Least Privilege

Security First Principles

Wyatt Tauber
November 27th, 2023

Three Main Ideas

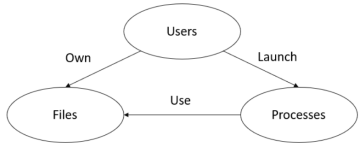
What are privileges and permissions and how are they implemented?

How is security first design supported by the principle of least privilege?

- **What role does access control play in security first design?**
- **What role do security models play in security first design?**

What are privilege escalation vulnerabilities, and how are they introduced when the principle is not followed?

Contents



Privileges and Permissions

- What is a Privilege?
- Components with Privileges
- What is Least Privilege”?
- Windows Privilege Configuration Tools
- Linux Privilege Configuration Tools

Security Models

- What is a Security Model?
- Bell-LaPadula (BLP)
- Biba
- Clark-Wilson (CW)

“No read up, no write down”

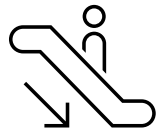
“No read down, no write up”

Access Control

- What is Access Control?
- Role Based Access Control (RBAC)
- Rule Based Access Control (RuBAC)
- Mandatory Access Control (MAC)
- Discretionary Access Control (DAC)
- Attribute Based Access Control (ABAC)

Privilege Escalation

- What is Privilege Escalation?
- Horizontal Privilege Escalation
- Vertical Privilege Escalation
- Windows Configuration Errors
- Linux Configuration Errors

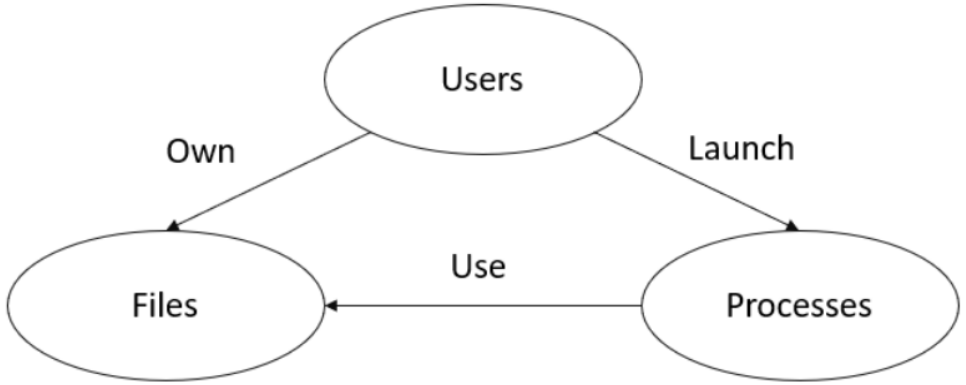


Privileges and Permissions

Concepts
Components
Configuration Tools

What is a Privilege?

A ***privilege*** is the ability of a user or service to **act on managed computer resources.**



In cybersecurity, the term is used interchangeably with ***permission***.

Components with Privileges

- ***Users* own files** and have privileges to **read/modify/delete** files and resources owned by others following the rules of a **security model**.
  - Administrators (Windows) and root (Linux) have the greatest privileges on a system.
    - SYSTEM (Windows) is an administrator account used by the operating system and services.
  - Standard users (Windows)/non-sudoers (Linux) have access to their files and processes and those they are granted access to by others.
- ***Processes*** generally operate with the **privileges of the user that runs them**, though there are **exceptions**:
  - Linux users can run processes as root using `sudo` (if they are in the sudoers file) and they can run processes as other users by using `su` or `runuser` or if the SUID/SGID bit is set on the executable.
  - Windows users can use the "Run As" (`RunAs`) dropdown option or command.
    - Services can also be configured to run as a different user or SYSTEM with the proper privileges.
    - All of these methods require the secondary logon (seclogon) service to be running.
- ***File* ownership and permission** usually follow some form of **access control**.

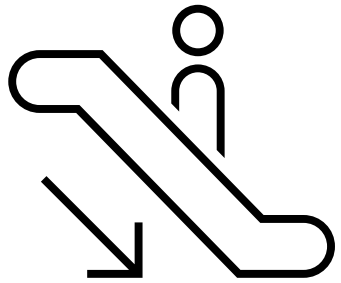
What is Least Privilege?

The ***Principle of Least Privilege*** includes:

- **Minimizing the number of privileges** granted to a user for accomplishing assigned duties
- **Disabling privileges** that are **unused** or **not required** for accomplishing assigned duties

These actions improve **accountability** and limit the potential for accidental or intentional ***privilege escalation (PrivEsc).***

Operating systems may frequently prompt users to **elevate** their privilege for tasks that have **high consequences**, such as installing software or deleting a system file.



Windows Privilege Configuration Tools

Component	Standard Management/Monitoring Tool(s)
Users	Local Users and Groups User Account Control AccessChk, AccessEnum – SysInternals
Processes	Process Explorer, Process Monitor - SysInternals Memory Integrity – Defender
Services	Services MSC Autoruns – SysInternals
Files	Properties > Security > Advanced Security Settings Local Security Audit Policy Controlled Folder Access – Defender
Registry	Registry Editor

Linux Privilege Configuration Tools

Component	Standard Management/Monitoring Tool(s)
Users	/etc/passwd /etc/sudoers User Mapping – SELinux SUID, SGID
Processes	Capabilities Process Domain – SELinux `cvechecker`
Services	`service` `cron`

Files	`chmod`
	`setfacl`
	File Labeling – SELinux
	Sticky Bit

Security Models

Concept
Classic Types

What is a Security Model?

- ***Security models*** define **rules of behavior** for an information system to enforce policies related to system security.
  - Typically involve confidentiality and/or integrity policies of the system
  - Define allowable behavior for one or more aspects of system operation
  - Technology enforces rules of behavior to ensure security goals
- **Most OSs** implement **elements** of these security models.
  - Not perfect implementations of the academic models
  - Practical implementations which are functionality consistent

Subject-Object Notation

- S is a subject, O is an object
- Property denoted by P(O) and P(S)

Bell-LaPadula (BLP)

The ***Bell-LaPadula model*** describes essential requirements for **multi-layered security** (MLS).

- Assigns a **security label (L)** to subjects and objects
- Defines rules for interactions between them
- Concerned with **confidentiality**
- Intended to prevent **unauthorized reading**

"No read up, no write down"

Property	Condition
Simple Security	S can read O if and only if L(O) ≤ L(S)
Star (*)	S can write O if and only if L(S) ≤ L(O)
Strong Star (**)	S can write O if and only if L(S) = L(O)

Biba

The ***Biba model*** describes essential requirements for **state consistency** and **data validation.**

- Assigns an **integrity level (I)** to subjects and objects
- Defines rules for interactions between them
- Concerned with **integrity**
- Intended to prevent **unauthorized writing**

"No read down, no write up"

Property	Condition
Simple Integrity	S can read O if and only if I(S) ≤ I(O)
Star (*)	S can write O if and only if I(O) ≤ I(S)
Invocation	If S reads O, then I(S) = min(I(S), I(O))

Clark-Wilson (CW)

The ***Clark-Wilson model*** **improves** the **Biba model** by describing its **properties** at the **transaction level**.

- Assigns a **subject-program-object** binding for each transaction
- Defines **rules** of two types **guaranteeing system integrity**
- Concerned with **separation of duties**
- Intended to prevent **undesirable changes by authorized subjects**

Rule Type	Description
Certification	Must separate duties between certification and enforcement When verification is executed, it guarantees all data are valid For some data, a transformation must produce valid data For all unverified data, a transformation must produce verified data or reject All transformations must log sufficiently to recreate operation
Enforcement	Must maintain list of triples and ensure only certified transformations are run. Must authenticate users. Verify that authenticated users are authorized to run a transformation on a data. Only the certifier for a transformation may change triples associated w/ that transformation.

Access Control

Concepts
Types

What is Access Control?

Access control methods are **formal presentations** of the **security policies** enforced by the operating system.

- Useful for proving theoretical limitations of systems
- Bridge the gap in abstraction between policy and mechanism

Security models align with access control methods to enforce policies, supporting the principle of least privilege.

Duty Type	Description
Fundamental (Atomic) • Objects • Security parameters	• Read, write to or modify, create, delete and load for execution • Read, define or change security parameters associated with an object, or affect how these properties are inherited by other objects or subjects that use it in any way
Complex	Sequences of fundamental operations

Role Based Access Control (RBAC)

In ***Role Based Access Control,*** the **roles assigned** to groups of users by systems administrators **determine access privileges**.

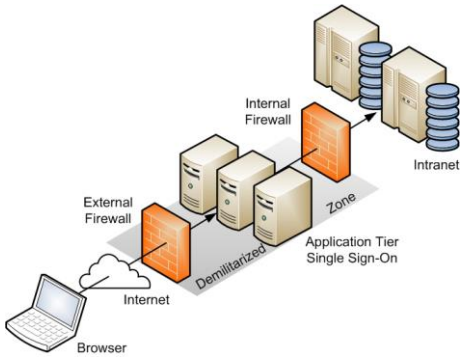
- Centralized management (+)
  - Consistent deployments
  - Ease of compliance
- Limited policy scope
  - Understandable and auditable (+)
  - Each policy may not intuitively align with a role (-)
- Significant overhead (-)
  - Each account requires a role
  - New policies require new role creation

In many ways, RBAC is like an org chart.

Rule Based Access Control (RuBAC)

In ***Rule Based Access Control***, rule lists define access parameters and apply to all subjects and objects equally.

- Flexible and granular rules (+)
  - Open-ended
  - Customizable
  - Scope is easily adjustable
- Rigid (-)
  - Strict adherence to rules
  - May not be able to account for exceptions
  - Typically contain implicit deny



RuBAC is frequently implemented as ACLs.

Mandatory Access Control (MAC)

In ***Mandatory Access Control***, access permissions and restrictions are enforced by the operating system based on a security hierarchy.

- Best model for enforcing least privilege (+)
  - Implemented by many security models
  - Added to Linux via SELinux
- Standardized format
  - Keeps security and data consistency high (+)
  - Rigid format due to a finite number of classifications (-)
- Significant overhead (-)
  - Every subject and object needs a security classification

Military and government organizations typically use MAC.

Discretionary Access Control (DAC)

In ***Discretionary Access Control***, the data owner has full control over objects they create, and they assign specific privileges to other users.

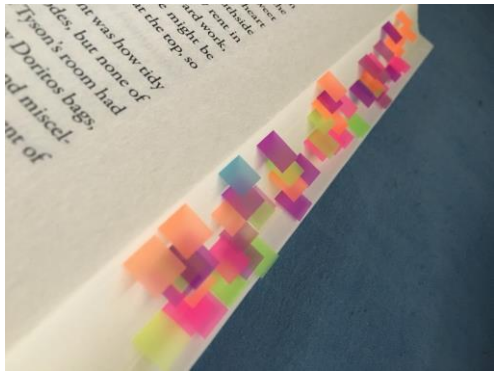
- Default access control method in most OSes (+)
  - Windows and macOS file systems default to DAC
  - Linux distros without SELinux may also default to DAC
- Not standardized
  - Doesn't require formal roles (+)
  - Implicitly trusts every user (-)
- Not scalable (-)
  - Poor management of resources
  - Increased burden on users

In DAC, users set their own access rules.

Attribute Based Access Control (ABAC)

In ***Attribute Based Access Control,*** the **attribute(s) assigned** to groups of users by systems administrators **determine access privileges**.

- Very flexible (+)
  - Provides a wide range of nuance in attribute creation and assignment
  - Works well in remote and cloud environments
- Attributes are not related to organization structure
  - Easy management and maintenance (+)
  - Difficult to organize and control without routine management (-)



ABAC is very similar to RBAC but applies multiple attributes instead of a single role.

Privilege Escalation

Concepts
Techniques
Configuration Errors

What is Privilege Escalation?

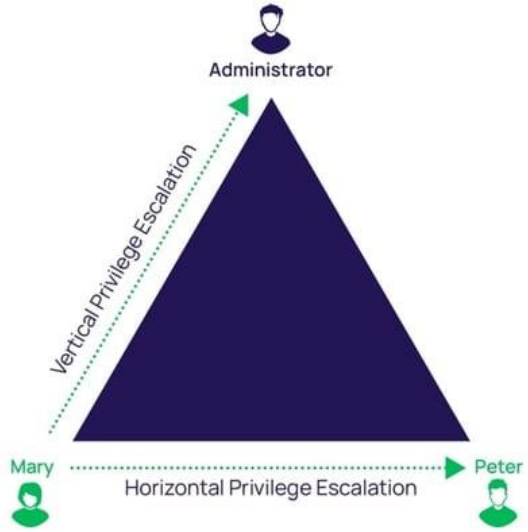
Privilege escalation (PrivEsc) is the act of exploiting a **bug**, a **design flaw**, or a **configuration error** to gain **unauthorized access** into a system and access **additional rights**, **permissions**, **entitlements**, **privileges**, or **resources** that are normally protected from an application or user.

Privilege escalation is typically part of the middle steps of the cyber kill chain, after initial access to the network is obtained.

- Exploitation
- Installation

Techniques:

- Horizontal PrivEsc
- Vertical PrivEsc



Horizontal Privilege Escalation

Horizontal PrivEsc involves **accessing another user's account** and **abusing the legitimate privileges** granted to them.

- Also called ***lateral movement*** or ***account takeover***
- Can be easier to obtain access to protected enclaves than going through administrator or root accounts first
  - Payroll and HR often have access to specific systems containing employee personally identifiable information (PII).
  - Engineering staff may have access to proprietary company information, including trade secrets.
  - Construction, maintenance, and janitorial staff may have access to building design blueprints.

Horizontal Privilege Escalation Attack

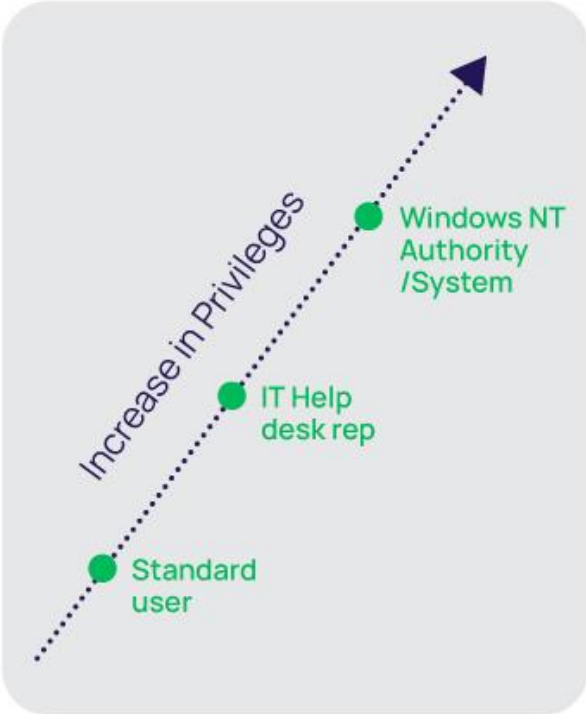
Vertical Privilege Escalation

Vertical PrivEsc involves increasing privileges by **obtaining access** to **administrative, system, or root** level **user or service** accounts.

Attackers typically use initial access accounts to search for other accounts with higher privileges (such as administrator or root).

If the attacker can get this level of access, they can control or add services that start with the privileges of the system itself.

MITRE ATT&CK defines 14 different techniques for vertical PrivEsc under ID TA0004, including those outlined next.



Vertical Privilege Escalation Attack

Windows Configuration Errors

Common Windows Configuration Errors leading to Privilege Escalation	
Privileged services editable by standard users	Service executables modifiable by standard users
Registry editable by standard users	Unquoted service paths
Passwords stored in text files	Insecure local administrator accounts
Unprotected Security Account Manager (SAM)	Third party apps that launch SYTEM-level processes or shells

Linux Configuration Errors

Common Linux Configuration Errors leading to Privilege Escalation	
System-wide cronjobs with editable paths	Folder with write permissions in PATH
Programs with significant capabilities	Insecure sudoers configuration
Improper use of SUID/SGID/Sticky Bit	Passwords stored in text files
Allowing SSH as root	PAM misconfiguration

Conclusion

Review: Three Main Ideas
Additional Resources

Review: Three Main Ideas

A privilege is the ability of a user or service to act on managed computer resources.

- **Privileges are typically implemented on users and processes.**
- **Privileges affect access to and the ability to act on files and other resources.**

The principle of least privilege supports security first design by minimizing the number of privileges granted to a user for accomplishing assigned duties and disabling privileges that are unused or not required for accomplishing assigned duties.

- ***Access control methods*** **are formal presentations of the security policies enforced by the operating system.**
- **Security models align with access control methods to enforce policies, supporting the principle of least privilege.**

Horizontal and vertical privilege escalation (PrivEsc) vulnerabilities exploit configuration errors to gain unauthorized access into a system and access additional privileges or resources when the principle is not followed.

Additional Resources

Privileges and Permissions

- Introduction to Cybersecurity First Principles - Least Privilege

Windows

- Hidden Danger: How To Identify and Mitigate Insecure Windows Services

Linux

- Linux permissions: SUID, SGID, and sticky bit
- Chapter 12. Managing sudo access

Security Models

- Introduction To Classic Security Models

Access Control

- Verification and Test Methods for Access Control Policies/Models
- Comparing Access Control: RBAC, MAC, DAC, RuBAC, ABAC

Privilege Escalation

- What Is Privilege Escalation? - ProofPoint
- What is Privilege Escalation? - CrowdStrike
- Understanding Privilege Escalation and 5 Common Attack Techniques

Windows

- Privilege escalation on Windows: When you want it and when you don't

Linux

- A Guide On Linux Privilege Escalation - The Journey Towards System Control

Lab: Privilege Escalation

Lab Description: In this lab, you will investigate, exploit, and correct many common vulnerabilities that may result from not adhering to the security first principle of least privilege. This lab will span both Windows and Linux privilege escalation techniques.

Lab Environment: The instructor has shared the "csc840-securityfirst" vApp with you in DSU's IALab. Clone the vApp to your account before starting it or making any changes so you can revert if needed.

Box Name	Type	Lab Credentials
Linux Exercises	Kali 2023.3	student / Password1!
Windows Exercises	Windows 10 Pro build 19044	student / Password1!

You are free to install whatever tools you would like to complete the assignment. Several potentially helpful tools are pre-installed on the Linux Exercises VM if needed:

- gcc/make
- Metasploit
- SCP server