INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO

TEMA:

INVESTIGACIÓN DOCUMENTAL

“Capitulo 6”

ASIGNATURA

Sistemas Operativos

PRESENTA:

SANDOVAL RODRIGUEZ LAURA AIDE

CRUZ PÉREZ CESAR GABRIEL

SAAVEDRA MATA KARLA SOFÍA

HERNÁNDEZ DE LA CRUZ EDUARDO IVÁN

GRUPO: 4CV4

PROFESOR:

Dr. Israel Salas Ramirez

INTRODUCCIÓN GENERAL

En la actualidad, la seguridad informática se ha convertido en un pilar fundamental para el funcionamiento correcto y confiable de los sistemas computacionales. Las amenazas cibernéticas son cada vez más frecuentes y sofisticadas, lo que exige un entorno de seguridad robusto y bien estructurado. En este contexto, comprender el ambiente de seguridad y cómo se integra dentro de los sistemas operativos es esencial para garantizar la integridad, confidencialidad y disponibilidad de los recursos informáticos.

El ambiente de seguridad se compone de un conjunto de políticas, herramientas, procesos y tecnologías diseñadas para proteger los sistemas informáticos frente a posibles ataques o accesos no autorizados. Este entorno debe estar preparado no solo para prevenir incidentes, sino también para responder eficazmente ante cualquier vulnerabilidad o amenaza detectada. La gestión adecuada del ambiente de seguridad permite identificar riesgos, establecer controles y mantener la confianza en los sistemas digitales.

Uno de los elementos más relevantes dentro de este entorno es el sistema operativo, ya que actúa como intermediario entre el hardware y el usuario, y regula el uso de los recursos del sistema. La seguridad en los sistemas operativos se refiere a la implementación de mecanismos que permitan resguardar tanto la información como los procesos que se ejecutan. Esto incluye funciones como el aislamiento de procesos, la autenticación de usuarios, el control de privilegios y la instalación de parches de seguridad. Sin estas medidas, los sistemas quedarían expuestos a malware, robo de información y otros tipos de ataques.

Otro aspecto clave es el control de acceso a los recursos, el cual permite definir quién puede hacer qué dentro del sistema. Existen diferentes modelos de control de acceso, como el discrecional, el obligatorio y el basado en roles, cada uno con características particulares. Estos modelos permiten asignar permisos de manera precisa y eficaz, asegurando que solo los usuarios autorizados puedan acceder a información o ejecutar determinadas acciones.

6.1 El ambiente de seguridad

El ambiente de seguridad en sistemas informáticos se refiere al conjunto de políticas, procedimientos, herramientas y tecnologías diseñadas para proteger los activos digitales de una organización. Estos activos incluyen datos, redes, sistemas operativos, hardware y usuarios. La seguridad informática busca preservar la confidencialidad, integridad y disponibilidad de la información, así como asegurar que los usuarios y sistemas funcionen de acuerdo con los permisos y roles definidos.

Un ambiente seguro contempla tanto medidas técnicas como organizacionales. Las medidas técnicas incluyen firewalls, antivirus, sistemas de detección de intrusos y encriptación de datos. Las organizacionales abarcan políticas de contraseñas, capacitaciones, auditorías y planes de contingencia.

La implementación de un ambiente de seguridad adecuado es fundamental en la era digital, ya que las amenazas como el malware, ransomware, ataques de denegación de servicio (DDoS), y la ingeniería social son cada vez más sofisticadas y frecuentes.

La seguridad no es un estado permanente, sino un proceso continuo que requiere vigilancia constante, actualizaciones, revisiones y ajustes conforme evolucionan las amenazas tecnológicas.

6.1.1 Seguridad en los sistemas operativos

Los sistemas operativos son una de las piezas fundamentales de seguridad en un sistema informático. Su papel es administrar el hardware y permitir la interacción del usuario con el sistema, pero también garantizar que las operaciones se realicen de manera segura.

Los sistemas operativos modernos implementan diversas funciones de seguridad, tales como:

- Control de acceso: Determina quién puede acceder a qué recursos y bajo qué condiciones.
- Aislamiento de procesos: Previene que un proceso interfiera o acceda a la información de otro sin autorización.
- Administración de cuentas de usuario: Permite crear y gestionar perfiles con distintos niveles de privilegios.
- Actualizaciones de seguridad: Corrigen vulnerabilidades descubiertas en el sistema operativo.

Sistemas operativos como Windows, Linux y macOS han evolucionado para incluir mecanismos de seguridad más robustos como SELinux (Linux), BitLocker (Windows), y Gatekeeper (macOS), entre otros.

La seguridad del sistema operativo también depende del comportamiento del usuario, ya que

acciones como descargar software no confiable o no aplicar actualizaciones puede comprometer la integridad del sistema.

6.1.2 Control de acceso a los recursos

El control de acceso es un componente esencial de la seguridad informática, y se refiere al conjunto de mecanismos y políticas que regulan quién puede ver o usar recursos en un entorno computacional.

Los principales tipos de control de acceso incluyen:

- Control de acceso discrecional (DAC): El propietario del recurso determina quién tiene permiso para acceder.
- Control de acceso obligatorio (MAC): Las políticas de acceso son definidas por la organización y no pueden ser modificadas por los usuarios.
- Control de acceso basado en roles (RBAC): Se asignan permisos según el rol del usuario dentro de la organización.

Los sistemas operativos implementan estos mecanismos mediante listas de control de acceso (ACL), permisos de archivos y políticas de seguridad. Además, pueden emplearse tecnologías adicionales como autenticación multifactor, tokens de seguridad, y biometría.

El objetivo del control de acceso es prevenir accesos no autorizados, asegurar el principio de privilegio mínimo, y mantener un registro de las actividades que pueden auditarse para detectar comportamientos anómalos o maliciosos.

¡Por supuesto! Aquí tienes una versión **más desarrollada y completa** de los temas que pediste. Esta versión está pensada para abarcar varias cuartillas con un lenguaje formal y universitario:

6.1.2 Control de acceso a los recursos

El control de acceso a los recursos es uno de los mecanismos más esenciales en el ámbito de la seguridad informática. Este proceso se encarga de garantizar que únicamente los usuarios autorizados puedan interactuar con determinados recursos del sistema, ya sean archivos, dispositivos, bases de datos o servicios. Su implementación es vital para prevenir accesos no deseados, modificaciones indebidas o la filtración de información confidencial.

En esencia, el control de acceso regula "quién puede hacer qué" dentro de un sistema informático. Para ello, utiliza políticas de seguridad y mecanismos técnicos que permiten

asignar permisos específicos a usuarios individuales o a grupos. Este control puede aplicarse de manera manual (configuración directa de permisos) o automática (mediante sistemas y políticas centralizadas).

Existen diversos modelos de control de acceso, cada uno con sus propias características y niveles de flexibilidad:

- **Control de Acceso Discrecional (DAC)**: Este modelo da al propietario de un recurso la capacidad de decidir qué usuarios pueden acceder al mismo y qué acciones pueden realizar. Es común en sistemas como Unix y Windows, donde el usuario puede definir permisos como lectura, escritura o ejecución.
- **Control de Acceso Obligatorio (MAC)**: Aquí, las decisiones de acceso están dictadas por políticas de seguridad centralizadas que los usuarios no pueden modificar. Este modelo es típico en entornos militares y gubernamentales, ya que permite una clasificación rigurosa de la información por niveles de seguridad.
- **Control de Acceso Basado en Roles (RBAC)**: En este modelo, los permisos no se asignan directamente a los usuarios, sino a los roles que desempeñan dentro de una organización (por ejemplo, administrador, analista, invitado). Esto permite una gestión más sencilla y coherente de los privilegios.

Además de estos modelos, existen mecanismos tecnológicos que refuerzan el control de acceso, como los sistemas de autenticación (contraseñas, biometría, tokens), firewalls, sistemas de detección de intrusos (IDS), y registros de auditoría. Todos estos elementos trabajan en conjunto para crear un entorno seguro, donde los recursos del sistema están protegidos de manera efectiva.

6.1.3 Implementación de matrices de acceso

La matriz de acceso es una herramienta teórica y práctica empleada para representar de forma clara los derechos de acceso que los diferentes usuarios o procesos tienen sobre los recursos del sistema. Se trata de un modelo que permite organizar y visualizar los permisos de manera estructurada, facilitando la implementación de controles más precisos y efectivos.

Esta matriz se compone de **filas**, que representan a los **sujetos** (usuarios, grupos o procesos del sistema), y **columnas**, que representan a los **objetos** (archivos, impresoras, bases de datos, etc.). En cada intersección (celda), se especifica el conjunto de operaciones que el sujeto tiene permitidas sobre ese objeto: lectura, escritura, ejecución, borrado, entre otras.

Ejemplo básico de una matriz de acceso:

	ArchivoA	ArchivoB	Impresora
Usuario1	Leer	Ninguno	Imprimir
Usuario2	Leer/Escribir	Leer	Ninguno

Aunque es un modelo muy útil, en sistemas con muchos usuarios y recursos, esta matriz puede volverse extremadamente grande e ineficiente. Por esta razón, los sistemas modernos suelen implementar formas derivadas de esta matriz:

- **Listas de Control de Acceso (ACLs)**: Se asocian a cada objeto del sistema, y contienen una lista de los sujetos con sus respectivos permisos. Es una forma eficiente de representar las columnas de la matriz.
- **Listas de Capacidades**: Se asocian a los sujetos del sistema y contienen las capacidades o privilegios que tienen sobre diversos objetos. Representan las filas de la matriz.
- **Bitmaps de acceso**: Representan los permisos mediante matrices binarias que facilitan el procesamiento automático.

Implementar correctamente matrices de acceso ayuda no solo a mejorar la seguridad, sino también a evitar errores administrativos, reducir el riesgo de privilegios excesivos y facilitar la auditoría del sistema.

6.1.4 Modelos formales de seguridad

Los modelos formales de seguridad son estructuras teóricas que permiten definir, analizar y verificar las políticas de seguridad en los sistemas computacionales. Estos modelos utilizan matemáticas y lógica para expresar cómo debe comportarse un sistema seguro, y se aplican principalmente en entornos donde la protección de la información es crítica, como en el sector militar, gubernamental o financiero.

Modelo Bell-LaPadula (BLP)
Diseñado en los años 70 para proteger la confidencialidad de la información en entornos militares, el modelo Bell-LaPadula se basa en el principio de que la información clasificada no debe ser accesible por usuarios con un nivel de autorización inferior. Se centra exclusivamente en la **confidencialidad**, y define dos reglas fundamentales:

- **No read up (NRU)**: Un sujeto no puede leer información de un nivel de clasificación superior al suyo.

- **No write down (NWD)**: Un sujeto no puede escribir información hacia un nivel de clasificación inferior.

Modelo Biba

A diferencia del modelo BLP, el modelo Biba se enfoca en garantizar la **integridad** de la información. Establece reglas que impiden que datos con poca integridad puedan contaminar datos de alta integridad. Las reglas básicas son:

- **No read down (NRD)**: Un usuario no puede leer datos de un nivel inferior de integridad.
- **No write up (NWU)**: Un usuario no puede escribir datos a un nivel superior de integridad.

Modelo Clark-Wilson

Este modelo está orientado al ámbito comercial y propone un enfoque práctico para mantener tanto la **integridad como la trazabilidad** de las operaciones. Introduce los conceptos de separación de funciones, verificación por parte de usuarios autorizados y el uso de transacciones bien definidas.

El uso de modelos formales permite a los diseñadores de sistemas establecer políticas sólidas desde la base del sistema operativo, y verificar mediante pruebas lógicas que los sistemas se comporten de forma segura bajo diversas condiciones.

6.2 Virtualización

La virtualización es una de las tecnologías más transformadoras en el ámbito de la informática moderna. Consiste en la creación de versiones virtuales de componentes físicos, como servidores, sistemas operativos, almacenamiento o redes, utilizando software especializado denominado **hipervisor**. Esto permite ejecutar múltiples entornos de computación independientes sobre una misma infraestructura física, maximizando el aprovechamiento de los recursos y proporcionando una gran flexibilidad.

Existen distintos tipos de virtualización:

- **Virtualización de servidores**: Permite dividir un servidor físico en múltiples servidores virtuales. Cada uno opera como una máquina independiente, con su propio sistema operativo, aplicaciones y configuración.
- **Virtualización de escritorios**: Utilizada para centralizar escritorios en un servidor y acceder a ellos desde distintos dispositivos remotos.
- **Virtualización de aplicaciones**: Las aplicaciones se ejecutan en un entorno virtual que es independiente del sistema operativo del usuario, facilitando la compatibilidad y el despliegue.

- **Virtualización de red y almacenamiento**: Crea representaciones virtuales de switches, routers o discos duros, facilitando su administración y escalabilidad.

Desde el punto de vista de la **seguridad**, la virtualización ofrece numerosos beneficios:

- **Aislamiento**: Cada máquina virtual está separada de las demás. Si una VM es comprometida, las demás no se ven afectadas.
- **Entornos de pruebas seguros**: Es posible probar software, configuraciones o incluso malware dentro de una VM sin poner en riesgo el sistema anfitrión.
- **Snapshots y restauración rápida**: Se pueden capturar estados del sistema y revertirlos en caso de incidentes de seguridad.
- **Mejor gestión de parches y políticas**: La virtualización permite actualizar sistemas de manera eficiente y aplicar políticas de seguridad homogéneas.

Sin embargo, esta tecnología también plantea **nuevos retos** de seguridad. Por ejemplo, si un atacante logra comprometer el hipervisor (lo que se conoce como "escape de VM"), podría obtener control total sobre todas las máquinas virtuales. Por ello, es indispensable implementar controles adicionales como firewalls virtuales, segmentación de red, monitoreo de tráfico y actualizaciones constantes del software de virtualización.

6.2.1 Emulación

La emulación es una técnica mediante la cual un sistema informático simula el comportamiento de otro sistema distinto, tanto en hardware como en software. A diferencia de la virtualización, que ejecuta múltiples entornos sobre una arquitectura similar, la emulación permite que un sistema funcione como si tuviera una arquitectura de hardware completamente diferente.

Por ejemplo, es posible emular una consola de videojuegos antigua en una computadora moderna, o ejecutar software diseñado para un procesador ARM en una arquitectura x86. Esto se logra mediante un programa denominado **emulador**, que traduce las instrucciones del sistema original a instrucciones entendibles por el sistema anfitrión.

Ventajas de la emulación:

- **Compatibilidad completa:** Puede ejecutar software de plataformas obsoletas o incompatibles.

- **Preservación digital:** Permite conservar y ejecutar sistemas antiguos para investigación, historia o nostalgia.
- **Pruebas multiplataforma:** Los desarrolladores pueden probar aplicaciones en diferentes arquitecturas sin necesidad de tener el hardware real.

Desventajas:

- **Desempeño reducido:** Al tener que traducir cada instrucción, la ejecución suele ser más lenta que en una máquina física o virtual.
- **Complejidad técnica:** Emular sistemas complejos requiere una programación cuidadosa y mucha potencia de cómputo.

La emulación es útil en contextos específicos donde la virtualización no puede aplicarse directamente, especialmente en casos donde el software o sistema operativo requiere una arquitectura específica para funcionar.

6.2.2 Virtualización asistida por hardware

La **virtualización asistida por hardware** es una mejora de la virtualización tradicional, en la que el procesador físico (CPU) proporciona soporte específico para facilitar y optimizar la ejecución de máquinas virtuales. Este tipo de virtualización reduce la carga del software hipervisor, permitiendo una ejecución más eficiente, estable y rápida de entornos virtuales.

Las tecnologías de virtualización asistida por hardware más conocidas son:

- **Intel VT-x (Virtualization Technology)**
- **AMD-V (AMD Virtualization)**

Estas extensiones permiten que las instrucciones de la CPU que normalmente se reservarían para el sistema operativo anfitrión puedan ser utilizadas directamente por las máquinas virtuales. Esto reduce la necesidad de técnicas complejas de traducción o intervención del hipervisor.

Beneficios clave:

- **Mejor rendimiento:** Las operaciones virtuales se ejecutan más cerca del hardware real.
- **Mayor aislamiento:** Las máquinas virtuales son menos propensas a interferirse entre sí.
- **Soporte de 64 bits y multiprocesamiento:** Las VMs pueden aprovechar múltiples núcleos y ejecutar sistemas operativos modernos.

Gracias a estas tecnologías, los hipervisores modernos como VMware, Hyper-V y VirtualBox pueden crear entornos virtuales mucho más cercanos al rendimiento de una máquina física, mejorando la experiencia de uso y facilitando la consolidación de servidores.

6.2.3 Paravirtualización

La **paravirtualización** es un enfoque de virtualización en el que el sistema operativo invitado (guest OS) es modificado para que sea consciente de que está corriendo en un entorno virtual. A diferencia de la virtualización completa, donde el sistema operativo funciona como si estuviera en una máquina real, en la paravirtualización se adaptan ciertas funciones para que interactúen directamente con el hipervisor.

En lugar de simular todo el hardware, como ocurre en la emulación o en la virtualización completa, el sistema operativo en una paravirtualización realiza llamadas especiales (llamadas hipervisor o *hypercalls*) que son más eficientes.

Ventajas:

- **Rendimiento mejorado:** Al evitar la emulación completa del hardware, se reduce la sobrecarga.
- **Interacción más directa con el hipervisor:** Esto permite optimizaciones adicionales en la gestión de recursos.

Desventajas:

- **Requiere modificar el sistema operativo invitado:** No todos los sistemas están preparados para paravirtualizarse.
- **Menor compatibilidad:** No se puede usar con sistemas operativos propietarios que no permitan modificaciones.

Esta técnica fue utilizada ampliamente en tecnologías como Xen, especialmente antes de que existiera la virtualización asistida por hardware. Hoy en día, sigue siendo útil para entornos que requieren alto rendimiento y donde es posible modificar el sistema operativo huésped.

6.2.4 Contenedores

Los **contenedores** son una forma moderna de virtualización a nivel de sistema operativo. A diferencia de las máquinas virtuales tradicionales, los contenedores no requieren un sistema operativo completo por cada instancia. En cambio, comparten el kernel del sistema operativo anfitrión, pero mantienen sus propias bibliotecas, dependencias y entorno de ejecución.

La tecnología de contenedores más popular es **Docker**, aunque existen otras como **LXC**, **Podman** o **Kubernetes** (para orquestación de contenedores).

Características clave:

- **Ligereza:** Ocupan mucho menos espacio que una VM y se inician en segundos.
- **Portabilidad:** Los contenedores pueden ejecutarse en cualquier entorno que tenga el mismo motor de contenedores, sin importar el sistema operativo subyacente.
- **Aislamiento:** Aunque comparten el mismo kernel, los contenedores están aislados unos de otros mediante espacios de nombres y controladores de recursos (namespaces y cgroups).

Ventajas:

- Perfectos para desarrollar, probar y desplegar aplicaciones rápidamente.
- Escalabilidad sencilla mediante sistemas como Kubernetes.
- Ideales para arquitecturas de microservicios.

Limitaciones:

- Menor aislamiento comparado con las máquinas virtuales completas.
- No son adecuados para ejecutar sistemas operativos completamente distintos del host (por ejemplo, no se puede correr Windows en un contenedor Linux).

En resumen, los contenedores representan una evolución natural de la virtualización, ofreciendo un equilibrio entre rendimiento, portabilidad y facilidad de gestión. Se han convertido en un pilar fundamental del desarrollo moderno de software, especialmente en entornos DevOps y de computación en la nube.

CONCLUSIÓN GENERAL

A lo largo del estudio de los temas relacionados con la seguridad informática y la virtualización, se ha evidenciado la profundidad, complejidad y relevancia de estos conceptos en el diseño, implementación y protección de sistemas computacionales modernos. En primer lugar, el **control de acceso a los recursos** representa un pilar fundamental para

asegurar que los datos y servicios solo sean utilizados por los usuarios autorizados, lo que protege la confidencialidad, integridad y disponibilidad de la información. Este control se implementa mediante diversos modelos, como el acceso discrecional, obligatorio o basado en roles, los cuales permiten establecer políticas de seguridad adaptadas a distintos entornos.

El análisis de la **implementación de matrices de acceso** permitió comprender la forma estructurada en que se asignan y gestionan los permisos dentro de los sistemas. Estas matrices, y sus variantes como las listas de control de acceso y las capacidades, proporcionan una base lógica para administrar quién tiene derecho a interactuar con qué recursos, reforzando así el marco de control. Por su parte, los **modelos formales de seguridad**, como Bell-LaPadula, Biba y Clark-Wilson, introducen un enfoque teórico riguroso para validar la seguridad de los sistemas mediante principios matemáticos y lógicos. Estos modelos resultan especialmente útiles en contextos donde las exigencias de seguridad son críticas, como en la administración pública, la defensa nacional o la industria financiera.

En el ámbito de la infraestructura, el estudio de la **virtualización** reveló su importancia como tecnología habilitadora para la optimización de recursos, la mejora en la gestión de sistemas y la consolidación de servicios. La virtualización permite ejecutar múltiples entornos de trabajo sobre una sola plataforma física, incrementando la eficiencia operativa y facilitando la administración y recuperación ante fallos. Dentro de este marco, exploramos varios tipos de virtualización avanzada, como la **emulación**, útil para ejecutar sistemas sobre arquitecturas diferentes; la **virtualización asistida por hardware**, que mejora el rendimiento gracias al soporte del procesador; la **paravirtualización**, que optimiza el rendimiento mediante modificaciones al sistema operativo huésped; y los **contenedores**, que han revolucionado el desarrollo de software gracias a su ligereza, portabilidad y eficiencia.

En conjunto, estos temas muestran cómo la seguridad y la virtualización son áreas interrelacionadas que deben considerarse de manera integral al diseñar sistemas confiables y robustos. El dominio de estos conceptos no solo es esencial para los profesionales en informática, sino también para cualquier organización que desee proteger su información, optimizar sus recursos y adaptarse a un entorno tecnológico cada vez más dinámico, distribuido y exigente.

BIBLIOGRAFÍA

- Stallings, W. (2020). *Computer Security: Principles and Practice* (4th ed.). Pearson Education.

Referencia clave para temas de seguridad, control de acceso, modelos formales y virtualización.

- Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). *Operating System Concepts* (10th ed.). Wiley.

Fuente fundamental para comprender el funcionamiento de sistemas operativos y mecanismos de virtualización.

- Bishop, M. (2003). *Computer Security: Art and Science*. Addison-Wesley.

Texto clásico y profundo sobre modelos formales de seguridad.

- Tanenbaum, A. S., & Bos, H. (2015). *Modern Operating Systems* (4th ed.). Pearson.

Excelente fuente para virtualización, contenedores y emulación.

- Hwang, K., Fox, G. C., & Dongarra, J. J. (2012). *Distributed and Cloud Computing: From Parallel Processing to the Internet of Things*. Morgan Kaufmann.

Contexto moderno sobre virtualización, contenedores y su uso en la nube.

- Docker Inc. (2024). *Docker Documentation*. Recuperado de https://docs.docker.com

Documentación oficial de Docker para contenedores y orquestación.

- Xen Project. (2024). *Xen Hypervisor Documentation*. Recuperado de https://xenproject.org

Fuente específica sobre paravirtualización y uso de hipervisores.

- Intel Corporation. (2023). *Intel® Virtualization Technology (VT-x) Documentation*. Recuperado de https://www.intel.com

Información oficial sobre virtualización asistida por hardware.
