publishes a list of template policies that you can edit for your own needs. At the time of writing, its list of topics are: Acceptable Encryption Policy, Acceptable Use Policy, Clean Desk Policy, Disaster Recovery Plan Policy, Digital Signature Acceptance Policy, Email Policy, Ethics Policy, Pandemic Response Planning Policy, Password Construction Guidelines, Password Protection Policy, Security Response Plan Policy, End User Encryption Key Protection Policy, Acquisition Assessment Policy, Bluetooth Baseline Requirements Policy, Remote Access Policy, Remote Access Tools Policy, Router and Switch Security Policy, Wireless Communication Policy, Wireless Communication Standard, Database Credentials Policy, Technology Equipment Disposal Policy, Information Logging Standard, Lab Security Policy, Server Security Policy, Software Installation Policy, Workstation Security (For HIPAA) Policy, Web Application Security Policy

https://learning.oreilly.com/library/view/defensive-security-handbook/9781491960370/ch03.html#:-:text=SANS%2C%20for%20example,Application%20Security%20Policy

Source: *Defensive Security Handbook*, Lee Brotherston, Amanda Berlin ISBN: 978-1-491-96038-7

Example Academic Computer Security Policies

Free graphics by Lumen Design Studio

Lets discuss 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 Policies up to 45min

which is number 24 in the exercise PDF.

How do we choose a good password policy in the first place?

NIST Special Publication 800-63B Digital Identity Guidelines Authentication and Lifecycle Management:

> The ongoing authentication of subscribers is central to the process of associating a subscriber with their online activity. Subscriber authentication is performed by verifying that the claimant controls one or more authenticators (called tokens in earlier versions of SP 800-63) associated with a given subscriber. A successful authentication results in the assertion of an identifier, either pseudonymous or non-pseudonymous, and optionally other identity information, to the relying party (RP).

Source: https://nvlpubs.nist.gov/nistpubs/specialpublications/nist.sp.800-63b.pdf

-
- NIST Special Publication 800-63: Digital Identity Guidelines Frequently Asked Questions
  https://pages.nist.gov/800-63-FAQ/

Cornerstones of Authentication

The classic paradigm for authentication systems identifies three factors as the cornerstones of authentication:

- Something you know (e.g., a password).
- Something you have (e.g., an ID badge or a cryptographic key).
- Something you are (e.g., a fingerprint or other biometric data).

MFA refers to the use of more than one of the above factors. The strength of authentication systems is largely determined by the number of factors incorporated by the system — the more factors employed, the more robust the authentication system.

Source: NIST Special Publication 800-63-3 *Digital Identity Guidelines* 2021

Multi-Factor Authentication Recommended

Upon completion of the authentication process, the verifier generates an assertion containing the result of the authentication and provides it to the RP.

- Security Assertion Markup Language (SAML) assertions are specified using a mark-up language intended for describing security assertions. They can be used by a verifier to make a statement to an RP about the identity of a claimant. SAML assertions may optionally be digitally signed.
- OpenID Connect claims are specified using JavaScript Object Notation (JSON) for describing security, and optionally, user claims. JSON user info claims may optionally be digitally signed.
- Kerberos tickets allow a ticket-granting authority to issue session keys to two authenticated parties using symmetric key based encapsulation schemes.

Source: NIST Special Publication 800-63-3 *Digital Identity Guidelines* 2021

Periodic password change, every 30 or 90 days

NIST Special Publication 800-63B Digital Identity Guidelines Authentication and Lifecycle Management:

> 5.1.1.2 Memorized Secret Verifiers ...
> Verifiers SHOULD NOT impose other composition rules (e.g., requiring mixtures of different character types or prohibiting consecutively repeated characters) for memorized secrets.**Verifiers SHOULD NOT require memorized secrets to be changed arbitrarily (e.g., periodically**). However, verifiers SHALL force a change if there is evidence of compromise of the authenticator.

Source: https://nvlpubs.nist.gov/nistpubs/specialpublications/nist.sp.800-63b.pdf

- Periodic change has been the norm, but not anymore!
- We can now detect *outdated* policies, if they require this – they need to be updated
- NIST Special Publication 800-63: Digital Identity Guidelines Frequently Asked Questions https://pages.nist.gov/800-63-FAQ/

Password expiration no longer recommended

Q-B05: Is password expiration no longer recommended?
A-B05: SP 800-63B Section 5.1.1.2 paragraph 9 states:
"Verifiers SHOULD NOT require memorized secrets to be changed arbitrarily (e.g., periodically). However, verifiers SHALL force a change if there is evidence of compromise of the authenticator."

Users tend to choose weaker memorized secrets when they know that they will have to change them in the near future. When those changes do occur, they often select a secret that is similar to their old memorized secret by applying a set of common transformations such as increasing a number in the password. This practice provides a false sense of security if any of the previous secrets has been compromised since attackers can apply these same common transformations. But if there is evidence that the memorized secret has been compromised, such as by a breach of the verifier's hashed password database or observed fraudulent activity, subscribers should be required to change their memorized secrets. However, this event-based change should occur rarely, so that they are less motivated to choose a weak secret with the knowledge that it will only be used for a limited period of time.

Source: https://pages.nist.gov/800-63-FAQ/#q-b05

- Since 2021 this new version of the *NIST Special Publication 800-63: Digital Identity Guidelines* has begun to become agreed upon

Password policies

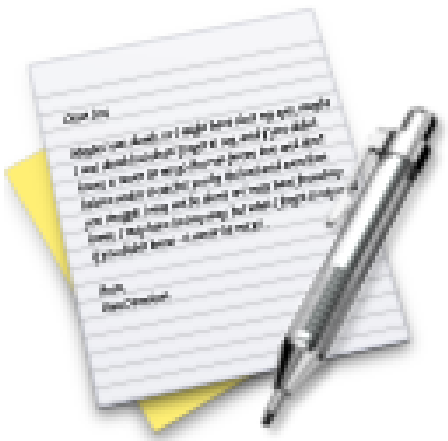
Let's take a look at passwords on Linux.

> Enforcing strong password criteria You wouldn't think that a benign-sounding topic such as strong password criteria would be so contro- versial, but it is. The conventional wisdom that you've undoubtedly heard for your entire computer career says:

- Make passwords of a certain minimum length.
- Make passwords that consist of a combination of uppercase letters, lowercase letters, numbers, and special characters.
- Ensure that passwords don't contain any words that are found in the dictionary or that are based on the users' own personal data.
- ~~Force users to change their passwords on a regular basis.~~

Source: *Mastering Linux Security and Hardening* (MLSH), third edition

Exercise

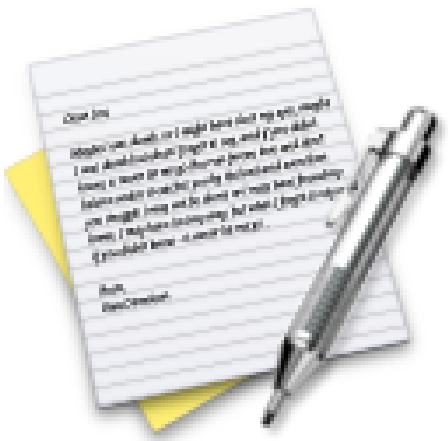


Now lets do the exercise

⚠ Password Cracking 15min

which is number 21 in the exercise PDF.

Exercise



Now lets do the exercise

⚠ Example Password policies on Linux up to 30min

which is number **25** in the exercise PDF.

Storage and Communication

> The nature of policies and procedures is meant to lend as much standard communication as possible to the organization as a whole. To do this, policies must be easily accessible. There are **many software packages** that can not only provide a **web interface for policies**, but also have **built-in review, revision control, and approval processes**. Software with these features makes it much easier when there are a **multitude of people and departments creating, editing, and approving policies**.
>
> Another good rule of thumb is to, at least **once per reviewal process, have two copies of all policies printed out**. As the majority of them will be used in digital format, there will be many policies that refer to and are in direct relation to **downtime or disaster recovery procedures**. In cases such as these, they may not be accessible via digital media so having a backup in physical form is best.

Source: *Defensive Security Handbook*, Lee Brotherston, Amanda Berlin ISBN: 978-1-491-96038-7

- I highly recommend having policies online, and NOT in a word processor document. It may be that you produce a combined/longer document *from* the online system, but edit in some Wiki or similar

Linux Aide

Many sources mention 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

How can this be applied as a policy, and what does it detect?

DSH chapter 4: Standards and Procedures

Standards and procedures are two sets of documentation that support the policies and bring them to life. In this chapter we will learn what standards and procedures are, how they relate to policies, and what they should contain.

If we consider the policies of an organization to be the "what" we are trying to achieve, standards and procedures form the "how." As with policies, standards and procedures bring with them many advantages:

- Consistency
- Distribution of knowledge
- Setting expectations
- Regulatory compliance
- Management endorsement

Source: NIST Special Publication 800-63-3 *Digital Identity Guidelines* 2021

Note: and then some standards are not optional, financial, GDPR, NIS etc. does have some requirements

Information Security Management System (ISMS)

> An information security management system (ISMS) represents the collation of all the interrelated/interacting information security elements of an organization so as to ensure policies, procedures, and objectives can be created, implemented, communicated, and evaluated to better guarantee the organization's overall information security.

Source: https://en.wikipedia.org/wiki/Information_security_management

From Mission to Instructions

From a high level we can say that we have multiple documents, about security:

Mission Statements with overall ideas

Strategy – strategic considerations

Policies – generic policy documents for the whole organisation

Standards – how do we want to implement

Procedures – how do we actually configure systems, select settings, algorithms etc.

Detailed instructions – example step-by-step how to configure a firewall rule

DSH chapter 5: User Education

User education and **security awareness** as a whole is **broken** in its current state. It is best to find a way to demonstrate with the right type of metrics that you are successfully implementing change and producing a more secure line of defense.

- Broken Processes – Ebbinghaus forgetting curve
- Bridging the Gap – Repetition is a proven, successful way to bridge the gap of compliance, teaching our users real-life skills

Source: *Defensive Security Handbook*, Lee Brotherston, Amanda Berlin ISBN: 978-1-491-96038-7

MLSH chapter 4: Securing Your Server with a Firewall

As a large example today, we are going to talk about firewall policies, since they play a huge part in keeping the systems secure.

> Security is one of those things that's best done in layers. **Security-in-depth**, we call it. So, on any given corporate network, you will find a **firewall appliance separating the Internet from the demilitarized zone (DMZ)**, where your Internet-facing servers are kept. You will also find a firewall appliance between the DMZ and the internal LAN, and **firewall software installed on each individual server and client**. We want to make it as tough as possible for intruders to reach their final destinations within our networks.

Source: *Mastering Linux Security and Hardening* (MLSH), third edition

- Best practice is to turn on firewall software on all systems
- Require developers and implementers to specify stricter rules for access, no *permit any any* anymore

Firewalls are Hardening Your Systems

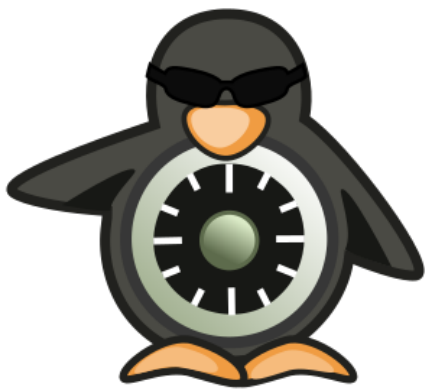
Since the focus of this book is on hardening our Linux servers, we'll focus this chapter on that last level of defense: the firewalls on our servers and clients. We'll cover both of the command-line netfilter interfaces, which are **iptables** and **nftables**.

- iptables: This replaced ipchains in Linux kernel version 2.6. It's still used in a lot of Linux distros but is rapidly disappearing.
- nftables: This is the new kid on the block and is rapidly replacing iptables. As we'll see later, it has a lot of advantages over the older iptables.

Source: *Mastering Linux Security and Hardening* (MLSH), third edition

- Lets move over to the book now. We already did some Nmap, so we can play with firewall on and off

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

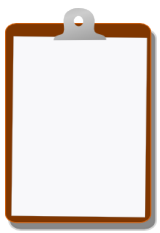
Exercise

Now lets do the exercise

ℹ SELinux Introduction up to 60min

which is number **26** 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