Authentication, Authorization, Auditing

Authentication

- Authentication can be done using
    - Something you have: Key fob, id card, token generator, authenticator apps …
    - Something you know: PIN, password, …
    - Something you are: Fingerprint, iris, …
    - Something you do: Handwriting, Voice pattern, etc
- Required strength of the passwords vary depending on the application
- For better security, we can use multi-factor authentication that combines two or more of the above techniques

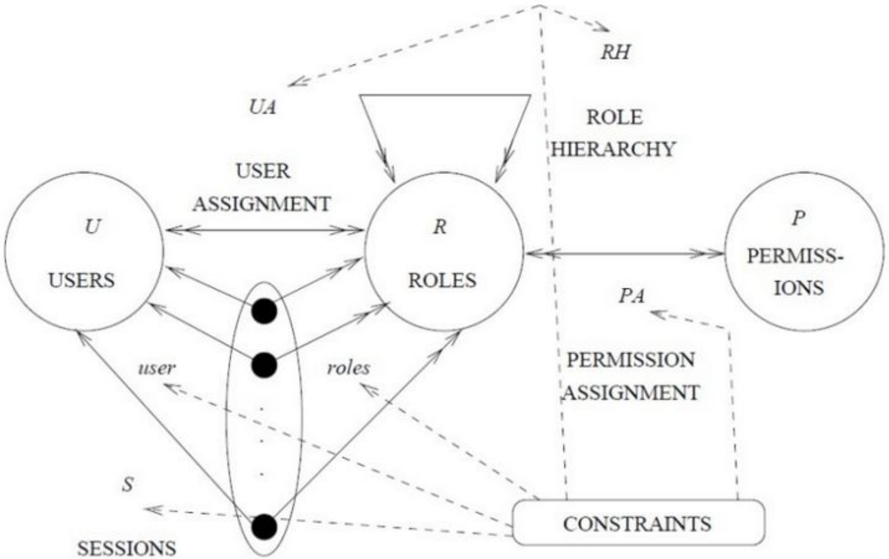
Authorization

- Authorization/Access control regulates access to resources
- There are several access control models
  - Discretionary access Control Model
  - Role-Based Access Control
  - Mandatory Access Control
  - Attribute-Based Access Control

Access Control Features

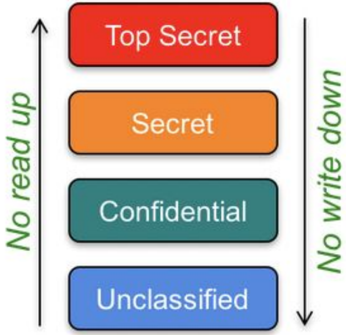
- Flexibility
- Manageability
- Dynamic
- Scalability
- Auditability
- Granularity
- Reliability
- Performance

Role-Based Access Control



Multilevel Security

- Use multiple security levels.
- Each entity is assigned a security level. Actions of an entity are decided based on its level.
- BLP Model: Used for Confidentiality
- It was originally designed for the U.S. Navy to enable users with different classification levels to use a single, shared computer system. The military uses four classification levels: unclassified, confidential, secret, and top secret.
- Every object is assigned one of those four classification levels. Every su associated with a clearance at one of those levels.



Multilevel Security

- Bell-LaPadula is summarized as no read up; no write down.
- the Bell-LaPadula model has three properties:
    - The Simple Security Property: A subject cannot read from a higher security level (no read up, NRU)
    - The *-Property (Star Property): A subject cannot write to a lower security level (no write down, NWD)
    - The Discretionary Security Property: DAC can be used with the model only after mandatory access controls are enforced.
- Tranquility principle: security labels never change during operation.

Biba Model

- A multilevel security model that is designed to address data integrity.
- Primarily concerned with imposing constraints on who can write data and ensuring that a lower-integrity subject cannot write or modify higher-integrity data.
- Biba model properties:
  - The Simple Integrity Property: A subject cannot read an object from a lower integrity level.
  - The *-Property (Star Property): A subject cannot write to an object of a higher integrity level
- The Biba model is no read down, no write up.

Type Enforcement Model

- Also known as Type Domain Model
- The model defines domains and types.
- Subjects (users and processes) are assigned to domains.
- Objects (files, sockets, devices) are assigned to types.
- Rules are specified in terms of domains and types
- Example: SELinux

Multilateral Security

- Chinese Wall Model: A Chinese Wall is a set of rules that are designed to prevent conflicts of interest.
- It is commonly used in the financial industry but is also common in law firms, consultancies, and advertising agencies.
- There are three layers of abstraction in implementing a Chinese wall:
  - Objects: These are files that contain resources about one company.
  - Company groups: These are the set of files that belong to one company.
  - Conflict classes: These identify groups of competing company groups.
- The basic rule of the Chinese Wall is that a subject can access objects from a company as long as it never accessed objects from competing companies.

Principle of Least Privilege

- Every element should have access only to the resources necessary to perform its task
- Examples that violate the principle:
    - Default permissions on a file set to read-write for all introduces the risk that people who have no business touching the file will modify it.
    - A mail server running with root privileges because it needs to listen on port 2510 but that gives it access to all other files and devices on the system as well as to privileged system calls.

Separation of Duty

- To break the privilege into multiple parts.
- In case of an application, this can be achieved by breaking the application into multiple parts. Each part runs with only the privileges it needs to perform its task. If one part becomes compromised, potential damage is limited to only that component

SELinux

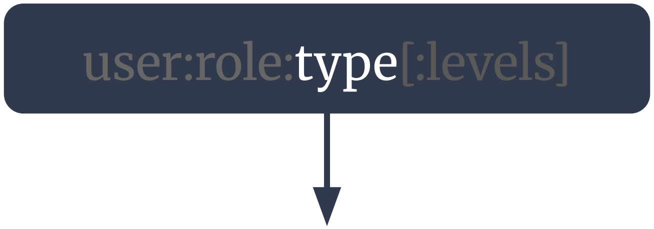
- Mandatory Access Control (MAC) based system developed by the National Security Agency (NSA).
- Works on top of Linux Discretionary Access Control (DAC), provides confinement, and helps in proactive security.
- It assigns labels to subjects and objects and specified policy in terms of these labels.
- Successfully protected systems against several zero-day attackse specially privilege escalation attacks such as DirtyCOW, ShellShock
- Also being used in Android as SEAndroid
- The policy source file is compiled and the binary policy is loaded during booting.

SELinux Label

user:role:type[:levels]

- User: Allows grouping of similar login users
- Role: A user can be assigned to multiple roles but can have only one active role at any point of time
- Type: Logical grouping of resources. Types of subjects are called domains. Basis for Type Enforcement policies.
- Level: Optional field. Used for Multi-Level Security (MLS) policies.

SELinux Label



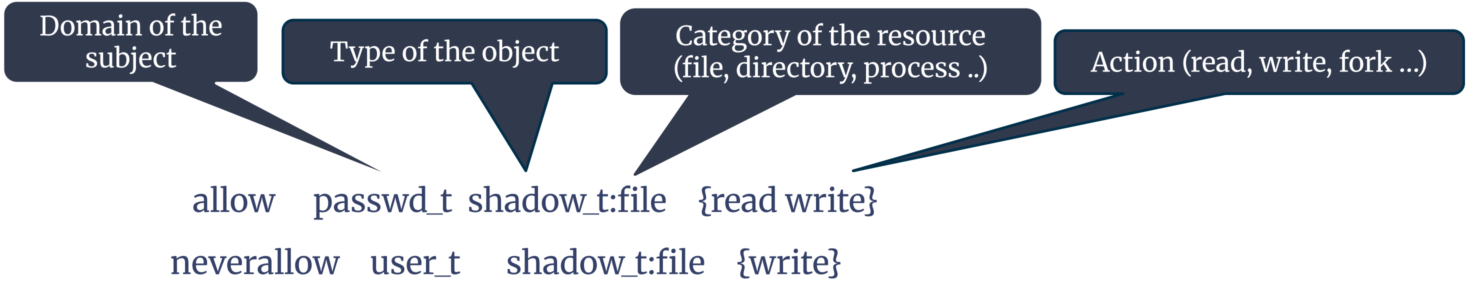
user:role:type[:levels]

Logical grouping of entities
Example: http_t, user_home_t

Type of a subject is usually
referred to as a Domain

SELinux Rules

- By default, SELinux denies all the requests
- This can be overridden by **allow** rules


Domain of the subject

Type of the object

Category of the resource (file, directory, process ..)

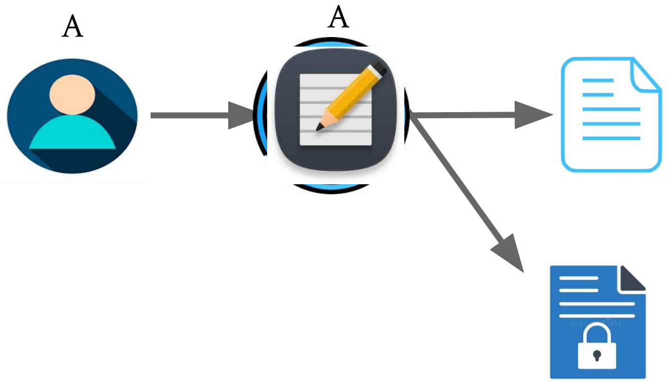
Action (read, write, fork ...)

```
allow       passwd_t  shadow_t:file   {read write}
neverallow   user_t     shadow_t:file   {write}
```

- **Neverallow** rules are used to prevent adding of corresponding allow rules into the policy
- If a policy has an allow rule corresponding to a neverallow rule, the policy compilation fails

Drawbacks of Linux DAC

- A process gets all the permissions of the user who invokes it
- Can only specify three types of accesses
- Coarse-grained access control
- Root is omnipotent



SELinux

- Additional security measure that works on top of the existing access control
- Groups related entities into *types*
- It assigns labels to subjects and objects and specified policy in terms of these labels.
- Has several object classes such as files, directories, sockets, etc.
- Each object class has a separate set of permissions

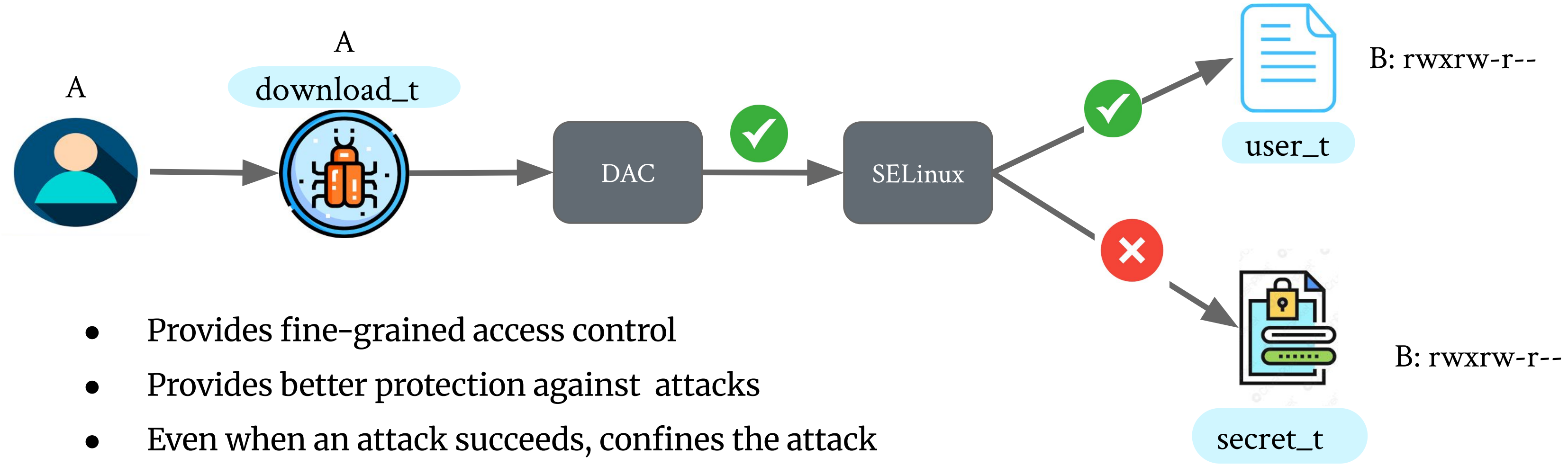
download_t

user_t

secret_t

allow download_t user_t:file {read, write}

SELinux Security



- Provides fine-grained access control
- Provides better protection against attacks
- Even when an attack succeeds, confines the attack
- Is also used in Android (SEAndroid)

Reference: “SELinux by Example” by Frank Mayer, Karl MacMillan, and David Caplan

Attribute-Based Access Control

- Entities in the system are given attributes and access control policies specify rules in terms of these attributes
- More flexible than RBAC
- Can configure other major access controls

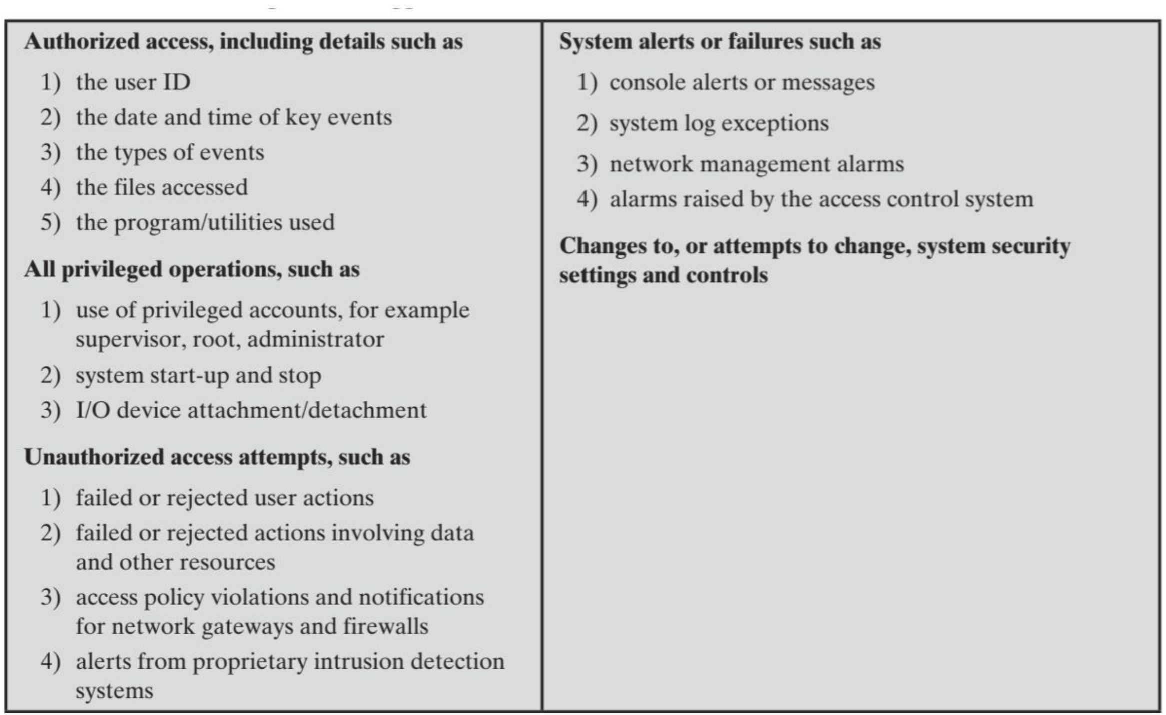
Auditing

- Auditing is the process of reviewing and analyzing the records of actions and events that affect the security and integrity of a system.
- Audit Planning is done based based on risk analysis
- Uses of Auditing
  - Resource Utilization and performance analysis
  - Compliance and regulations (Security and Privacy)
  - Provide a level of assurance concerning the proper operation of the computer with respect to security.
  - Generate data that can be used in after-the-fact analysis of an attack, whether successful or unsuccessful.
  - Provide a means of assessing inadequacies in the security service
  - Provide data that can be used to define anomalous behavior.

Logging

- Recording the actions/events that help
- Deciding the required level of granularity for logs is crucial
- There are multiple levels of logs
    - **Critical**, also known as fatal, is the highest log level. Critical is used when an application won't boot at all or in situations where there is guaranteed data corruption or loss.
    - **Error** is used for things like unhandled exceptions and other issues that impact the operation but not necessarily the application or service.
    - **Warning** is used for things that are potential problems. In isolation, a single warning likely wouldn't cause issues, but dozens of warning-level events could be an indicator of a more serious error.
    - **Information** is typically used as a summary of debug requests. Usually, you have one information log per request.
    - **Debug** is used to track the steps you took when you made a request and record diagnostically helpful details.

Common Logs



Authorized access, including details such as

1) the user ID
2) the date and time of key events
3) the types of events
4) the files accessed
5) the program/utilities used

All privileged operations, such as

1) use of privileged accounts, for example supervisor, root, administrator
2) system start-up and stop
3) I/O device attachment/detachment

Unauthorized access attempts, such as

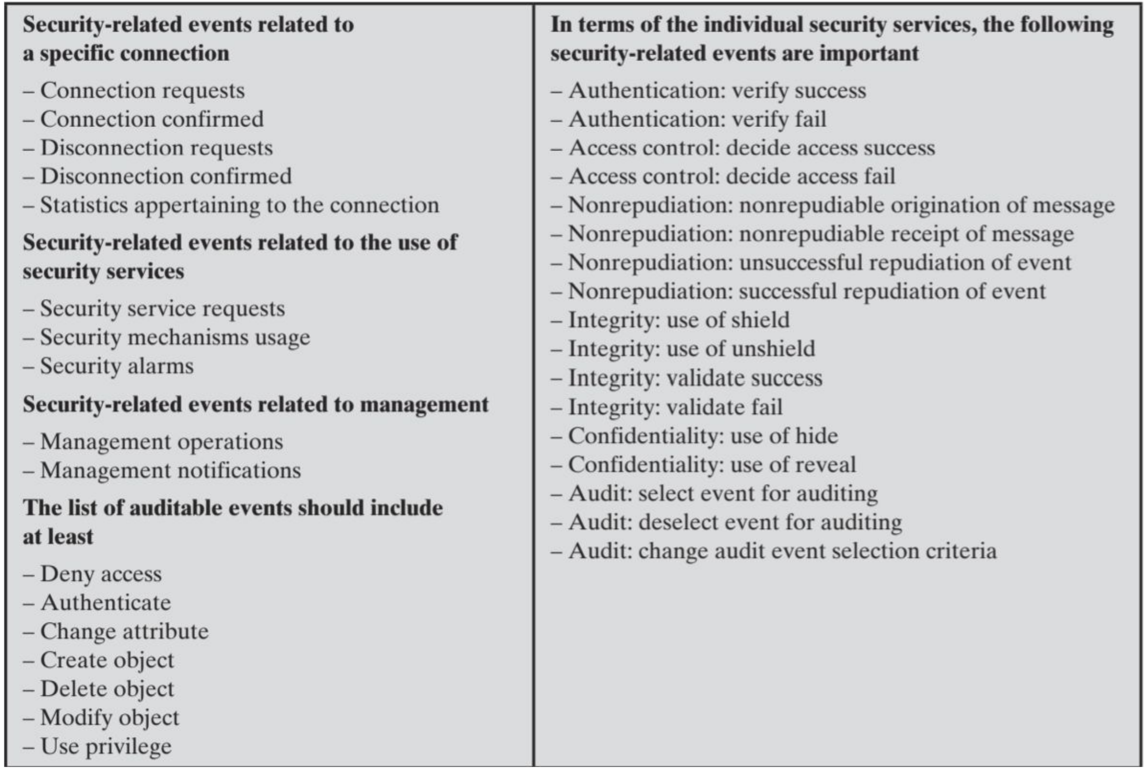
1) failed or rejected user actions
2) failed or rejected actions involving data and other resources
3) access policy violations and notifications for network gateways and firewalls
4) alerts from proprietary intrusion detection systems

System alerts or failures such as

1) console alerts or messages
2) system log exceptions
3) network management alarms
4) alarms raised by the access control system

Changes to, or attempts to change, system security settings and controls

Common Logs



Security-related events related to a specific connection

– Connection requests
– Connection confirmed
– Disconnection requests
– Disconnection confirmed
– Statistics appertaining to the connection

Security-related events related to the use of security services

– Security service requests
– Security mechanisms usage
– Security alarms

Security-related events related to management

– Management operations
– Management notifications

The list of auditable events should include at least

– Deny access
– Authenticate
– Change attribute
– Create object
– Delete object
– Modify object
– Use privilege

In terms of the individual security services, the following security-related events are important

– Authentication: verify success
– Authentication: verify fail
– Access control: decide access success
– Access control: decide access fail
– Nonrepudiation: nonrepudiable origination of message
– Nonrepudiation: nonrepudiable receipt of message
– Nonrepudiation: unsuccessful repudiation of event
– Nonrepudiation: successful repudiation of event
– Integrity: use of shield
– Integrity: use of unshield
– Integrity: validate success
– Integrity: validate fail
– Confidentiality: use of hide
– Confidentiality: use of reveal
– Audit: select event for auditing
– Audit: deselect event for auditing
– Audit: change audit event selection criteria

Logging in Linux

- In Linux, logs are stored in /var/log
- System Logs: System-related logs, such as kernel messages, boot logs, and general system activity logs, are stored directly in the /var/log directory.
- Application Logs: Logs generated by specific applications, such as Apache web server logs (/var/log/apache2/), MySQL database server logs (/var/log/mysql/), and mail server logs (/var/log/mail/), are stored in separate subdirectories.
- Service Logs: Logs generated by system services, daemons, and background processes are typically stored in subdirectories named after the corresponding services. For example, logs for the SSH service may be found in /var/log/sshd/.
- User Logs: Logs related to user activities, such as login/logout records and command history, are stored in the /var/log/ directory or its subdirectories, such as /var/log/auth.log.

Auditing

- Consider the resources available for auditing
- Prepare audit report
- Identify the issues and remediation
- Follow up on the remediation

Reference

1. https://people.cs.rutgers.edu/~pxk/419/notes/access.html
2. https://www.youtube.com/watch?v=i7XGhj3UPxE
3. “Computer Security: Principles and Practice”, William Stallings and Lawrie Brown, Pearson Publication,Second Edition.
4. https://leantechniques.com/2023/10/23/the-subtle-differences-between-logs-metrics-and-audits/