

Franco Rayas Ángel Damián
• Gómez López Miguel Ángel
• López Coria Axel Yahir
• Vasco Giraldo Juan Esteba

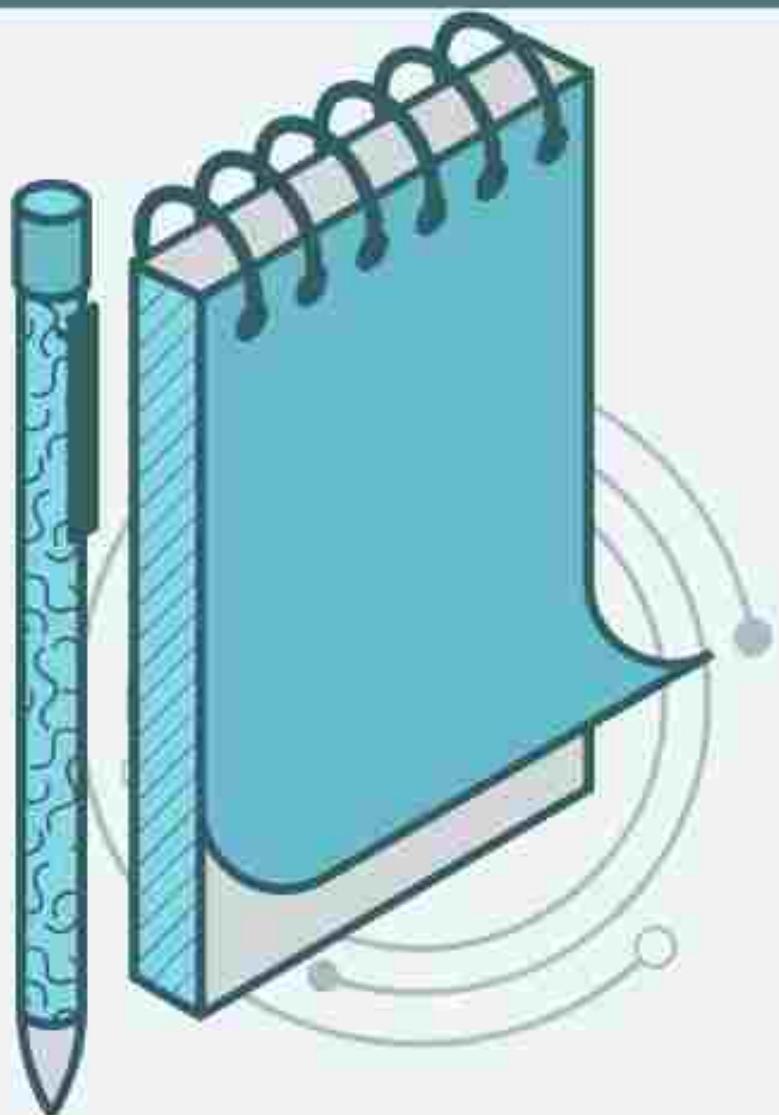
SEGURIDAD Y VIRTUALIZACIÓN

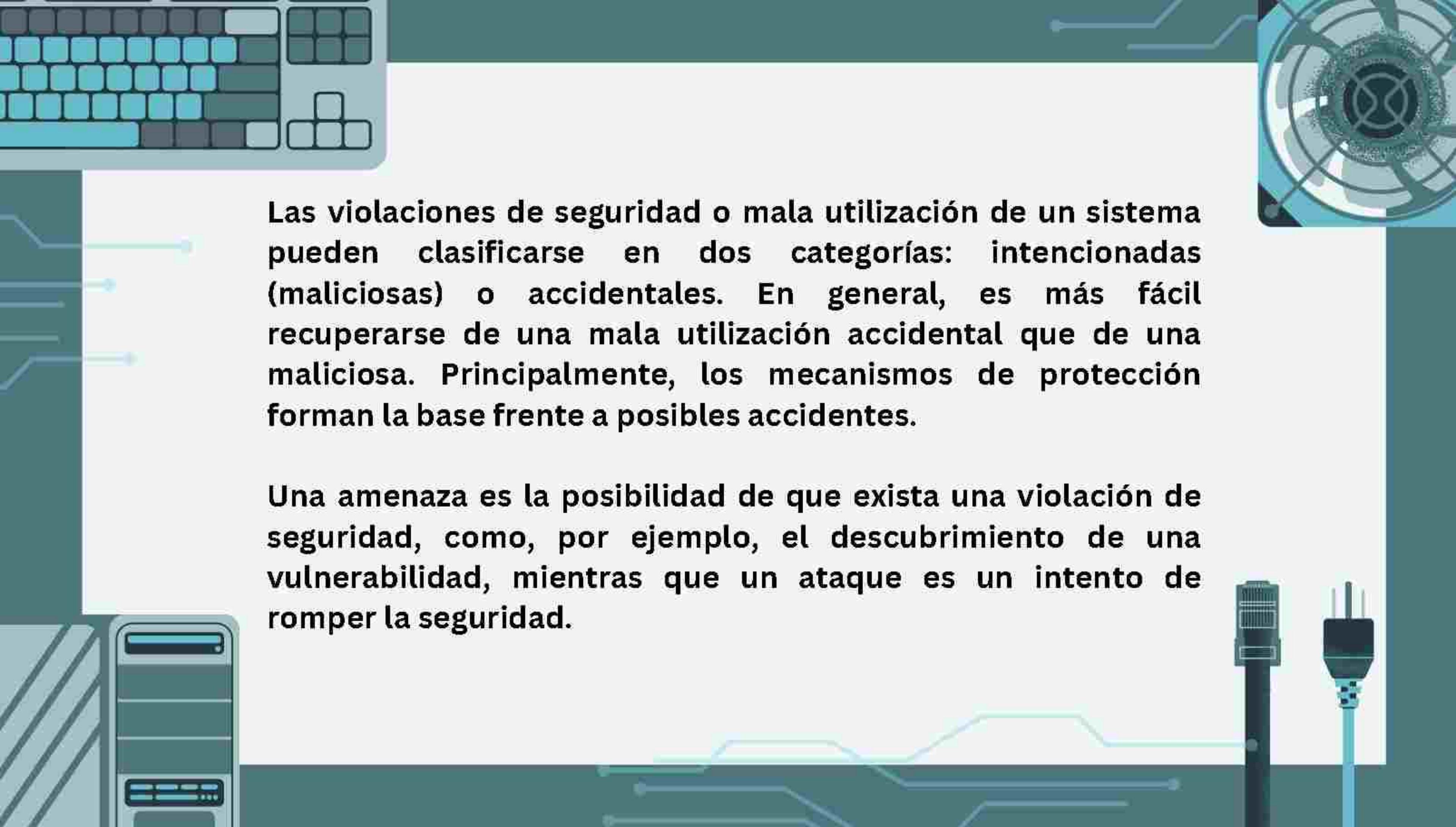
ÍNDICE

1. El problema de la seguridad
2. Amenazas relacionadas con los programas
 3. Amenazas del sistema y la red
4. La criptografía como herramienta de seguridad
 5. Autenticación de usuario
 6. Objetivos de la protección
 7. Principios de la protección
 8. Dominio de protección
 9. Matriz de acceso
10. Implementación de una matriz de acceso
 11. Control de acceso
12. Revocación de los derechos de acceso
13. Sistemas basados en capacidades
14. Virtualización

EL PROBLEMA DE LA SEGURIDAD

En muchas aplicaciones, debe dedicarse un esfuerzo considerable para poder garantizar la seguridad del sistema informático. Los sistemas que contengan datos personales, como direcciones, cuentas bancarias, o cualesquiera otros datos sensibles, son algunos de los principales blancos de los criminales.





Las violaciones de seguridad o mala utilización de un sistema pueden clasificarse en dos categorías: intencionadas (maliciosas) o accidentales. En general, es más fácil recuperarse de una mala utilización accidental que de una maliciosa. Principalmente, los mecanismos de protección forman la base frente a posibles accidentes.

Una amenaza es la posibilidad de que exista una violación de seguridad, como, por ejemplo, el descubrimiento de una vulnerabilidad, mientras que un ataque es un intento de romper la seguridad.

- **Ruptura de confidencialidad:** Este tipo de violación implica la lectura no autorizada de determinados datos o el robo de información, por lo general, el objetivo de los atacantes es la ruptura de la confidencialidad, lo cual implica la captura de datos secretos, como información de tarjetas de crédito, entre otras cosas.
- **Ruptura de la integridad:** Este tipo de ataque implica la modificación no autorizada de los datos, este tipo de ataques puede que se atribuya la responsabilidad a una persona a alguien que es inocente.
- **Ruptura de la disponibilidad:** Esta violación de seguridad implica la destrucción no autorizada de los datos. Algunos atacantes prefieren hacer daño y hacerse de un renombre en lugar de buscar beneficios financieros.

- **Robo de servicios:** Este tipo de violación implica el uso no autorizado de recursos, como, por ejemplo, la instalación de algún programa malintencionado.
- **Denegación de servicios:** Implica impedir el uso legítimo del sistema. Los ataques de denegación de servicios (DOS, denial-of-service) son en ocasiones accidentes.

Los atacantes usan diferentes métodos para poder romper la seguridad de un sistema. Uno de los más comunes es la mascarada, en la que un participante en una comunicación pretende ser otra persona o un host.



Otro de los ataques más comunes consiste en reproducir un intercambio de datos previamente capturado, los ataques de reproducción consisten en la repetición maliciosa o fraudulenta de una transmisión de datos no válida. Nuevamente el objetivo es obtener ciertos privilegios.

Otro tipo de ataque es el ataque por interposición (man-in-the-middle), en el cual el atacante se introduce dentro del flujo de datos de la comunicación, haciéndose pasar por el emisor a ojos del receptor y viceversa, el ejemplo más ilustrativo de este tipo de ataque es el secuestro de información (hijacking), en el que se intercepta una comunicación activa.

NIVELES



FÍSICO

Es el nodo o nodos que contengan los sistemas informáticos deben dotarse de medidas de seguridad físicas posibles frente a posibles intrusiones armadas, así como de dotar de seguridad a las habitaciones donde haya elementos del sistema.



HUMANO

La autorización de los servicios pueden llevarse a cabo con cuidado, con el fin de garantizar que solo los usuarios apropiados tengan acceso al sistema, sin embargo, estos mismos usuarios apropiados pueden verse tentados a permitir que otros usuarios tengan acceso (por ejemplo: un soborno), además de que también pueden verse engañados para permitir el acceso mediante técnicas de ingeniería social.



SISTEMA OPERATIVO

El sistema debe autoprotegerse frente a posibles fallos de seguridad accidentales o premeditados. Un proceso que esté fuera de control podría llegar a construir un ataque accidental de denegación de servicios.



RED

Son muchos los datos en los modernos sistemas informáticos que viajan a través de líneas privadas, de internet, de conexiones inalámbricas o de línea telefónica, la interceptación de esta información puede resultar tan dañina como el acceso a una computadora, por lo que la interrupción de esta comunicación puede construir un ataque remoto.

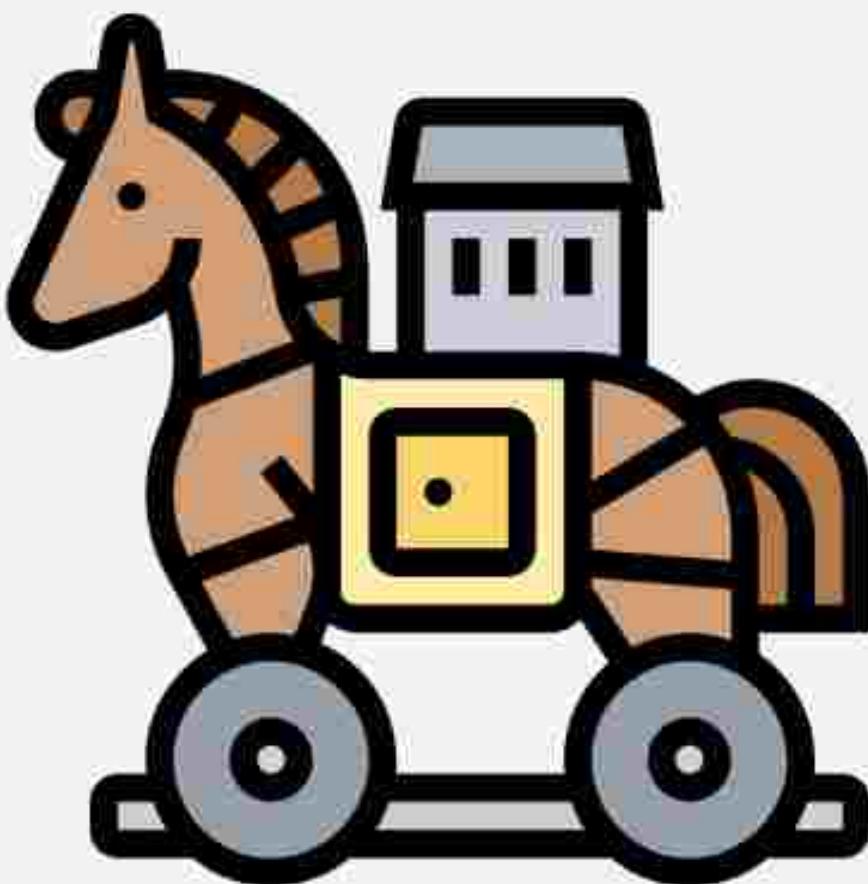
Al mismo tiempo, los sistemas deben proporcionar mecanismos de protección para permitir la implementación de las características de seguridad. Sin la capacidad de autorizar a los usuarios y procesos, de controlar su acceso y registrar sus actividades, prácticamente sería imposible que un sistema operativo implementara medidas de seguridad o se ejecutara de manera segura.

Desafortunadamente, casi ninguno de los aspectos relativos a la seguridad resulta sencillo, a medida que los intrusos aprendan a aprovechar las vulnerabilidades de los sistemas de seguridad, se crean e implantan las correspondientes contramedidas, lo cual hace que con el paso del tiempo los ataques sean más sofisticados.



Algunos sistemas tienen mecanismos para permitir que programas escritos por unos usuarios sean ejecutados por otros usuarios, si estos programas en un dominio que proporcione los derechos de acceso al usuario ejecutante, los usuarios pueden aprovechar inapropiadamente estos permisos.

Un segmento de código que use inapropiadamente su entorno de denominado caballo de Troya. Las rutas de búsqueda de gran longitud, como las que aparecen en los sistemas UNIX, agravan estos problemas. La ruta de búsqueda enumera el conjunto de directorio en el que hay que buscar cuando se proporciona un nombre de programa ambiguo. Lo que se hace es buscar en esa ruta un archivo con dicho nombre y ejecutar ese archivo. Todos los archivos de dicho directorio deben de ser seguros, de lo contrario es ahí cuando puede introducirse un caballo de Troya.



CABALLO DE TROYA

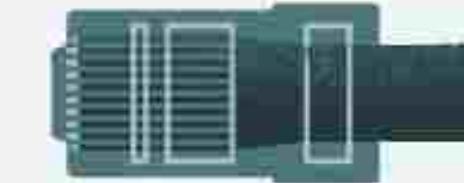
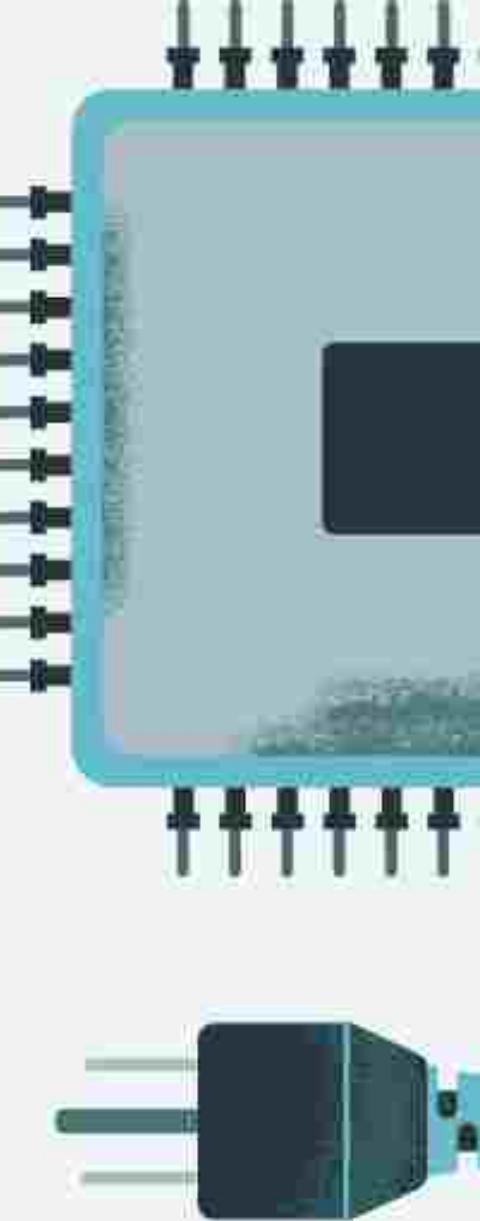
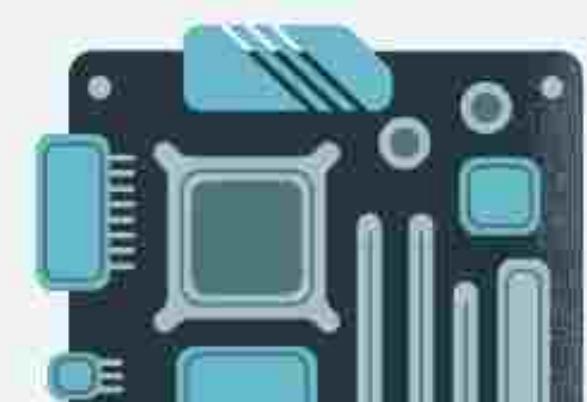
Una de las variantes más comunes del caballo de Troya es un emulador de inicio de sesión. Un usuario normal lo que tratará de hacer es iniciar sesión en un terminal y verá que ha escrito mal su contraseña, después de esto vuelve a iniciar sesión y esta vez lo hace de forma correcta. Lo que en realidad sucede en estos casos es que su clave de autenticación y contraseña han sido robadas por dicho emulador.

Otras de las variantes del caballo de Troya es el spyware. Este tipo de programas acompañan en ocasiones a ciertos programas que el usuario haya decidido instalar. En la mayoría de los casos se incluyen programas de tipo freeware o shareware, pero en otros casos son softwares de tipo comercial. El objetivo del spyware es descargar anuncios para mostrar el contenido al usuario, crear ventanas de explorador emergentes cuando se visiten ciertos sitios o capturar información del sistema del usuario y enviarla a un sitio central.

PUERTA TRASERA

El diseñador de un programa o un sistema puede dejar detrás suyo un agujero en el software que el solo sea capaz de utilizar, este tipo de brecha o de puerta trasera es lo que representa el problema. Se han detectado casos de programadores que fueron condenados por estafar a los bancos incluyendo los errores de redondeo en su código y haciendo que estas fracciones de céntimos fueran abanadas en sus cuentas, pues estos abonos si se hacen en una gran cantidad pueden llegar a acumular grandes cantidades de dinero.

Las puertas traseras plantean un difícil problema, pues al detectarlas se tiene que analizar todo el código fuente de los componentes del sistema, puesto que los softwares pueden estar compuesto de millones de líneas de código.



BOMBA LÓGICA

Si consideramos un programa que es capaz de iniciar un incidente de seguridad solo cuando se dan determinadas circunstancias, es difícil detectar, en condiciones normales de operación, pues no existirá ningún agujero de seguridad. Este escenario recibe el nombre de bomba lógica.

Por ejemplo, un programador puede hacer un código para determinar si es que la empresa aún sigue operando, en caso de que esta comprobación falle, podría crearse un acceso remoto o introducir un pequeño código que pueda causar un gran daño.



DESBORDAMIENTO DE PILA Y BÚFER

Este tipo de ataque es la forma más común que tiene un atacante externo al sistema, a través de una conexión de red o acceso telefónico, obtenga acceso no autorizado al sistema operativo. Por otro lado, los usuarios autorizados pueden utilizar este tipo de ataque para aumentar sus privilegios.

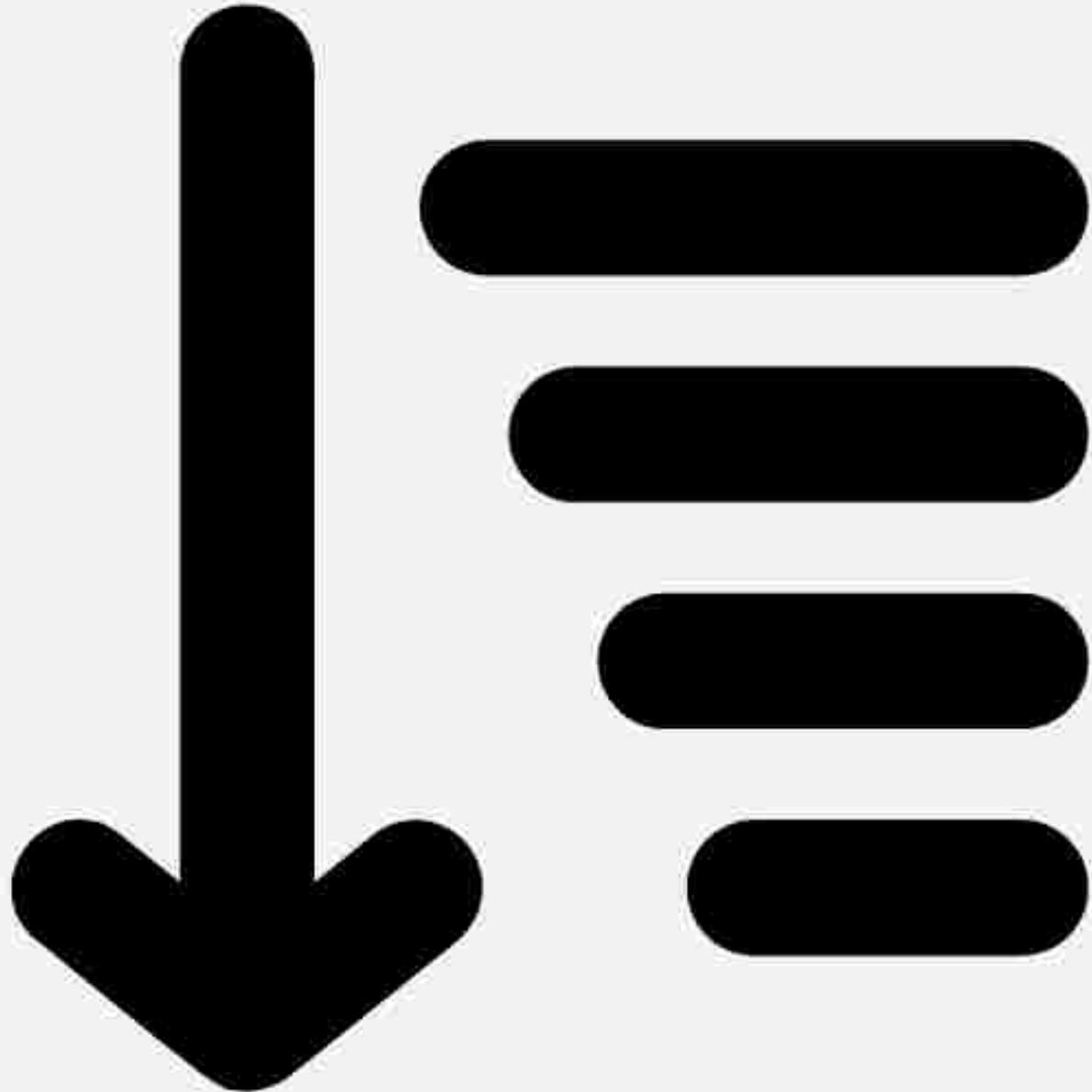
Utilizando un método de prueba y error, o analizando el código, el atacante puede encontrar una vulnerabilidad y escribe un programa que hace lo siguiente:



- Desbordamiento de un campo de entrada, un argumento de una línea de comandos o un búfer de entrada hasta escribir en la zona correspondiente a la pila.
- Sobreescribir la dirección actual de retorno de la pila, sustituyéndola por la dirección de los códigos de ataque cargados.
- Escribir un fragmento de código, en el siguiente espacio de la pila, que incluye ahí los comandos que el atacante quiera ejecutar, como, por ejemplo, un código en Shell.

Cuando se invoca una función en una arquitectura informática típica, las variables definidas localmente a la función (conocidas como variables automáticas), los parámetros pasados a la función y a la dirección a la que volverá el control cuando la función termine se almacenan en una entrada de pila.

Después se encuentra el puntero a la entrada a la pila, que es la dirección del comienzo de la entrada de la pila. Finalmente se tiene la dirección de retorno, que determina a donde hay que devolver el control una vez terminada la función.

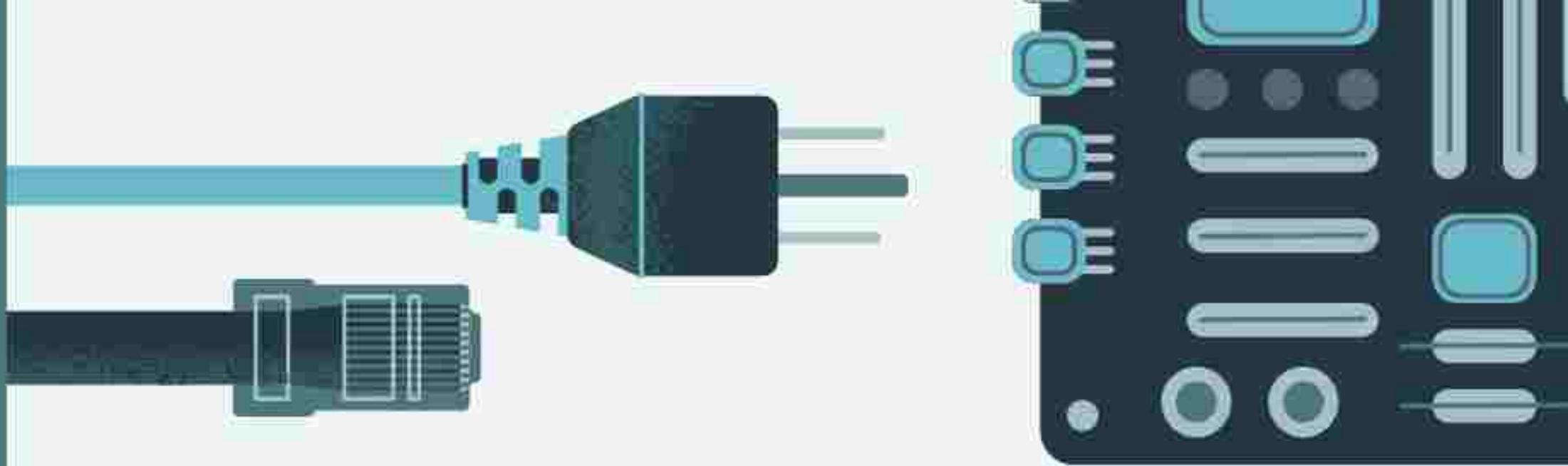
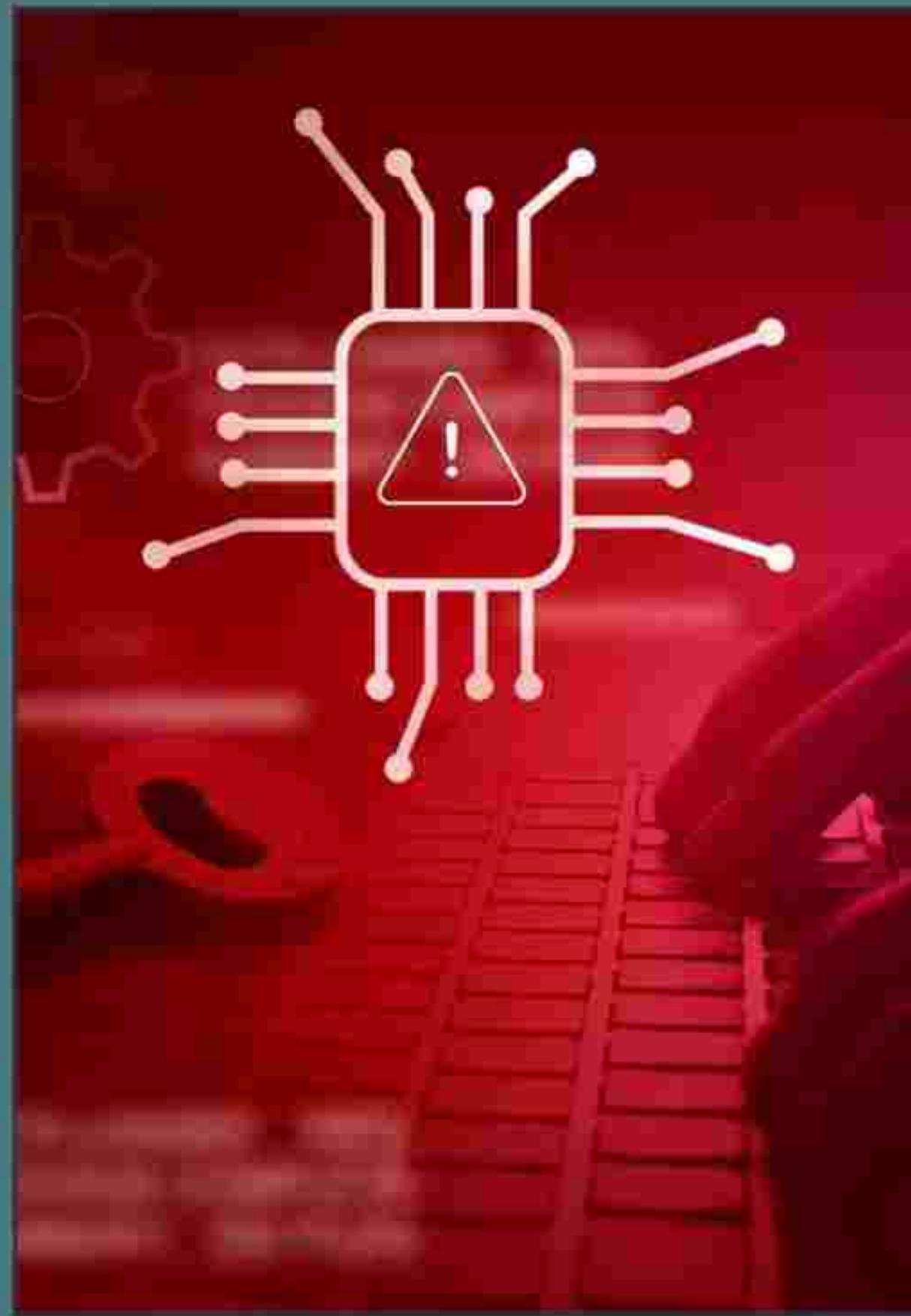


Hay muchas formas de tratar de aprovechar los problemas potenciales de desbordamiento de búfer, sin embargo, para poder lanzar ataques de seguridad, no hace falta ser un programador destacado, un informático podría detectar cual es el error existente en un cierto programa y poder escribir un módulo de ataque.

Cualquiera que tenga conocimientos de informática rudimentarios y que se haga con ese código de ataque podrían tratar de lanzar un ataque contra determinados sistemas objetivos.

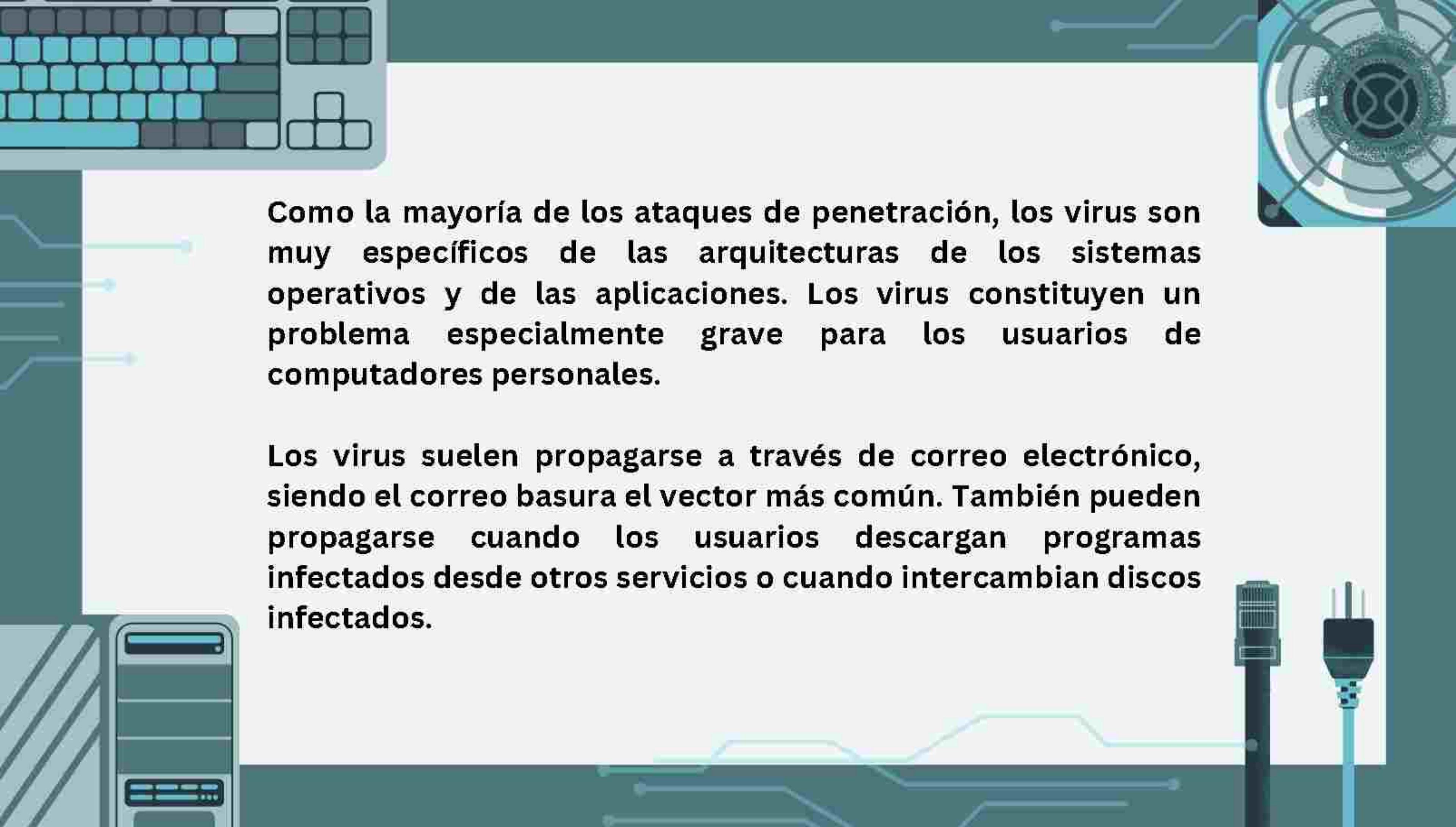
Una solución a este problema es que la CPU disponga de una característica que no permita la ejecución de código contenido en la sección de pila de la memoria.





VIRUS

Otro tipo de amenaza en forma de programa son los virus. Los virus son auto replicantes y están diseñados para infectar otros programas. Pueden causar estragos en un sistema modificando o destruyendo archivos y provocando funcionamientos inadecuados de los programas y fallos catastróficos al sistema.



Como la mayoría de los ataques de penetración, los virus son muy específicos de las arquitecturas de los sistemas operativos y de las aplicaciones. Los virus constituyen un problema especialmente grave para los usuarios de computadores personales.

Los virus suelen propagarse a través de correo electrónico, siendo el correo basura el vector más común. También pueden propagarse cuando los usuarios descargan programas infectados desde otros servicios o cuando intercambian discos infectados.

ARCHIVO

Un virus de este tipo infecta al sistema insertándose en un archivo y modificando el inicio del programa para que la ejecución salte al código del virus.

ARRANQUE

Infecta al sector de arranque del sistema, ejecutándose cada vez que el sistema arranca y antes de que se cargue el sistema operativo, el virus busca otros soportes de arranque para infectarlos.

MACRO

La mayoría de los virus están escritos en lenguajes de bajo nivel, como en ensamblador o en C, pero estos tipos de virus están escritos en lenguajes de alto nivel.

CÓDIGO FUENTE

Este tipo de virus busca un código fuente y lo modifica para poder incluir al virus y ayudar a su distribución.

POLIMÓRFICO

Este tipo de virus cambia cada vez que se instala con el fin de que el software antivirus no puedan detectarlo.



CATEGORIAS DE UN VIRUS

CIFRADO

Incluyen un código de descripción junto con el virus cifrado, para así poder evitar su detección, primero se descifra y luego se ejecuta.



ENCUBIERTO

Trata de evitar la detección modificando partes del sistema operativo que puedan ser usadas para poder detectarlo.



TÚNEL

Evita la detección por parte de programas antivirus instalándose a sí mismo en la rutina de tratamiento de interrupciones.



MULTIPARTE

Son capaces de infectar múltiples partes de un sistema, incluyendo los sectores de arranque, la memoria y los archivos.



ACORAZADO

Están codificados de tal manera que resultan difíciles de desentrañar y de comprender por parte de los investigadores que desarrollan programas antivirus. También pueden estar comprimidos para evitar la detección y desinfección.



AMENAZAS DEL SISTEMA Y LA RED

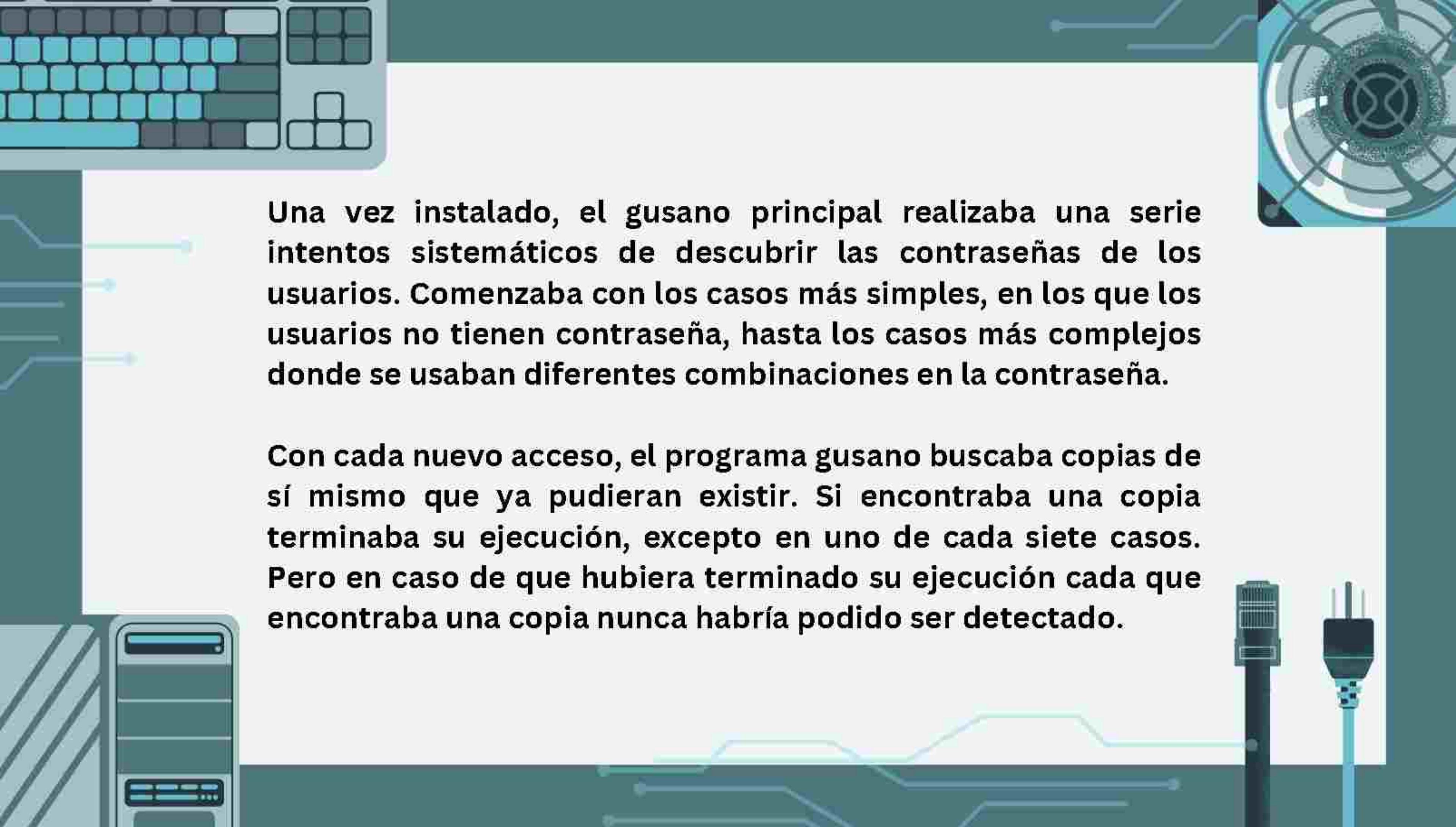
Gusanos

Un gusano es un proceso que utiliza un mecanismo de reproducción para afectar al rendimiento del sistema. El gusano crea copias de sí mismo, utilizando recursos del sistema y en ocasiones impidiendo operar a los demás procesos. En las redes informáticas, los gusanos son potencialmente peligrosos, pues pueden replicarse de un sistema a otro provocando el colapso de una red completa.



En noviembre de 1988, Robert Tappan Morris Jr., lanzó un programa gusano en una o más maquinas host conectadas en internet, teniendo por objetivo las estaciones de trabajo de Sun Microsystems y a las computadoras VAX.

El gusano estaba compuesto de dos programas. Un vector y el programa principal, el vector estaba compuesto de 99 líneas de código C y se ejecutaba en cada maquina a la que accedía, una vez establecido en el sistema, el vector se conectaba a la maquina donde se había originado y cargaba una copia del gusano original.



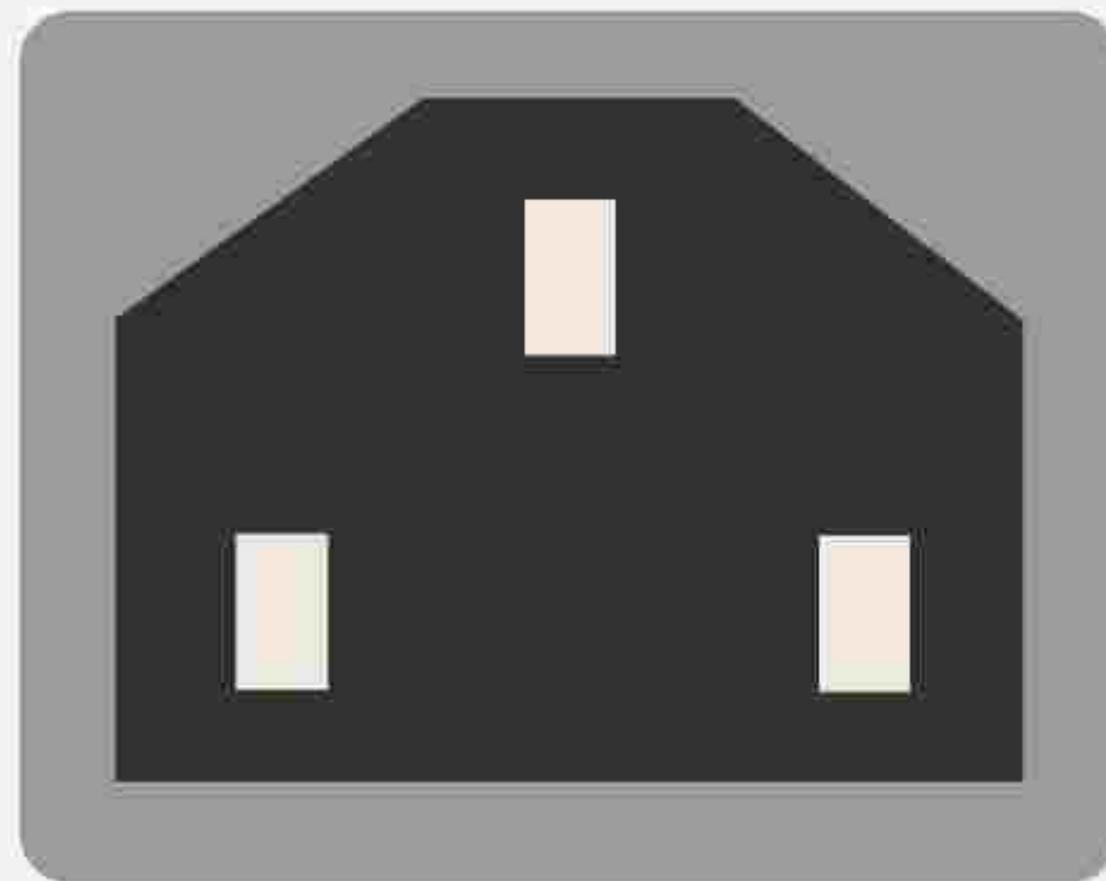
Una vez instalado, el gusano principal realizaba una serie intentos sistemáticos de descubrir las contraseñas de los usuarios. Comenzaba con los casos más simples, en los que los usuarios no tienen contraseña, hasta los casos más complejos donde se usaban diferentes combinaciones en la contraseña.

Con cada nuevo acceso, el programa gusano buscaba copias de sí mismo que ya pudieran existir. Si encontraba una copia terminaba su ejecución, excepto en uno de cada siete casos. Pero en caso de que hubiera terminado su ejecución cada que encontraba una copia nunca habría podido ser detectado.

ESCANEO DE PUERTOS

El escaneo de puertos en sí no es un ataque, sino una forma en la que los atacantes detectan las vulnerabilidades del sistema que puedan ser atacadas. El escaneo de puertos se realiza normalmente de forma automatizada, lo que implica usar una herramienta que trate de conectar una conexión TCP/IP a un puerto o rango de puertos en específico.

Con frecuencia, los errores encontrados son desbordamientos de búfer, que permiten la creación de un Shell de comandos privilegiada en el sistema.



Por suerte, no existe una herramienta que haga dicha función, sin embargo, si existen programas o herramientas que hagan un subconjunto de esas funciones.

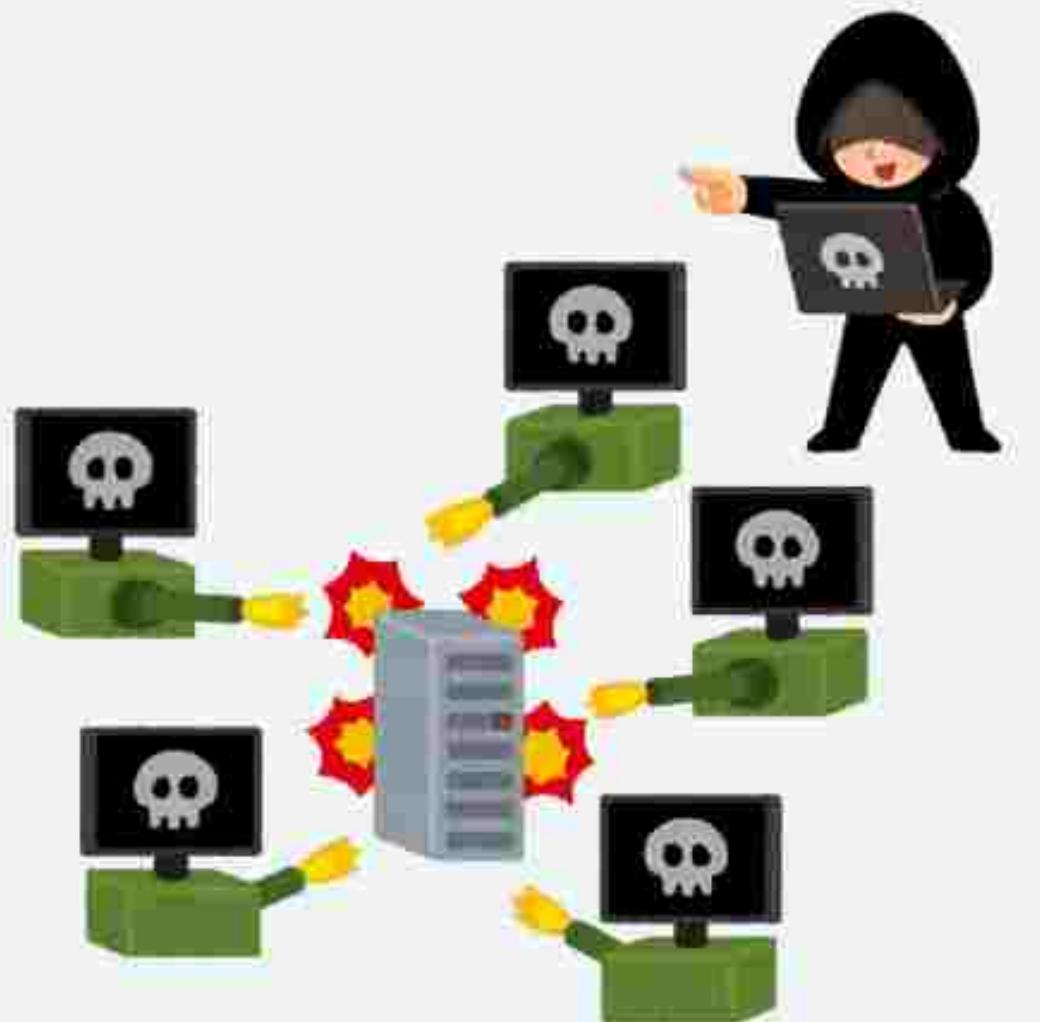
Cuando se le apunta a un puerto objetivo, determina qué servicios se están ejecutando, incluyendo los nombres y versiones de las aplicaciones. Puede determinar el sistema operativo host y también pueden proporcionar información acerca de las defensas.

Puesto que el escaneo de puertos si es detectable, se suelen realizar desde sistemas zombi. Dichos sistemas son maquinas independientes y previamente comprometidas que están prestando un servicio normal a sus propietarios y al mismo tiempo son utilizadas para propósitos inconfesables.

Los zombis hacen que resulte complicado el perseguir a los atacantes informáticos, ya que resulta complicado determinar el origen del ataque y la persona que lo ha iniciado. Esta es una de las razones por la cual también se aconseja dotar de seguridad a los sistemas “no importantes” y no solo a los sistemas que contengan información o servicios “valiosos”.



Los ataques DOS están dirigidos no a obtener información a o robar recursos, sino a impedir el uso legítimo de un sistema o funcionalidad. La mayoría de los ataques de denegación afectan a sistemas en los que el atacante no ha penetrado. De hecho, lanzar un ataque que impida el uso legítimo de un sistema resulta más sencillo que irrumpir en una máquina o instalación.



DENEGACIÓN DE SERVICIO

Este tipo de ataques se realizan generalmente a través de la red. Se les puede clasificar en dos categorías. El primer caso es el de los ataques que consumen tantos recursos de la máquina atacada que no puede realizarse ningún trabajo útil en ella. El segundo caso de ataque implica hacer caer la red o la instalación, los más comunes son los sitios web.

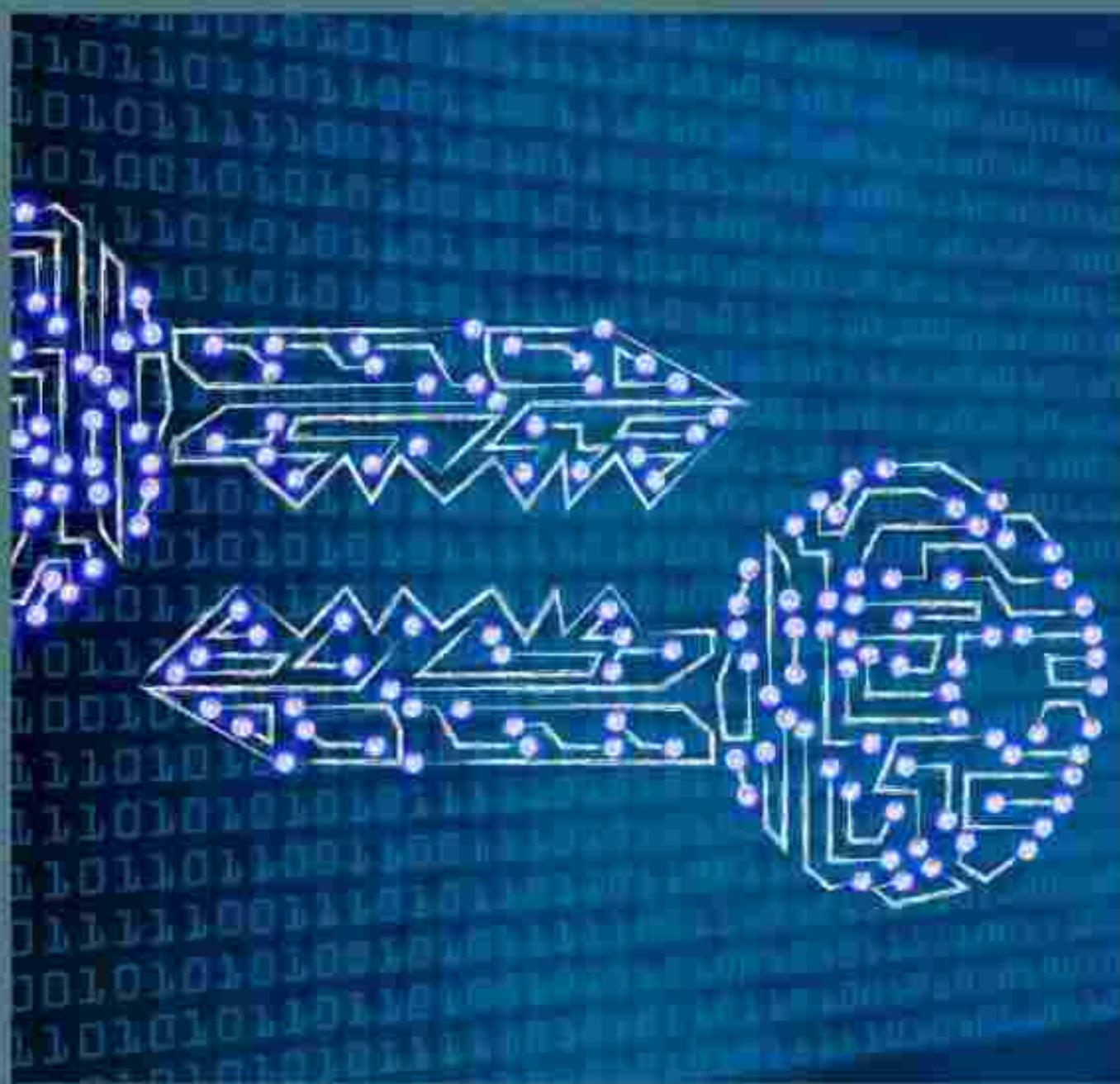
Por lo general, es imposible prevenir los ataques de denegación de servicio. Los ataques utilizan los mismos mecanismos que la operación normal. Todavía más difíciles de prevenir y de solucionar son los ataques distribuidos de denegación de servicio (DDOS). Estos ataques se inicián desde múltiples sitios a la vez, dirigidos hacia un objetivo en común, normalmente por programas zombis.

Un sitio ni siquiera es consciente de que está siendo atacado, pues resulta complicado determinar si la ralentización de un sistema se debe a un pico de utilización del mismo o a un ataque.



LA CRIPTOGRAFÍA COMO HERRAMIENTA DE SEGURIDAD

- La herramienta de carácter más general que está a disposición de los usuarios y de los diseñadores de sistemas es la criptografía.
- La criptografía se utiliza para restringir los emisores y receptores potenciales de un mensaje. Se basa en claves, que se distribuyen selectivamente a las computadoras de una red y se utilizan para procesar mensajes.
- Comúnmente, se utilizan las direcciones de red para inferir los emisores y receptores potenciales de los mensajes que circulan por la red.
-



CIFRADO



El cifrado es un medio de restringir los posibles receptores de un mensaje. Un algoritmo de cifrado permite al emisor de un mensaje garantizar que sólo pueda leer el mensaje una computadora que posea cierta clave. Un algoritmo de cifrado consta de los siguientes componentes: Un conjunto de claves, un conjunto de mensajes y un conjunto de mensajes de texto cifrado.

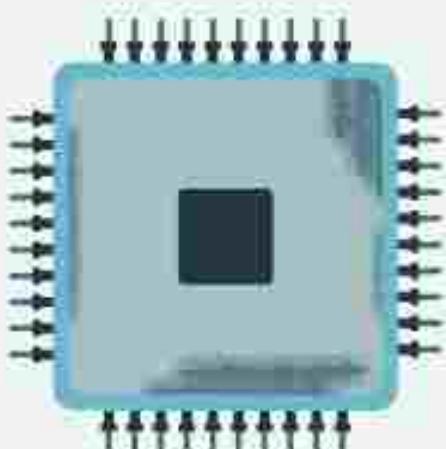
Así, una computadora que posea la clave puede descifrar los mensajes de texto cifrado para obtener los mensajes de texto en claro que se usaron para producirlos, pero una computadora que no lo posea, no puede descifrar esos mensajes cifrados.

CIFRADO SIMÉTRICO

En este algoritmo se utiliza la misma clave para cifrar y para descifrar. El más utilizado en los Estados Unidos ha sido el estándar DES, el cual funciona tomando un valor de 64 bits y una clave de 56 bits, realizando una serie de transformaciones. Estas transformaciones están basadas en operaciones de sustitución y permutación.

Algunas de las transformaciones son de las denominadas transformaciones de caja negra, en el sentido de que sus algoritmos están ocultos. Si se utiliza la misma clave para cifrar una gran cantidad de datos, esa clave comienza a ser vulnerable a los ataques.

RC4 es el algoritmo de cifrado de flujo más común. Un algoritmo de cifrado de flujo está diseñado para cifrar y descifrar un flujo de bytes o bits, en lugar de un bloque.



CIFRADO ASIMÉTRICO

En un algoritmo de cifrado asimétrico, las claves de cifrado y descifrado son distintas. El algoritmo RSA de cifrado es un algoritmo de cifrado de bloque de clave pública y es el algoritmo asimétrico más ampliamente utilizado.

Sin embargo, los algoritmos asimétricos basados en curvas elípticas están ganando cada vez más terreno, porque la longitud de clave de dichos algoritmos puede ser más corto para un grado determinado de fortaleza criptográfica.

El uso de un mecanismo de cifrado asimétrico comienza con la publicación de la clave pública del destino. Para la comunicación bidireccional, el origen debe también publicar su clave pública. La criptografía asimétrica se basa en funciones matemáticas en lugar de transformaciones, lo que hace que sea mucho más caro de implementar, en términos de los recursos de computación requeridos.



AUTENTICACIÓN

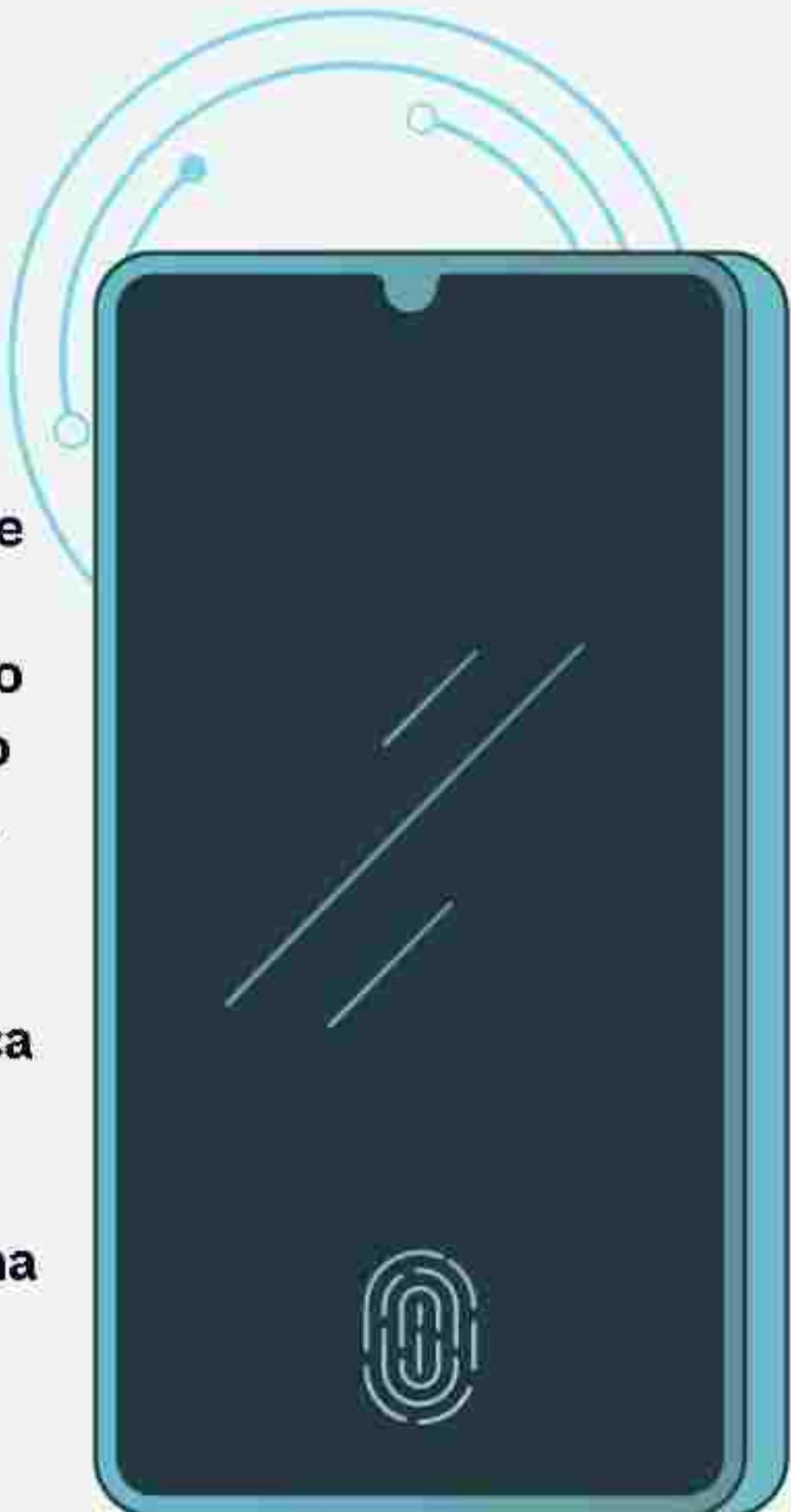
- El proceso de restringir el conjunto de potenciales emisores de un mensaje se denomina autenticación y es complementaria al cifrado. La autenticación también resulta útil para demostrar que un mensaje no ha sido modificado. Un algoritmo de autenticación consta de los siguientes componentes: Un conjunto de claves, un conjunto de mensajes y un conjunto de autenticadores.
- Las funciones hash operan tomando un mensaje en bloques de n bits y procesando los bloques para generar un valor hash de n bits. En un código de autenticación de mensajes, se genera una suma de comprobación criptográfica a partir del mensaje utilizando una clave secreta.



- El segundo tipo de autenticación es el algoritmo de firma digital, y los autenticadores generados por uno de estos algoritmos se denominan firmas digitales.
- Generalmente los algoritmos de autenticación requieren menos cálculos. Un autenticador de un mensaje casi siempre es más corto que el mensaje y su texto cifrado correspondiente. Es la base de los mecanismo de no repudio, que proporcionan una demostración de que una determinada entidad ha realizado una cierta acción.

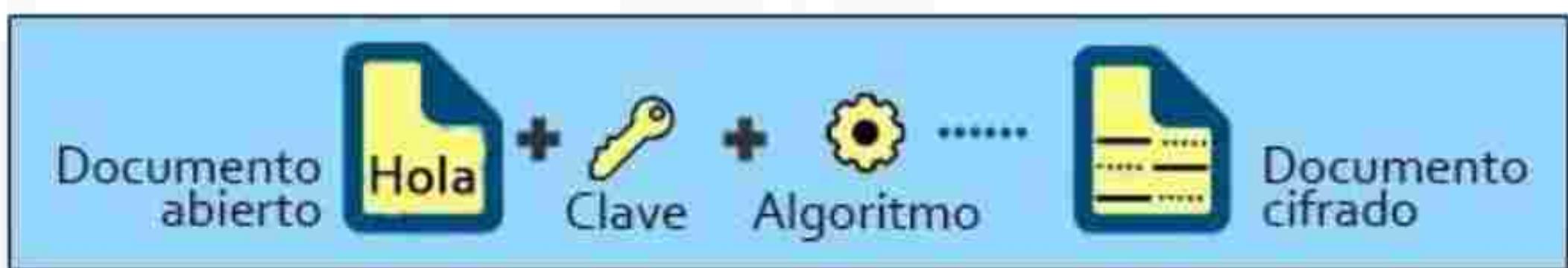
DISTRIBUCIÓN DE CLAVES

- No sólo las claves se pueden intercambiar en público, sino que un determinado usuario sólo necesita una clave privada, independientemente de con cuántas personas desee el usuario comunicarse. El problema se encuentra en la autenticación, lo que necesitamos es demostrar quien posee una determinada clave pública.
- Una forma de resolver este problema consiste en utilizar certificados digitales. Un certificado digital es una clave pública firmada digitalmente por un organismo de confianza. El organismo de confianza recibe la prueba de identificación de alguna entidad y certifica que la clave pública pertenece a dicha entidad.



IMPLEMENTACIÓN DE LOS MECANISMOS CRIPTOGRÁFICOS

Cifrado



Los protocolos de red se organizan en niveles, actuando cada nivel como un cliente del nivel inferior. La criptografía puede incluirse en casi cualquier nivel del modelo ISO. Utilizando cifrado simétrico y el protocolo IKE para el intercambio de claves.

La protección en los niveles inferiores de la pila de protocolos puede resultar insuficiente para los protocolos de los niveles superiores.

Para autenticar un usuario en una computadora cliente, el servidor puede tener que utilizar un protocolo del nivel de aplicación, requiriendo, por ejemplo, que el usuario escriba una contraseña.

Para que el correo sea seguro, los mensajes de correo electrónico tienen que cifrarse de manera que su seguridad sea independiente de los niveles de transporte utilizados para distribuirlos.

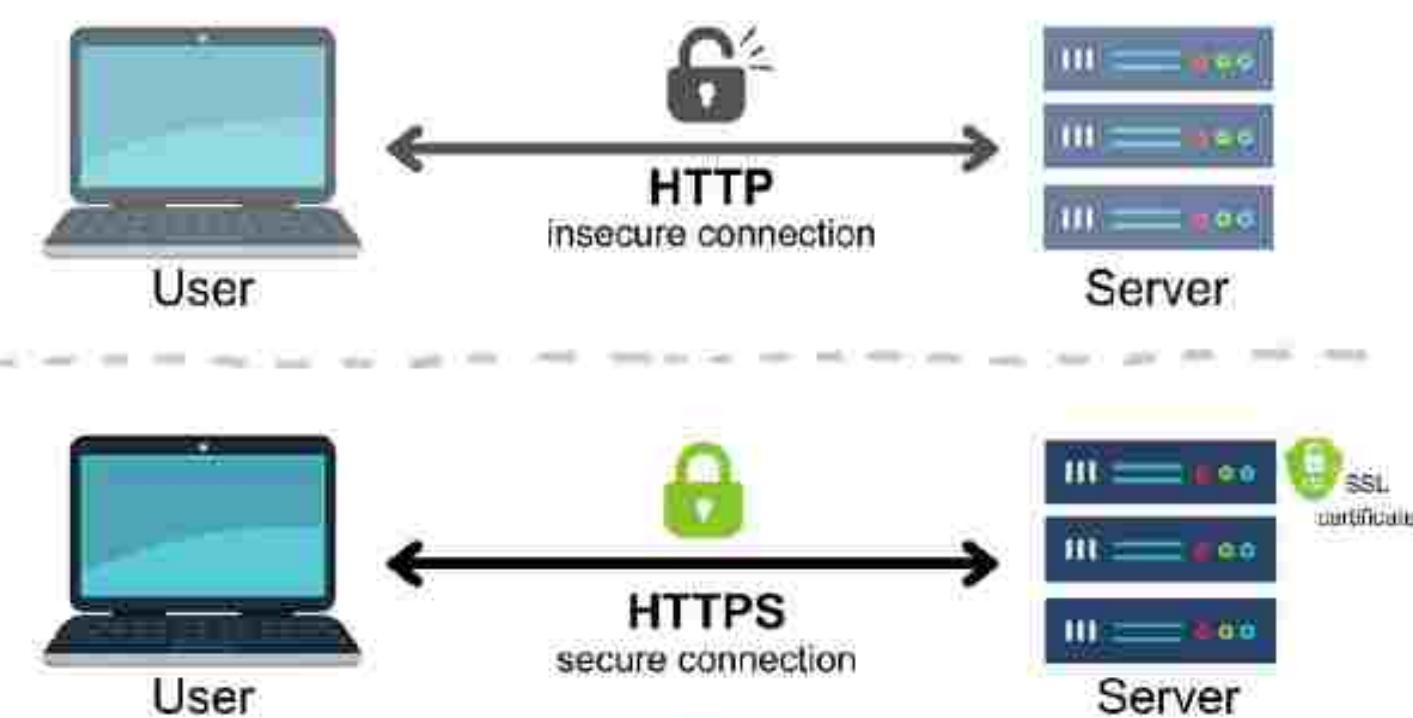
SSL

Es un protocolo criptográfico que permite que dos computadoras se comuniquen de forma segura; es decir, cada una de ellas puede establecer limitaciones que garanticen que el transmisor o receptor de los mensajes sea la otra computadora.

Para aumentar la fortaleza criptográfica, las claves de sesión se olvidan una vez que la sesión se ha completado. Este protocolo permite al servidor limitar los receptores de sus mensajes al cliente que generó el pms y limitar a dicho cliente los emisores de los mensajes que acepta.

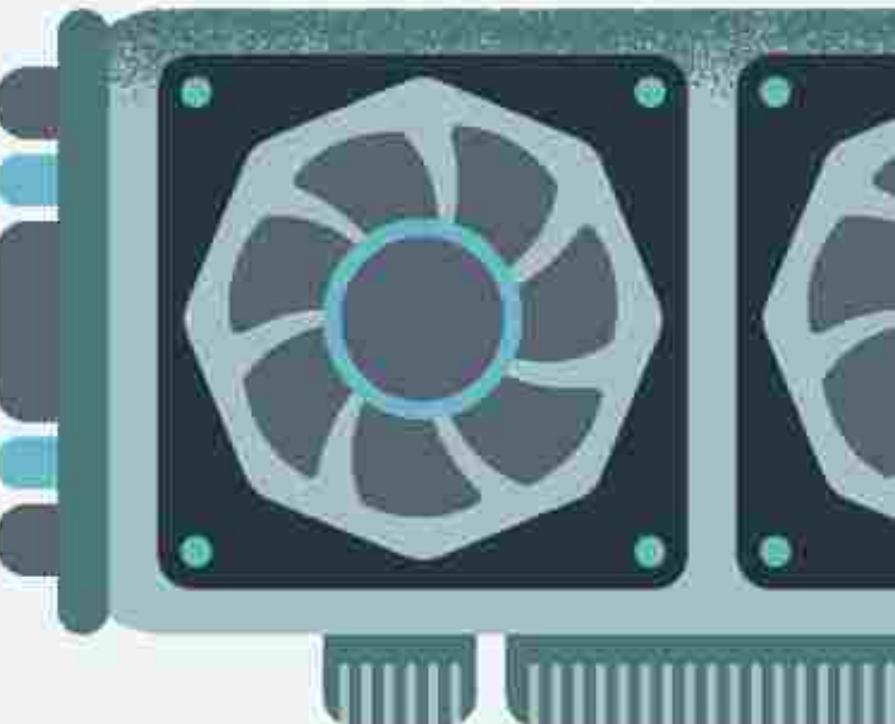
What is SSL certificate?

Secure Sockets Layer (SSL)



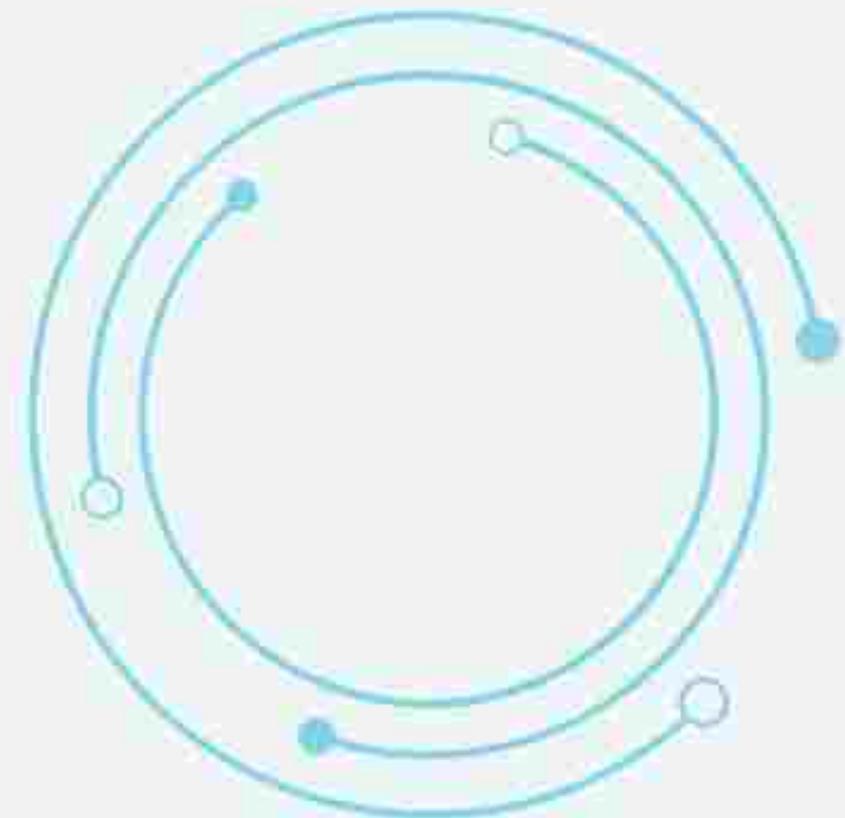
AUTENTICACIÓN DE USUARIO

El sistema de protección depende de la capacidad de identificar los programas y procesos que están actualmente en ejecución, lo que a su vez depende de la capacidad de identificar a cada usuario del sistema. La autenticación de usuario se basa en una o más de tres cuestiones: la posesión de algo por parte del usuario, el conocimiento de algo y un atributo del usuario.



CONTRASEÑAS

- El método más habitual para autenticar la identidad de un usuario consiste en usar contraseñas. Si la contraseña suministrada por el usuario coincide con la contraseña almacenada en el sistema, el sistema pone que el propietario de la cuenta esta accediendo a la misma. Las contraseñas pueden considerarse un caso especial de las claves o de las capacidades.
Pueden asociarse diferentes contraseñas con distintos derechos de acceso. En la práctica, la mayoría de los sistemas sólo requieren una contraseña para que le usuario pueda adquirir todos los derechos.



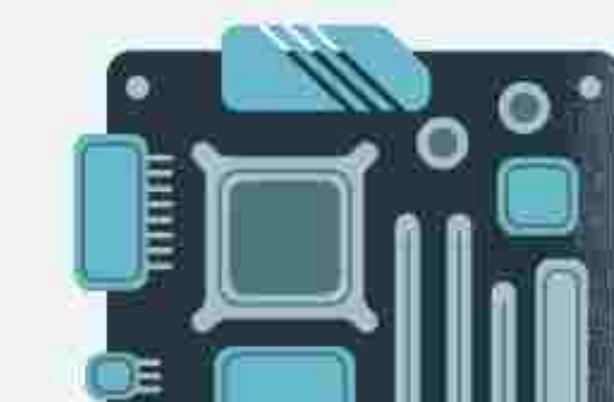


VULNERABILIDAD DE LAS CONTRASEÑAS

Las contraseñas son extremadamente comunes porque son fáciles de comprender y utilizar. Existen dos formas habituales de adivinar una contraseña.

Una forma consiste en que el intruso conoce al usuario o tiene información acerca de él. Otra forma consiste en emplear la fuerza bruta, probando a enumerar todas las posibles combinaciones formadas por caracteres válidos de la contraseña hasta encontrar la contraseña.

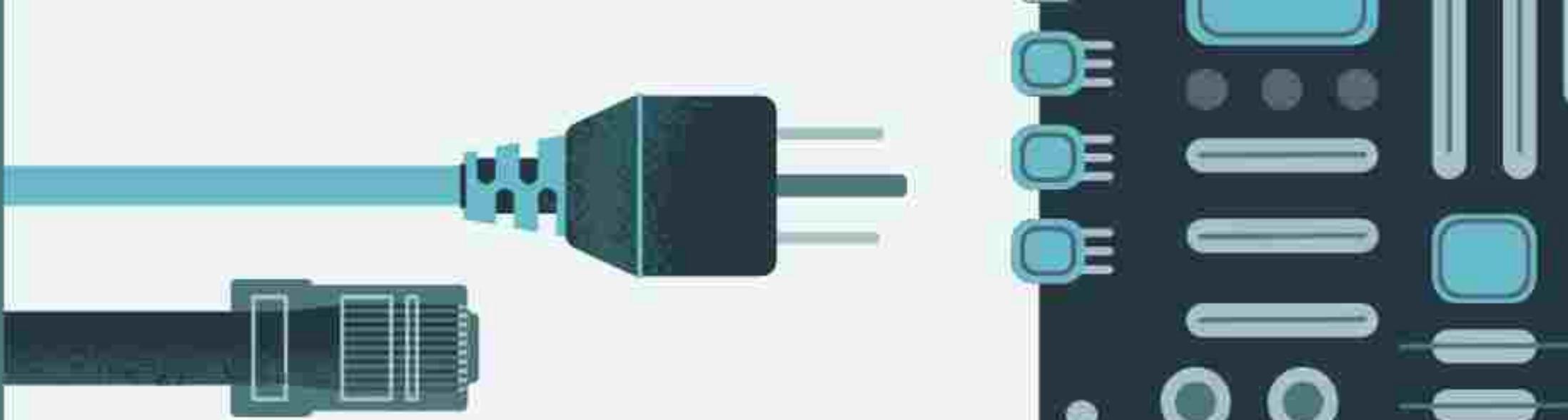
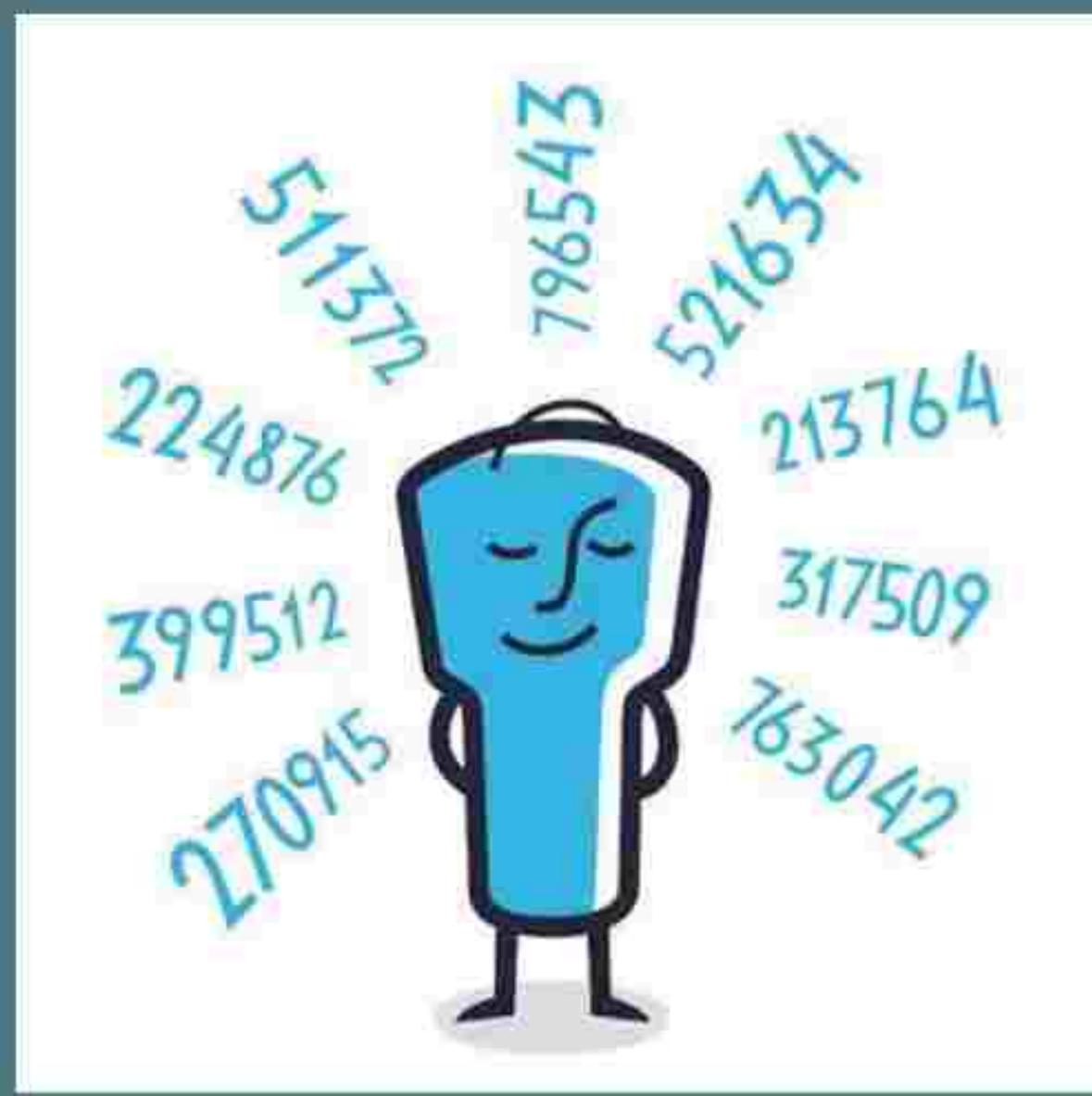
Alternativamente, cualquiera con acceso a la red en la que reside la computadora puede añadir sin problemas un monitor de red, que le permita ver todos los datos que están siendo transferidos a través de la red.



CONTRASEÑAS CIFRADAS

Los sistemas UNIX utilizan el cifrado para evitar la necesidad de mantener en secreto su lista de contraseñas. Cada usuario tiene una contraseña. El sistema contiene una función que es extremadamente difícil de invertir, pero fácil de calcular. Cuando un usuario presenta una contraseña, se codifica y se compara con la contraseña codificada que se tiene almacenada. El aleatorizador se añade a la contraseña para asegurar que, si dos contraseñas son iguales sin cifrar, darán lugar a contraseñas cifradas diferentes.





CONTRASEÑAS DE UN SOLO USO

Cuando se inicia una sesión, el sistema selecciona aleatoriamente una pareja de contraseñas y presenta una parte de la misma; el usuario debe suministrar la otra parte. En este sistema, el usuario es desafiado y debe responder con la respuesta correcta a dicho desafío.

Estas contraseñas algorítmicas no son susceptibles de ser reutilizadas; es decir, un usuario puede escribir una contraseña y ninguna entidad que intercepte dicha contraseña podrá reutilizarla.

BIOMÉTRICA

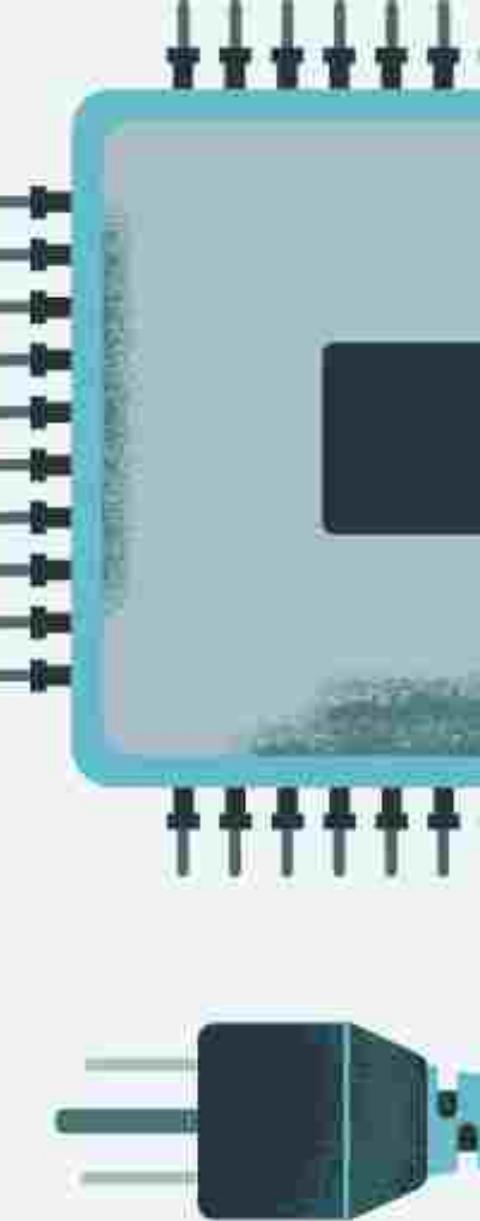
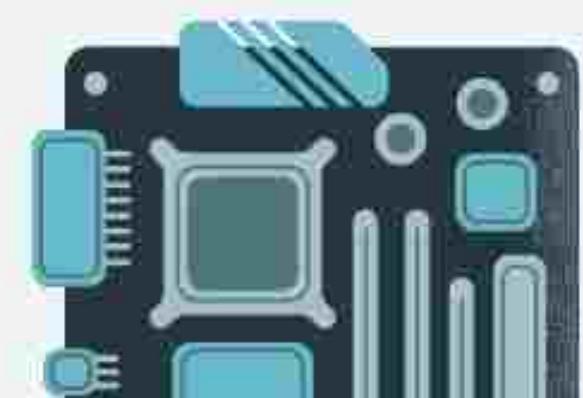
Otra variante implica el uso de medidas biométricas. Los lectores de manos se usan habitualmente para dotar de seguridad a los accesos físicos, como por ejemplo, el acceso a un centro de datos. Estos lectores establecen la correspondencia entre los parámetros almacenados y los que se obtienen mediante los lectores de manos.

Los parámetros pueden incluir un mapa de temperaturas, así como la longitud del dedo, la anchura del mismo y los patrones de las líneas. Los lectores de huellas digitales son muy precisos y su relación coste-efectividad es buena, por lo que en un futuro sean de uso común. Estos dispositivos leen los patrones de las líneas del dedo y los convierten en una secuencia de números.



PROTECCION

Los procesos en un sistema operativo deben protegerse de las actividades realizadas por otros procesos. Para proporcionar dicha protección, podemos utilizar diversos mecanismos para garantizar que sólo los procesos que hayan obtenido la adecuada autorización del sistema operativo pueden operar sobre los archivos, los segmentos de memoria, sobre la CPU y sobre otros recursos del sistema.

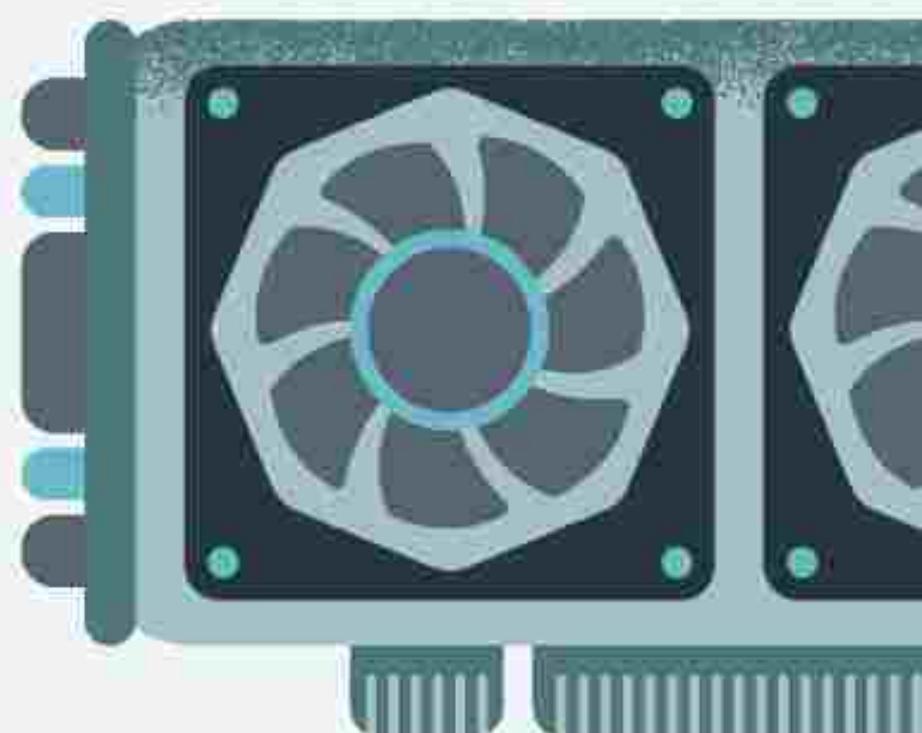


OBJETIVOS DE LA PROTECCION

A medida que los sistemas informáticos se han hecho más sofisticados y a medida que su rango de aplicaciones se ha ido incrementando, también ha crecido la necesidad de proteger la integridad de esos sistemas.

Se implementa la protección por la necesidad de impedir una violación maliciosa e intencionada de una restricción de acceso por parte

Los conceptos modernos de protección han evolucionado para incrementar la fiabilidad de cualquier sistema complejo que haga uso de recursos compartidos.



Los mecanismos de protección pueden mejorar la fiabilidad detectando los errores latentes en las interfaces definidas entre los distintos subsistemas componentes.

La detención temprana de errores de interfaz puede a menudo impedir que un subsistema correcto se vea contaminado por otro que no esté funcionando adecuadamente.

Las políticas de uso de recursos pueden variar según la aplicación y también pueden variar a lo largo del tiempo, por ello, la protección no es sólo cuestión del diseñador de un sistema operativo.



El papel de la protección en un sistema informático es proporcionar un mecanismo para la imposición de las políticas que gobiernan un recurso. Estas políticas pueden establecerse de diversas formas.

Algunas fijas en el diseño de un sistema.

Existen otras que son definidas por los usuarios individuales para proteger sus propios archivos y programas.

El programador de aplicaciones necesita utilizar también los mecanismos de protección para defender de un uso incorrecto los recursos creados y soportados por un subsistema de aplicación.

OBJETIVOS DE LA PROTECCION

LOS MECANISMOS DETERMINAN COMO SE LLEVARA ALGO A ACABO

LAS POLITICAS DECIDEN QUE ES LO QUE HACER.

La separación entre políticas y mecanismos resulta importante si queremos tener una cierta flexibilidad. Las políticas cambian de un lugar a otro, cada cambio requeriría también de un cambio en el mecanismo subyacente; la utilización de mecanismos generales permite evitar este tipo de situaciones.

PRINCIPIOS DE PROTECCION

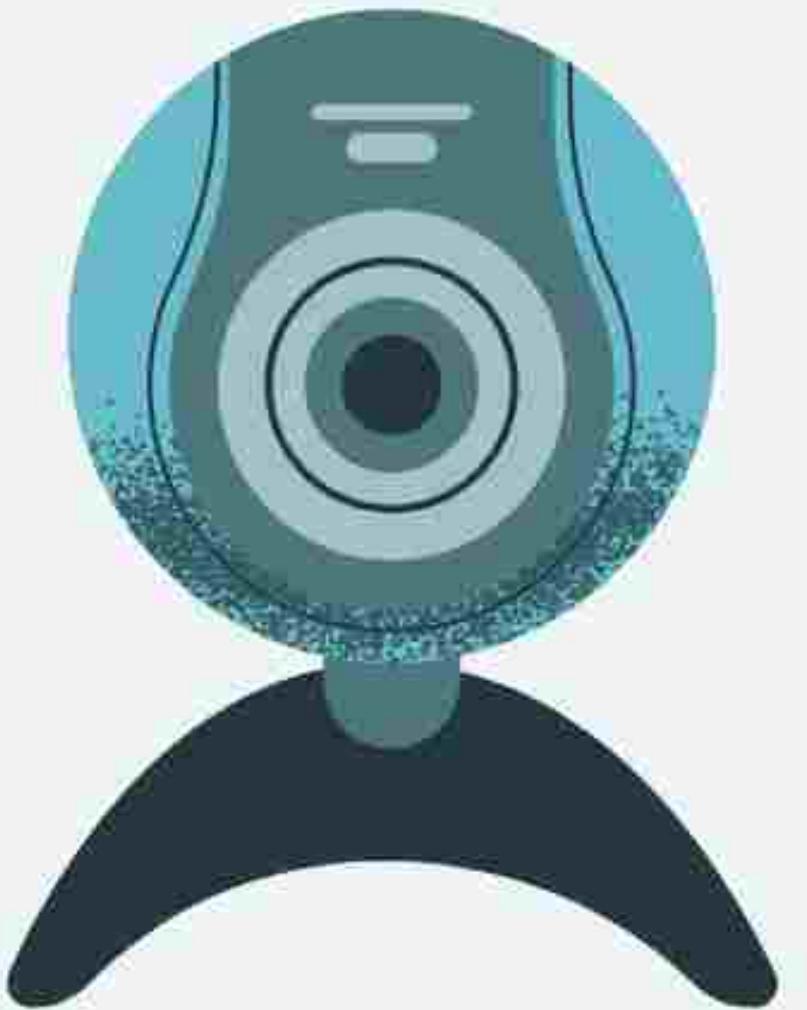
01

PRINCIPIO DE MÍNIMO PRIVILEGIO

Este privilegio dicta que, a los programas, a los usuarios e incluso a los sistemas se les concedan únicamente los suficientes privilegios para llevar a cabo sus tareas. Forma parte de los directores claves y que ha resistido con el paso del tiempo al momento de proporcionar protección.

Un sistema operativo que se ajuste al principio del mínimo privilegios implementará sus características, programas, llamadas a sistema y estructuras de datos de modo que el fallo o el compromiso de un componente provoquen un daño mínimo, únicamente.

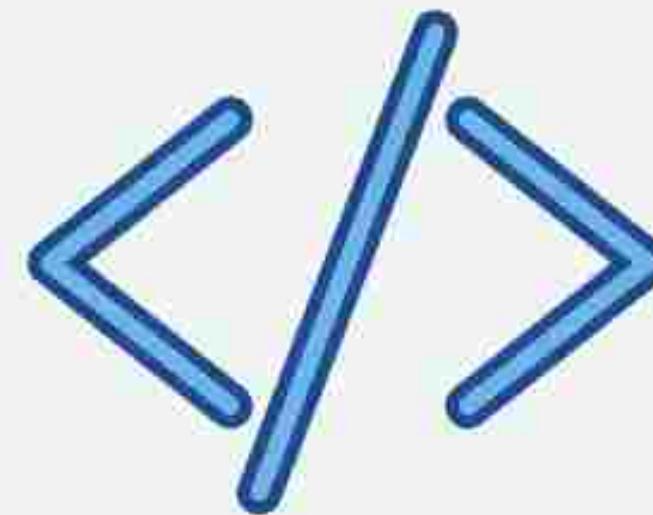
También proporcionará llamadas a sistema y servicios que permitan escribir aplicaciones con controles de acceso de granularidad fina, además de mecanismos para activar los privilegios cuando sean necesarios y desactivarlos cuando dejan de serlo.



La creación de pistas de auditoría resulta una ventaja para todos los accesos a funciones privilegiadas, permitiendo al programador, al administrador del sistema o a los miembros de las fuerzas del orden, revisar todas las actividades realizadas en el sistema que estén relacionadas con los mecanismos de protección y de seguridad.

La gestión de los usuarios con el principio del mínimo privilegios implica crear una cuenta separada para cada usuario, con sólo los privilegios que requiera. Algunos sistemas implementan mecanismos de control de acceso basado en roles para proporcionar esta funcionalidad.

El principio de mínimo privilegio puede ayudar a obtener un entorno informático más seguro. Desafortunadamente, con frecuencia no lo hace.



DOMINIO DE PROTECCION

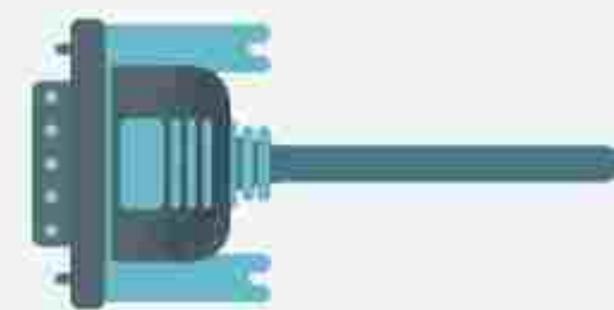
Un sistema informático es una colección de procesos y objetos.



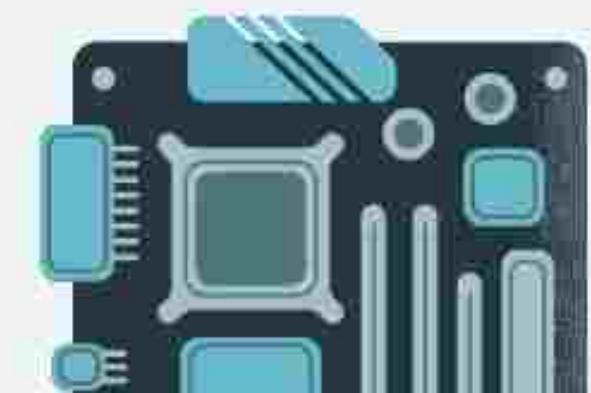
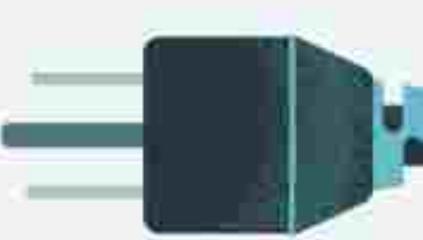
Con objetos se define tanto objetos hardware y objetos software. Cada objetos tiene un nombre distintivo que lo diferencia de todos los demás objetos del sistema y sólo se puede acceder a cada objeto mediante operaciones significativas bien definidas.



A un proceso sólo se le debe permitir acceder a aquellos recursos que tenga autorización, además, en cualquier instante, un proceso sólo debería poder acceder a aquellos recursos que necesite actualmente para completar su tarea.



El compilador no debería poder acceder a archivos arbitrariamente, sino que sólo debería acceder a un subconjunto bien definido de los archivos relacionados con el archivo que hay que compilar.

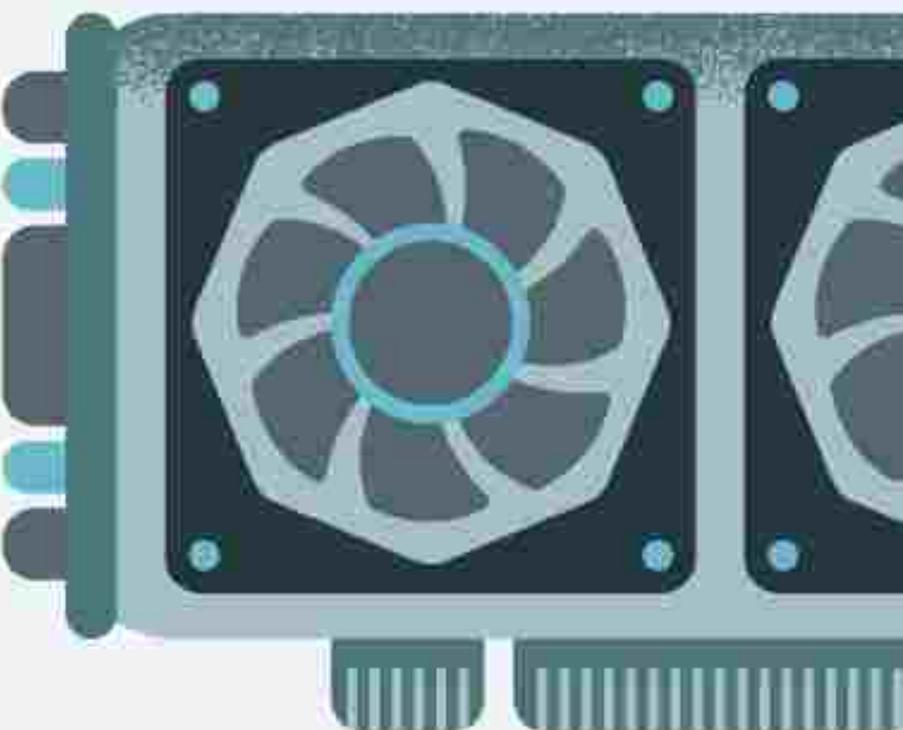


ESTRUCTURAS DE DOMINIOS

Un proceso opera dentro de un dominio de protección, que especifica los recursos a los que el proceso puede acceder. Cada dominio define un conjunto de objetos y los tipos de operaciones que pueden invocarse sobre cada objeto.

Los dominios no tienen por qué ser disjuntos, sino que pueden compartir derechos de acceso.

La capacidad de ejecutar una operación sobre un objeto es un derecho de acceso. Un dominio es una colección de derechos de acceso, cada uno de los cuales es una pareja ordenada <nombre-objeto,conjunto-derecho>.



La asociación entre un proceso y un dominio puede ser estática, si el conjunto de recursos disponibles para el proceso está fijo durante la vida del proceso, o dinámica. Establecer dominios dinámicos de protección es más complicado que establecer dominios estáticos de protección.

Si un domino es estático, se deberá definir el domino para que incluya tanto el acceso de lectura como el de escritura, sin embargo, esta disposición proporciona más derechos de los necesarios en cada una de las dos fases, ya que se tiene acceso de lectura en la fase donde sólo se necesita acceso a escritura y viceversa.

Si la asociación entre los procesos y los dominios es fija y se quiere adherir al principio de la necesidad de conocer, deberá haber disponible un mecanismo para cambiar el contenido de un dominio.

Si la asociación es dinámica, habrá disponible un mecanismo para permitir la conmutación de dominio, permitiendo al proceso conmutar de un dominio a otro. También se puede permitir que se modifique el contenido de un dominio.

Un dominio puede llevarse a la práctica de diversas formas:

CADA USUARIO PUEDE SER UN DOMINIO

El conjunto de objetos a los que se podrá acceder dependerá de la identidad del usuario. La conmutación de dominios tiene lugar cuando cambia el usuario, generalmente cuando un usuario cierre la sesión o otro usuario la inicie.

CADA PROCEDIMIENTO PUEDE SER UN DOMINIO

El conjunto de objetos a los que se podrá acceder se corresponderá con las variables locales definidas dentro del procedimiento. La conmutación de dominios sólo se puede cuando se lleve a cabo una llamada a procedimiento.

CADA PROCESO PUEDE SER UN DOMINIO

El conjunto de objetos a los que se podrá acceder dependerá de la identidad del proceso. La conmutación de dominio tendrá lugar cuando un proceso envíe un mensaje a otro proceso y espere una respuesta.

MATRIZ DE ACCESO

1

Las filas de la matriz de acceso representan dominios y las columnas representan objetos. Cada entrada de la matriz está compuesta de un conjunto de derechos de acceso. Puesto que la columna define los objetos, se puede omitir el nombre del objeto del derecho de acceso.

2

El esquema basado en matriz de acceso proporciona mecanismos para especificar una diversidad de políticas. Esté mecanismo consisten en implementar la matriz de acceso y garantizar que las propiedades semánticas que se han esbozado se cumplan.

3

La matriz de acceso puede implementar decisiones de política relativas a la protección. Las decisiones de política implican qué derechos deben incluirse en la entrada. Además, de definir el dominio en el que cada proceso se deberá de ejecutar (definida por el sistema operativo).

4

Proporciona el mecanismo apropiado para definir e implementar un control estricto de la asociación estática como dinámica entre procesos y dominios. Cuando se conmuta un proceso de un dominio a otro, se ejecuta una operación sobre un objeto (switch - dominio).

IMPLEMENTACION DE LA MATRIZ DE ACCESO

Tabla global

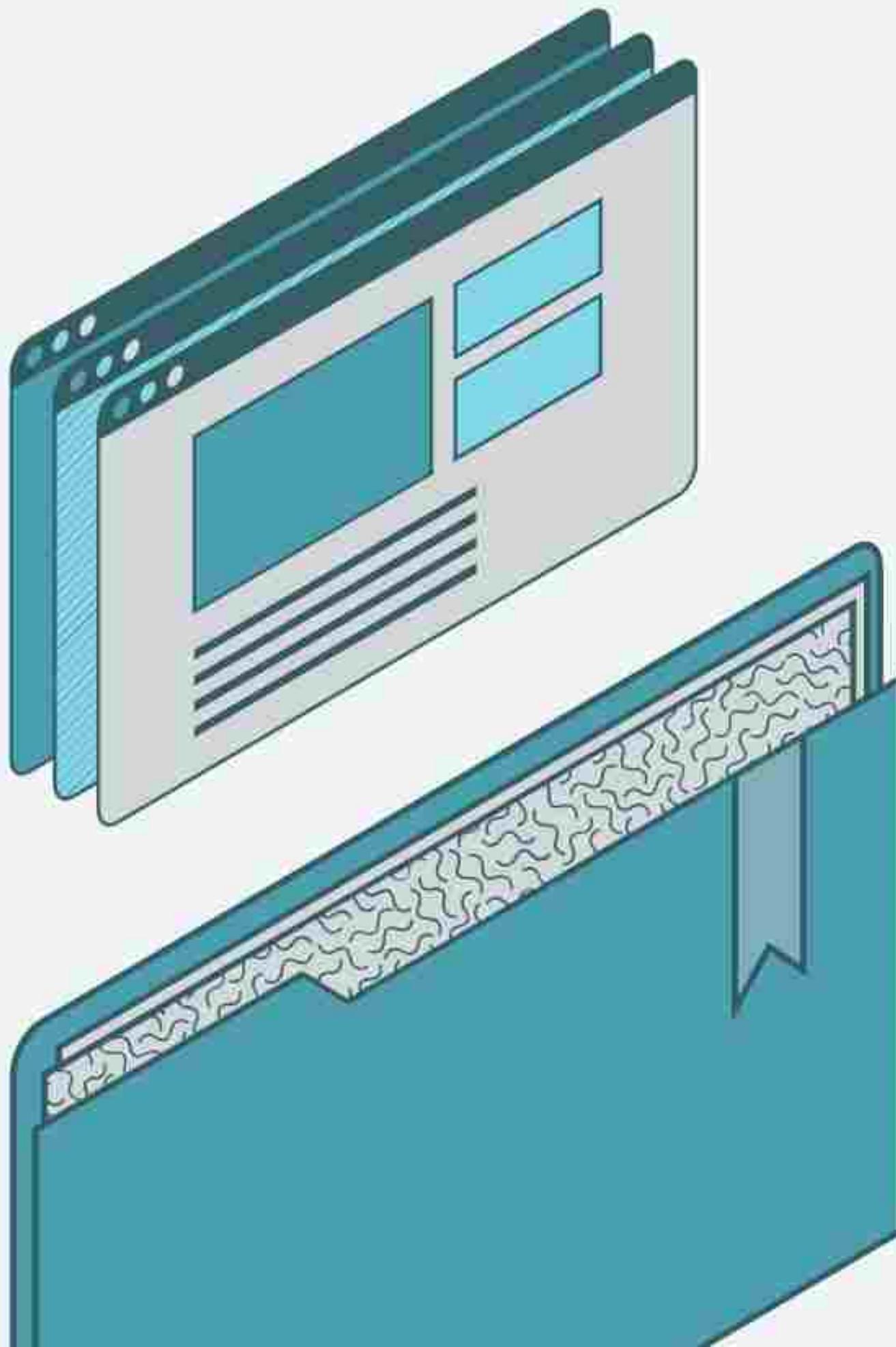
Es una tabla compuesta de un conjunto de tripletas ordenadas <dominio,objeto,conjunto-derecho>. Cada vez que se ejecuta una operación M sobre un objeto Oj dentro del dominio Di, se analiza la tabla global en busca de una tripleta $>Di, Oj, Rk>$, con $M \rightarrow Rk$. Si se encuentra, se permite que la operación continúe, en caso contrario se genera una condición de excepción (error).

Tiene varias desventajas, pues la tabla usualmente es muy grande y no puede, por tanto, ser conservada en la memoria principal, por lo que hacen falta operaciones adicionales de E/S. Y para ello suelen utilizarse técnicas de memoria virtual para gestionar cada tabla.

Lista de acceso para los objetos

Cada columna de la matriz puede implementarse como una lista de acceso de un objeto, donde las entradas vacías pueden descartarse. La lista resultante para cada objeto estará compuesta por una serie de parejas ordenadas <dominio, conjunto.derechos>, que definen todos los dominios que tengan un conjunto de derechos de acceso no vacío para dicho objeto.

Puede extenderse para definir una lista más un conjunto predeterminado de derechos de acceso. Cuando se intenta realizar una operación M sobre un objeto Oj en el dominio Di, se busca una entrada $<Di, Rk>$. Si se encuentra, se permite la operación y en caso contrario, se comprueba el conjunto predeterminado.



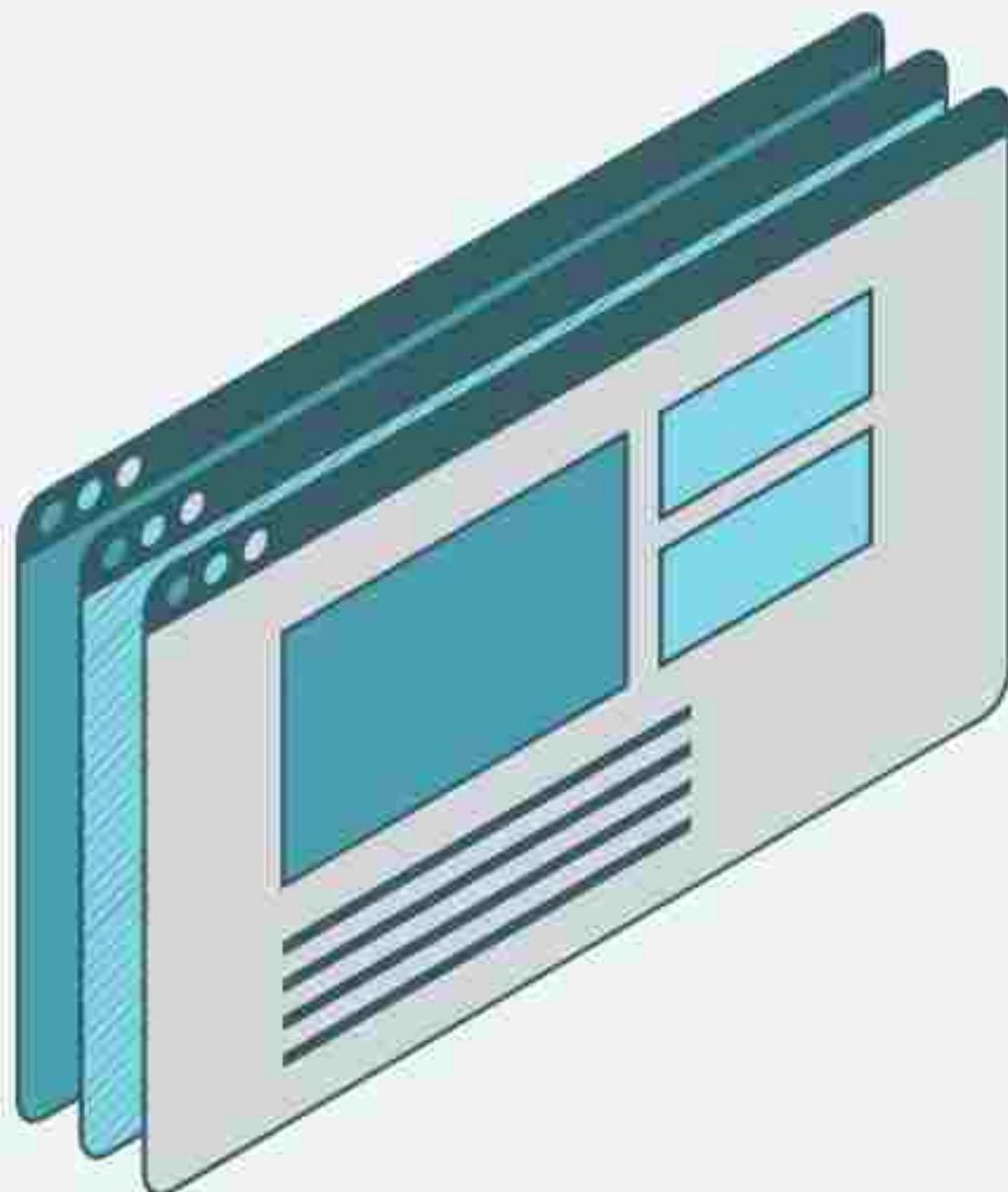
IMPLEMENTACION DE LA MATRIZ DE ACCESO

Lista de capacidades para los dominios

Es una lista de objetos junto con las operaciones permitidas sobre esos objetos. Cada objeto se suele representar mediante su dirección o nombre físico, denominada capacidad. Para ejecutar una operación, el proceso ejecuta la operación especificando la capacidad (o puntero) para el objeto como parámetro.

La lista de capacidades está asociada con un dominio, pero un proceso que se ejecute en ese dominio no puede nunca acceder directamente a ella. Por el contrario, la lista de capacidades es en sí misma un objeto protegido, mantenido por el sistema operativo y al que el usuario sólo puede acceder indirectamente.

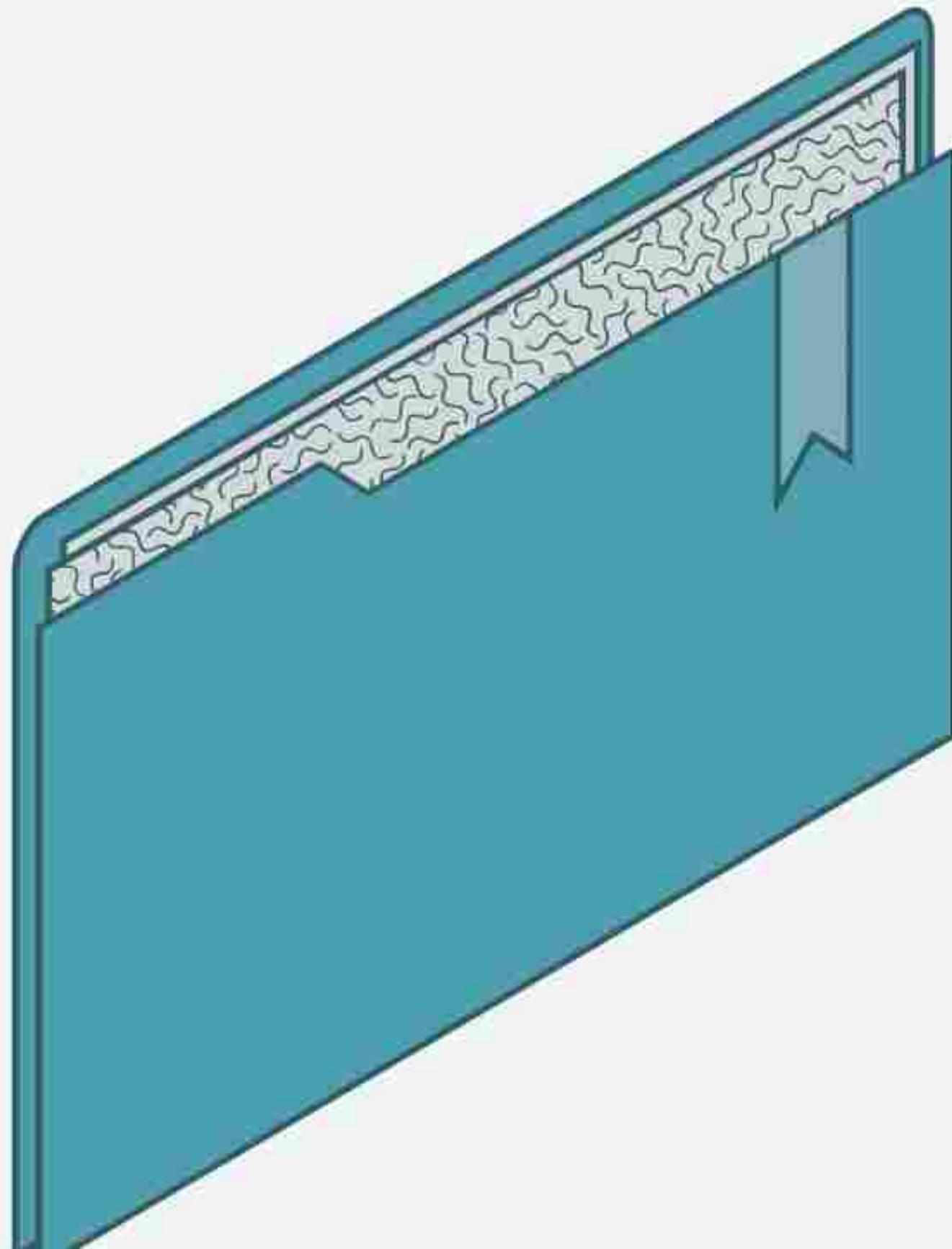
Estas capacidades se propusieron originalmente como una especie de puntero seguro, para satisfacer la necesidad de protección de los recursos que se preveía que iba a ser necesaria a medida que los sistemas informáticos multiprogramados se generalizaran.



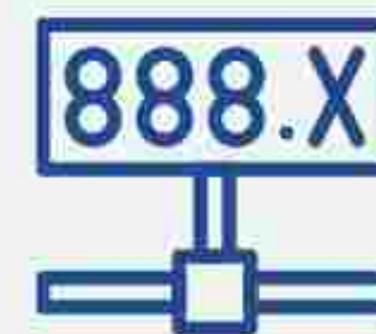
Lista de capacidades para los dominios

Cada objeto tiene un etiqueta para denotar su tipo, el cual indica si se trata de una capacidad o de un dato accesible. Las propias etiquetas no deben ser directamente accesibles por parte de un programa de aplicación.

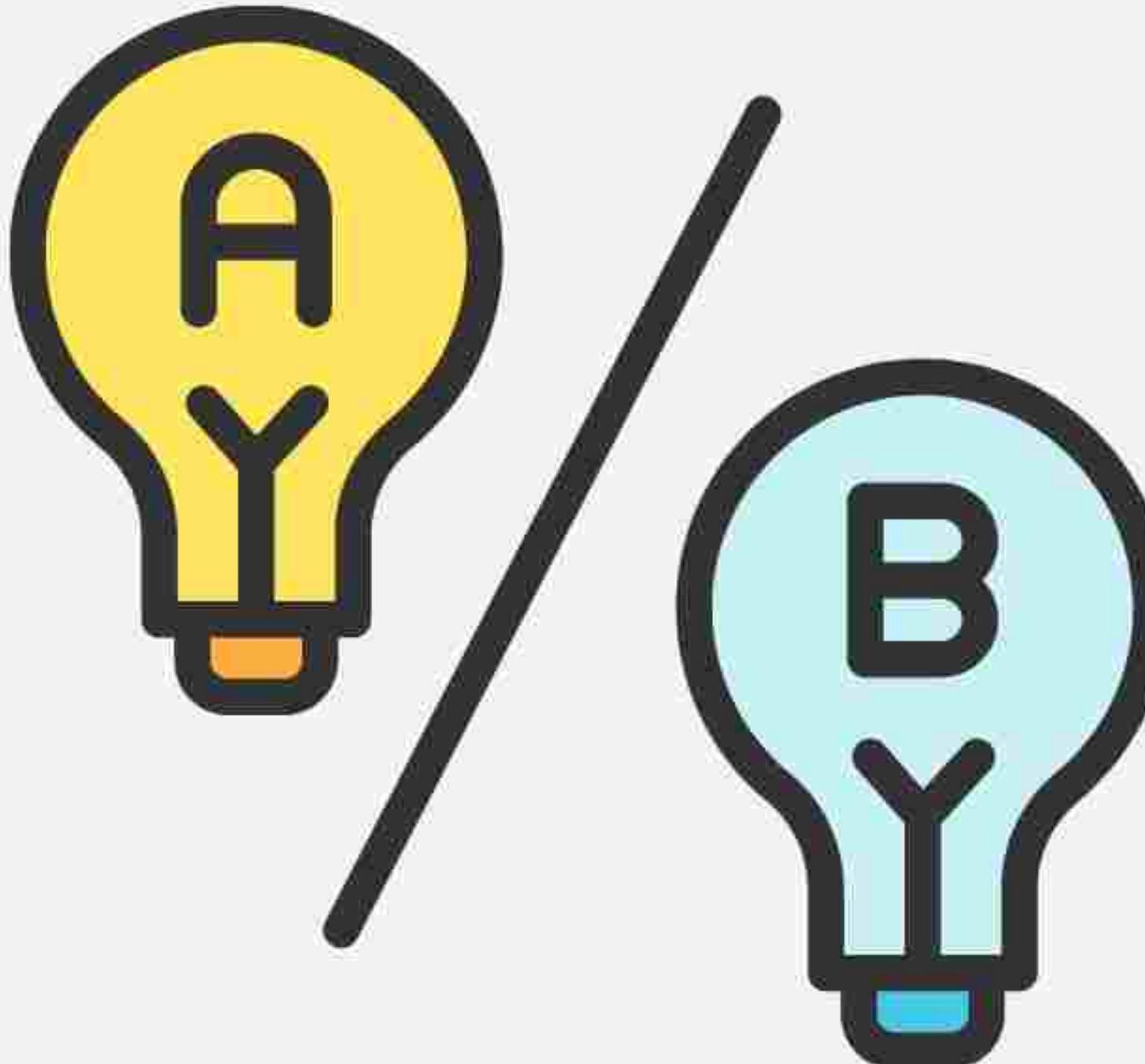
Puede utilizarse soporte hardware o firmware para imponer la restricción. Aunque sólo hace falta un bit para distinguir entre capacidades y otros objetos, a menudo se utilizan bits adicionales.



El espacio de direcciones asociado con un programa puede dividirse en dos partes. Una parte es accesible para el programa y contiene las instrucciones y los datos normales del programa. La otra parte, que contiene la lista de capacidades, sólo es accesible por el sistema operativo.



Mecanismo de bloqueo-clave



Es un compromiso entre las listas de acceso y las listas de capacidades. Cada objetos tienen una lista de patrones de bit distintivos, denominados bloquesos. De forma similar, cada dominio tiene una lista de patrones de bit distintivos, denominados claves. Un proceso que se ejecute dentro de un dominio podrá acceder a un objeto sólo si dicho dominio tiene una clave que se corresponda con uno de los bloqueos del objeto.

La lista de claves de un dominio debe ser gestionada por el sistema operativo por cuenta del dominio. Los usuarios no están autorizados a examinar o modificar la lista de claves (o de bloqueos) directamente.

Comparación

El espacio de direcciones asociado con un programa puede dividirse en dos partes. Una parte es accesible para el programa y contiene las instrucciones y los datos normales del programa. La otra parte, que contiene la lista de capacidades, sólo es accesible por el sistema operativo.

CONTROL DE ACCESO



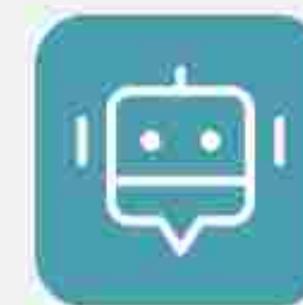
1

A cada archivo y directorio de los que se le asignan un propietario, un grupo o posiblemente una lista de usuarios y para cada una de estas entidades se asigna una información de control de acceso.



2

Se puede añadir una función similar a otros aspectos de un sistema informático. Por ejemplo en esta estrategia se encuentra en Solaris 10



3

Solaris 10 amplía el sistema de protección disponible en el sistema operativo Sun Microsystems añadiendo explícitamente el principio de mínimo privilegio mediante el control de acceso basado en roles.



4

Esa funcionalidad gira en torno a los privilegios. Un privilegio es el derecho a ejecutar una llamada a sistema o a usar una opción dentro de dicha llamada a sistema.

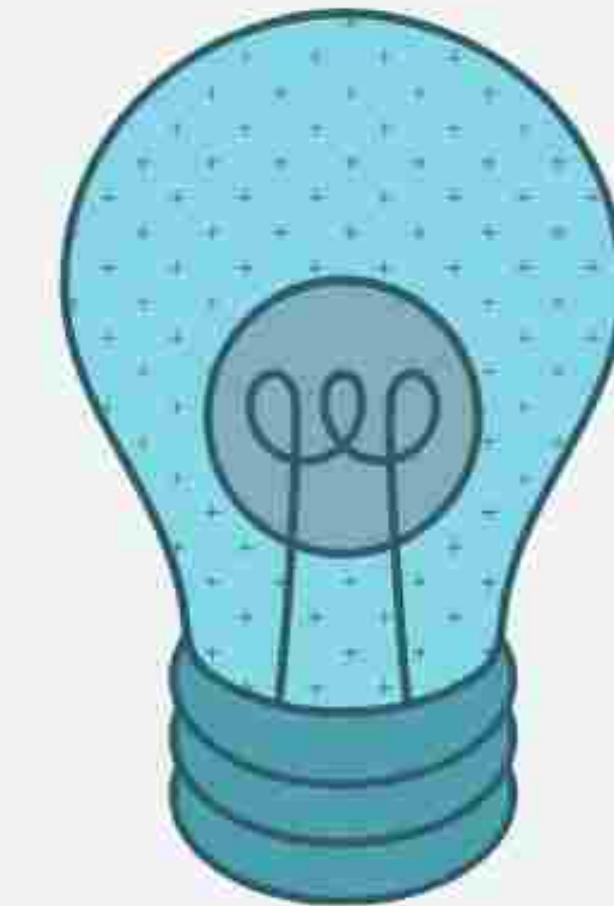


READQUISICION

Se borran las capacidades de cada dominio de forma periodica. Si un proceso quiere usar una capacidad, puede que se encuentre con que esa capacidad ha sido borrada.

RETROPUNTEROS

Cuando se necesita realizar una revocación, se pueden seguir los punteros, cambiando las capacidades según sea necesario. Dicho esquema fue adoptado en el sistema MULTICS.



INDIRECCION

Las capacidades apuntan indirectamente a los objetos. Cada capacidad apunta a una entrada única dentro de una tabla global, que a su vez apunta al objeto. Se implementa la revocación analizando la tabla global en busca de la entrada deseada y borrándola.

CLAVES

Es un patrón distintivo de bits que puede asociarse con una capacidad. Esta clave se define en el momento de crear la capacidad y no puede ser nunca modificada ni inspeccionada por el proceso que posee la capacidad.

REVOCACION DE DERECHOS DE ACCESO

Este esquema no permite la revocación selectiva, ya que con cada objeto sólo hay asociada una clave maestra. Si se asocia una lista de claves con cada objeto, entonces se podría implementar la revocación selectiva. Una capacidad será válida sólo si su clave se corresponde con alguna de las claves de la tabla global.

En los esquemas basados en claves, las operaciones de definición de claves, de inserción de claves en listas y de borrado de claves de las listas no deben estar disponibles para todos los usuarios.

SISTEMAS BASADOS EN CAPACIDADES

Hydra es un sistema basado en capacidades que ofrece flexibilidad en los derechos de acceso.

- Incluye derechos básicos como leer, escribir o ejecutar, y permite a los usuarios definir derechos adicionales.
- Las operaciones sobre objetos se definen proceduralmente y se accede a ellas mediante capacidades.
- Hydra permite la amplificación de derechos para procedimientos de confianza, permitiendo un acceso más amplio bajo ciertas condiciones.



HYDRA

- Los errores que pueden ocurrir tanto en el software como en el hardware y cómo Hydra limita las amplificaciones para resolver estos problemas.
- Se describe el mecanismo de llamada a procedimientos de Hydra como una solución al problema de subsistemas mutuamente sospechosos.
- Los subsistemas se construyen sobre el kernel de protección de Hydra y se protegen mediante un conjunto de primitivas definidas por el kernel.
- Los programadores pueden utilizar directamente el sistema de protección de Hydra familiarizándose con el manual y utilizando la biblioteca de procedimientos proporcionada.

SISTEMAS BASADOS EN CAPACIDADES

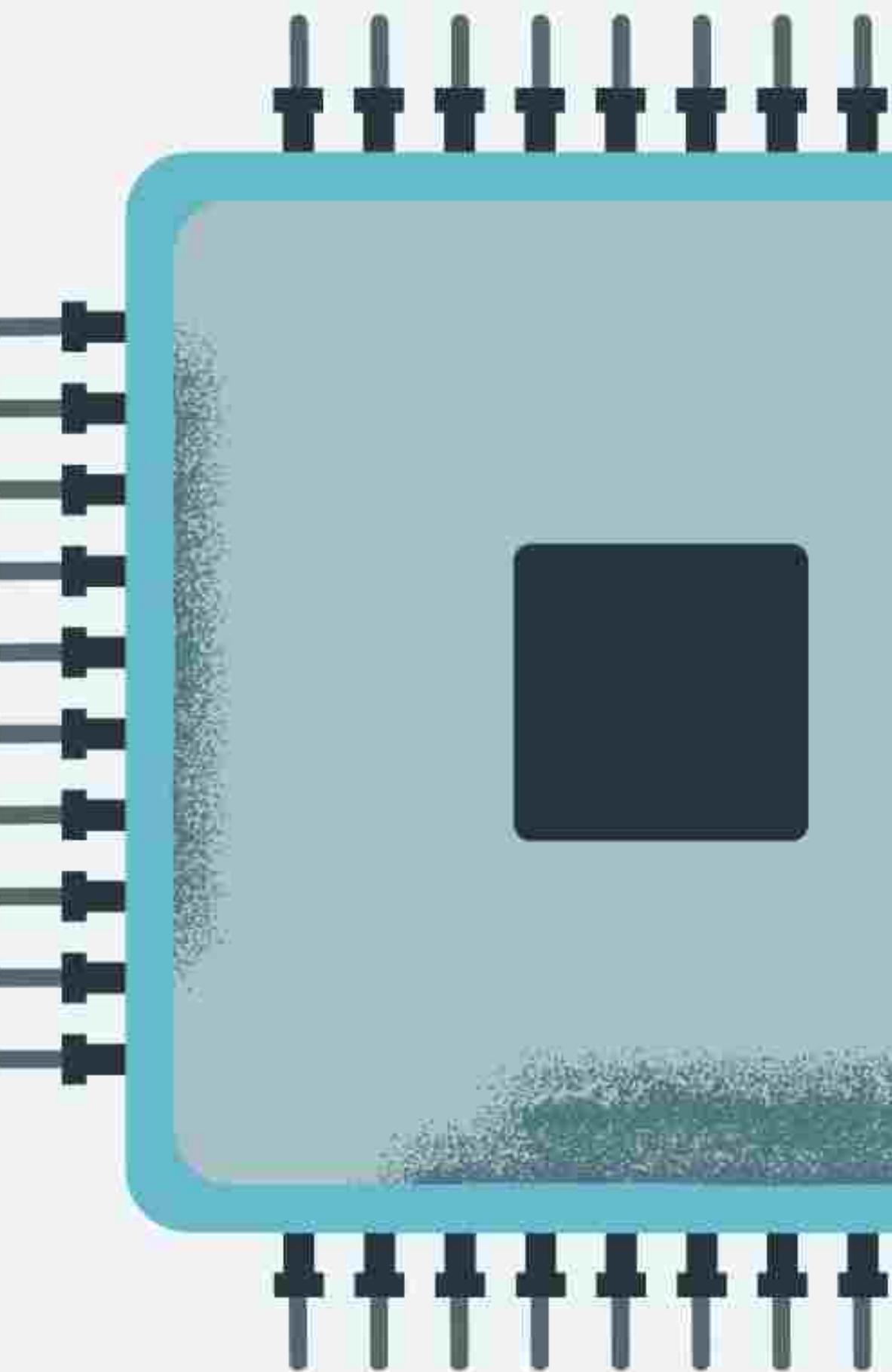
El Sistema CAP tiene un enfoque distinto para la implementación de protección basada en capacidades.

- Capacidades de Datos: Acceso a objetos con derechos normales de lectura, escritura y ejecución.
- Capacidad Software: Protegida y utilizada para implementar políticas de protección a través de procedimientos protegidos.
- Amplificación de Derechos: Procesos adquieren temporalmente derechos para leer o escribir en una capacidad software

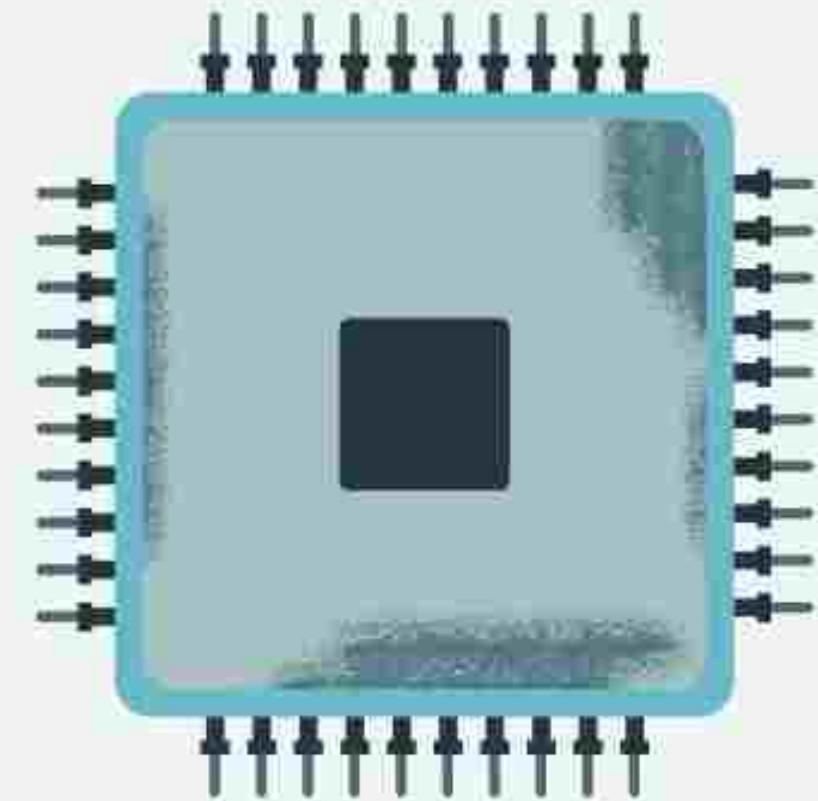


SISTEMA CAP DE CAMBRIDGE

- El sistema básico de protección evita que procedimientos no verificados accedan a segmentos de almacenamiento fuera de su entorno de protección.
- Un procedimiento protegido inseguro puede resultar en un fallo de protección del subsistema responsable.
- Los diseñadores utilizan capacidades software para formular e implementar políticas de protección sin depender de manuales o bibliotecas de procedimientos.



VIRTUALIZACIÓN



CONCEPTOS

REQUERIMIENTOS

HIPERVISORES

PARAVIRTUALIZACIÓN

VIRTUALIZACIÓN DE LA MEMORIA

VIRTUALIZACIÓN DE LA E/S

DISPOSITIVOS VIRTUALES

MAQUINAS VIRTUALES EN
MULTINUCLEOS

CUESTIONES SOBRE LICENCIAS

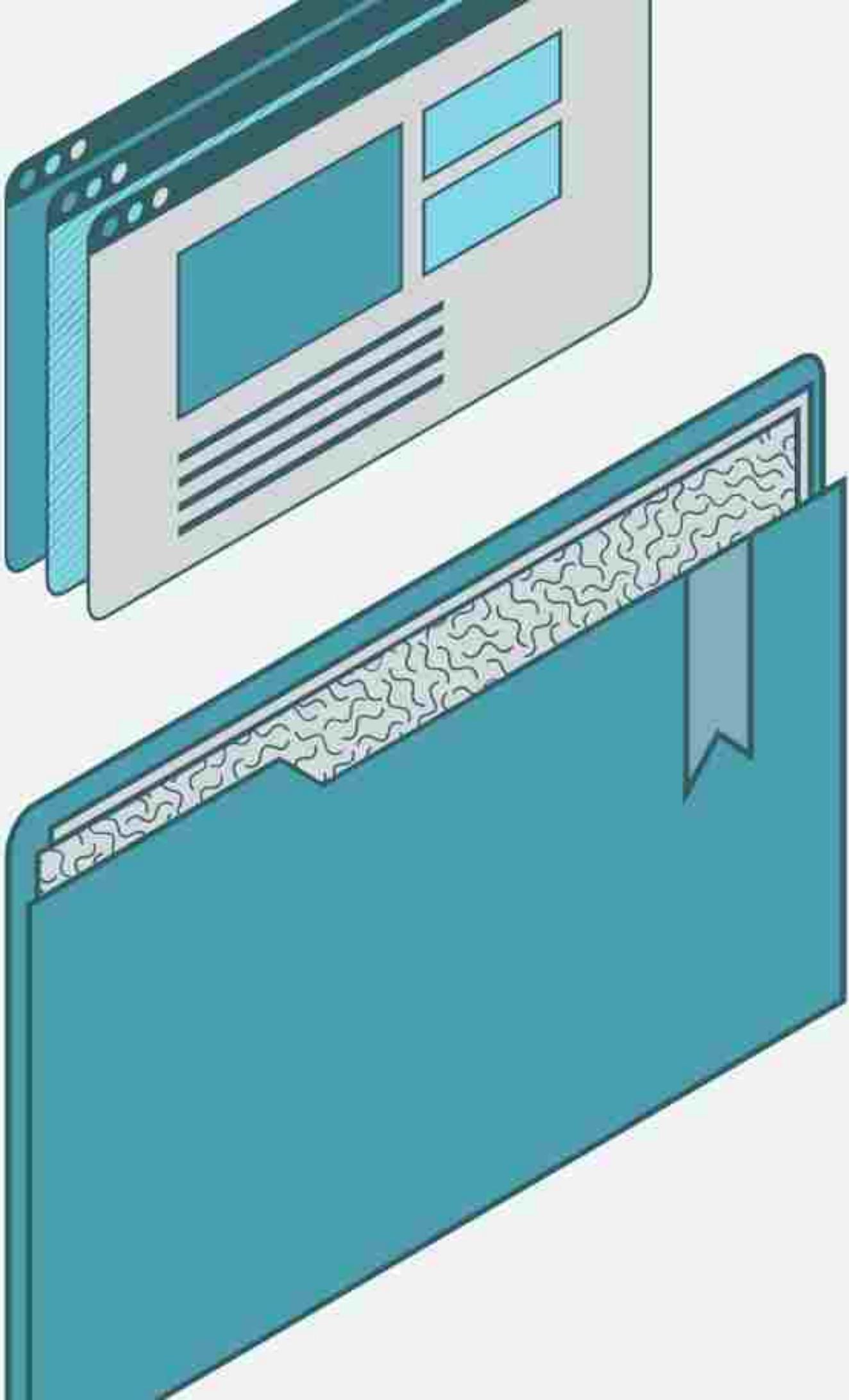
CONCEPTOS

01. Definición

Tecnología que permite la creación de entornos virtuales (sistemas operativos, redes o almacenamiento), en una sola computadora física. Que es logrado mediante la abstracción de los recursos físicos, permitiendo ejecutar múltiples máquinas virtuales de manera simultáneamente en una sola máquina.

02. Ventajas

1. **Ahorro de Recursos y Eficiencia Energética:** Reduce la cantidad de hardware necesario para ejecutar múltiples S.O y aplicaciones, y a menor hardware menor consumo.
2. **Facilidad de Administración:** Simplifica la gestión al centralizar la infraestructura.
3. **Tolerancia a Fallos y Aislamiento:** Mayor disponibilidad de los servicios que pueden migrar rápidamente a otro hardware en caso de falla, haciendo que las fallas en una máquina virtual no afectan a las demás.
4. **Flexibilidad:** Permite probar y desarrollar software en diferentes sistemas operativos sin necesidad de hardware adicional.
5. **Optimización de Recursos:** Facilita el balanceo de carga y la asignación dinámica de recursos según las necesidades.



CONCEPTOS

03. Historia

Década de 1960, IBM introdujo el IBM System/360 Model 67, modelo capaz de ejecutar múltiples S.O y aplicaciones simultáneamente, permitiendo simular múltiples entornos de computación en una sola máquina física.

Su desarrollo fue impulsado por la necesidad de mejorar la eficiencia y utilización de los recursos del hardware, permitido a las empresas ejecutar múltiples cargas de trabajo en un solo sistema, reduciendo los costos y aumentando la productividad.

Década de 1970, IBM dio el lanzamiento del sistema operativo VM/370, diseñado específicamente para permitir la ejecución de múltiples S.O en una sola computadora ("máquinas virtuales").

VM/370 introdujo el concepto de *hipervisor*, un software que se ejecuta directamente en el hardware y que gestiona la creación y el funcionamiento de las máquinas virtuales.



CONCEPTOS



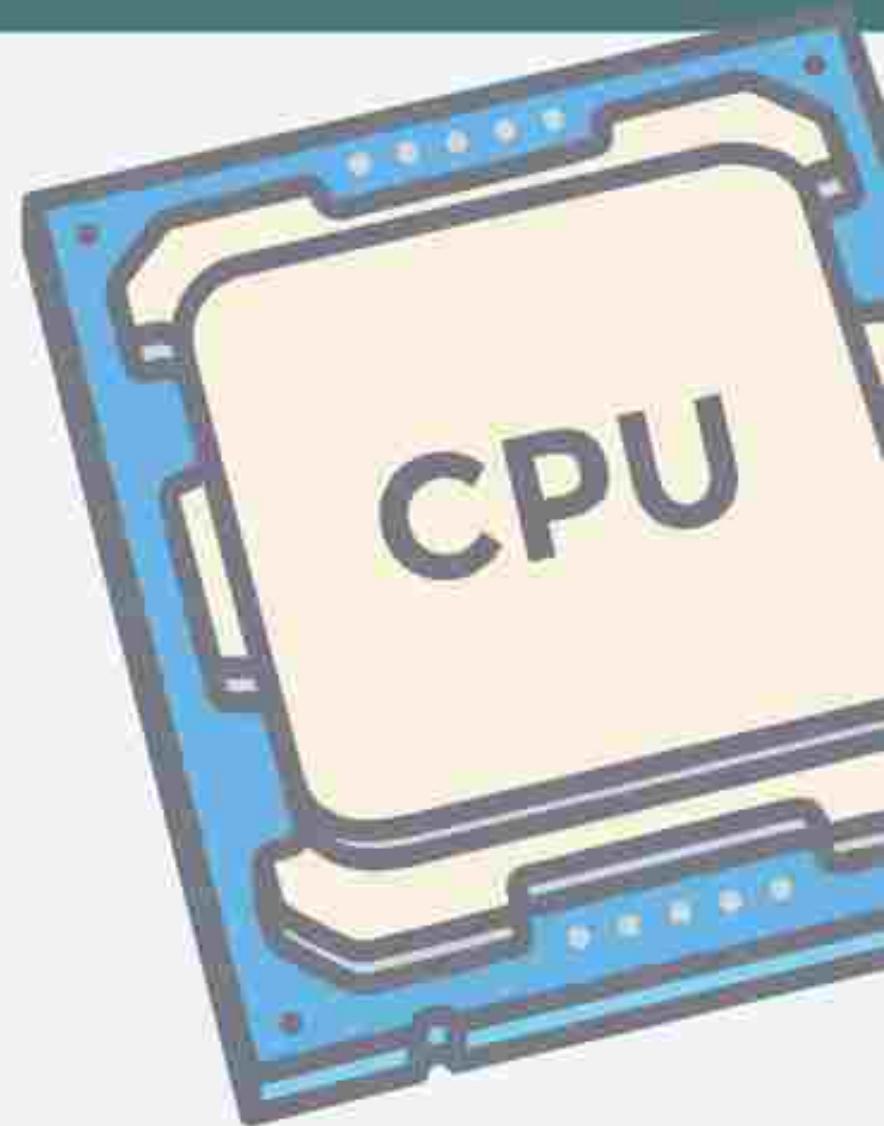
04. Usos

- 1. Servidores Virtuales:** Una sola computadora actúe como multiservidor, donde cada uno tiene su propio sistema operativo y aplicaciones.
- 2. Desarrollo y Pruebas de Software:** Los desarrolladores prueban su software en diferentes sistemas operativos sin necesidad de hardware adicional.
- 3. Continuidad de Negocios:** Permite la rápida restauración de servicios al facilitar la recuperación de fallos.
- 4. Aplicaciones Heredadas:** Permite la ejecución de aplicaciones antiguas en sistemas operativos que ya no son compatibles con el hardware moderno.

REQUERIMIENTOS

Estos sistemas operativos invitados creen que están ejecutándose en hardware real, cuando en realidad están aislados y gestionados por un hipervisor.

- **Compatibilidad:** Requiere que las máquinas virtuales actúen de la misma manera que el hardware real. El hipervisor debe proporcionar esta ilusión de manera eficiente.
- **Instrucciones Sensibles y Privilegiadas:** Popek y Goldberg definieron dos conjuntos de instrucciones relevantes: las instrucciones sensibles (solo se pueden ejecutar en modo kernel) y privilegiadas (generan una “trampa” si se ejecutan en modo usuario).
- **Tecnología:** Intel y AMD introdujeron en sus CPUs la tecnología para resolver los problemas de virtualización, la cual permite que las instrucciones sensibles se atrapen y emulen.
- **Evolución:** Se espera que la tecnología evolucione para permitir que los programas accedan al hardware directamente de manera segura, eliminando la necesidad de drivers en el hipervisor.



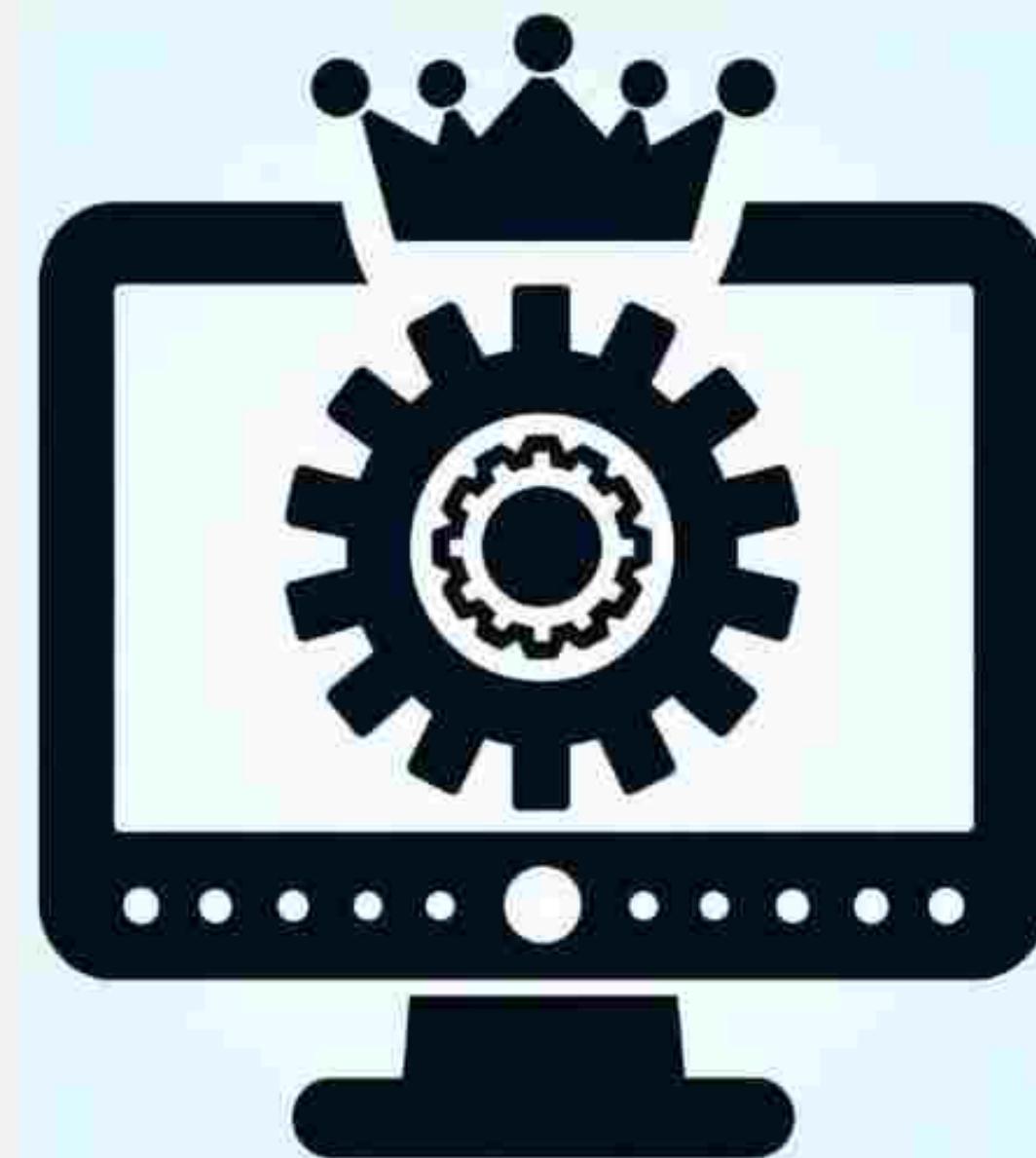
HIPERVISORES

TIPO 1

Conocidos como monitores de máquina virtual, se ejecutan directamente en el hardware y son responsables de gestionar de estas.

Se ejecutan en modo kernel y son el único software que se ejecuta en este modo, lo que les permite controlar directamente el hardware, así mismo deben ser compatibles para poder gestionar eficazmente las máquinas virtuales y las instrucciones sensibles.

Cuando el S.O invitado ejecuta una instrucción sensible, el hipervisor produce una interrupción para atraparla, de esta forma puede manejar la instrucción de manera adecuada.



TIPO 2

Son programas de usuario que se ejecutan encima de un S.O anfitrión y crean máquinas virtuales.

Estos se ejecutan como aplicaciones de usuario en un S.O anfitrión, (Windows o Linux).

Utiliza técnicas como la traducción binaria que reemplaza las instrucciones sensibles con llamadas que puede manejar, también emula las instrucciones sensibles que se intentan ejecutar, permitiendo que el S.O invitado funcione sin modificaciones.

PARAVIRTUALIZACION

Técnica que modifica el código fuente del S.O invitado para que haga llamadas directas al hipervisor en lugar de ejecutar instrucciones sensibles.

La paravirtualización puede mejorar el rendimiento al evitar la emulación de instrucciones sensibles y permitir que el S.O interactúe directamente con el hipervisor. Permite que un S.O paravirtualizado se ejecute en diferentes hipervisores, siempre que estos proporcionen la API adecuada.

Se sugiere que en el futuro, la API del hipervisor podría estandarizarse, lo que simplificaría el soporte y uso de la tecnología de máquinas virtuales



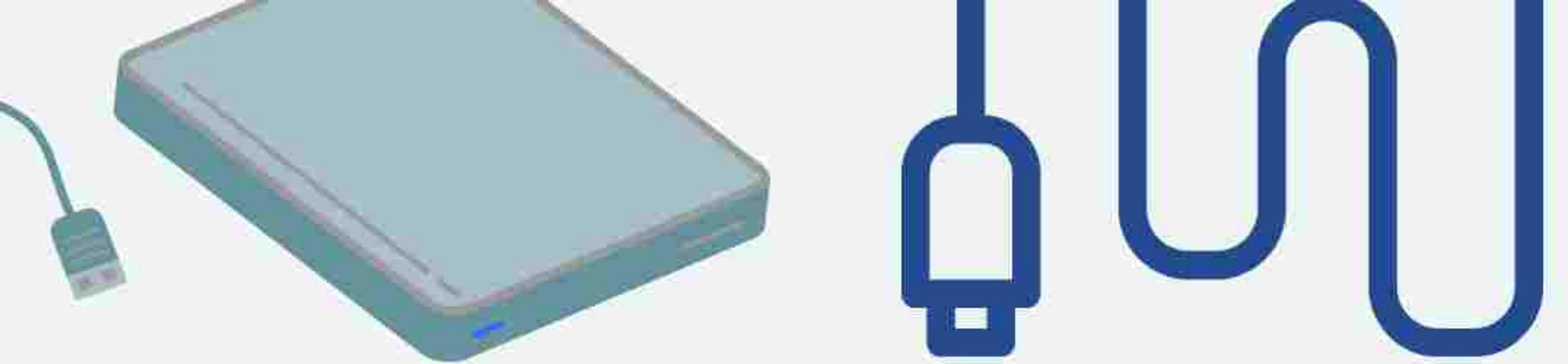


VIRTUALIZACIÓN DE LA MEMORIA

El hipervisor debe asignar páginas de memoria virtual a páginas de memoria física, lo cual es complejo cuando varias máquinas virtuales solicitan las mismas páginas físicas.

El hipervisor puede crear tablas de páginas ocultas para mapear las páginas virtuales de la M.V a las páginas físicas reales, requiriendo una gestión adicional cuando el S.O. cambia sus tablas de páginas, también puede marcar las tablas de páginas, como “solo lectura” para detectar cambios y actualizarlas en consecuencia.

En un S.O paravirtualizado, se notifica al hipervisor después de cambiar las tablas de páginas, lo que permite una gestión más eficiente.



VIRTUALIZACIÓN DE LA E/S

El hipervisor debe emular los dispositivos de E/S para que los S.O invitados puedan interactuar como si fuera el hardware real.

Asignar dispositivos virtuales a las M.V, implicando la creación de archivos/regiones en el disco para simular dispositivos de almacenamiento, así mismo traduce los comandos de E/S emitidos por los controladores en comandos que sean comprensibles para el hardware real.

Los de tipo 2 pueden aprovechar los drivers del S.O anfitrión, permitiendo simplificar la gestión de dispositivos de E/S en las M.V.



CONCEPTO

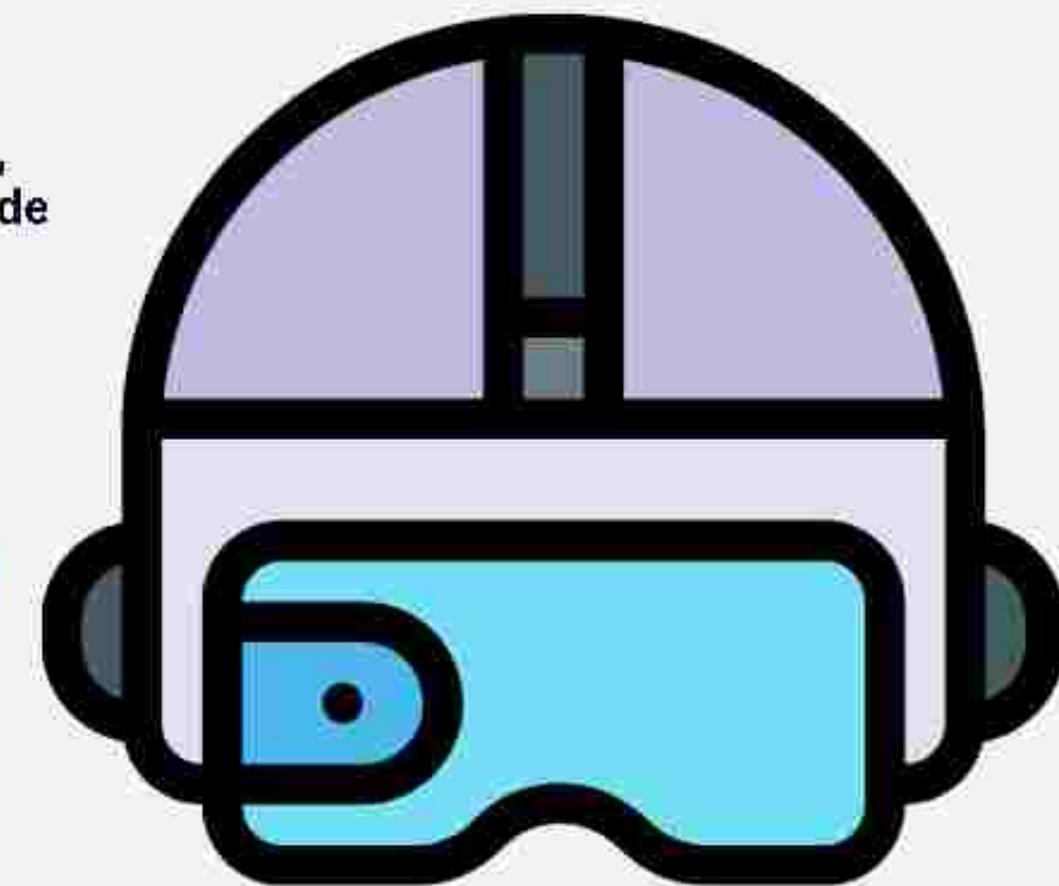
M.V preconfiguradas y listas para su uso, diseñadas para simplificar la instalación de nuevos programas.

DEPENDENCIA DEL SOFTWARE

Solución a las complejas dependencias de software, donde el desarrollador puede empaquetar cuidadosamente un S.O., compiladores, bibliotecas y código de aplicación.

DISTRIBUCION

Cambian la forma en que son distribuidas, ya que permite que los desarrolladores proporcionen un entorno completo y funcional.



DISPOSITIVOS VIRTUALES

INDEPENDENCIA DEL ANFITRION

Permite que los clientes ejecuten el software sin preocuparse por las dependencias con el S.O anfitrión o con otros software instalados.

FACILIDAD DE USO

Paquete completo que funciona sin necesidad de conocer todas las dependencias, facilitando la instalación y el uso de las aplicaciones.

PORTABILIDAD

Son portátiles y pueden ejecutarse en diferentes plataformas mientras el hipervisor sea compatible.

MAQUINAS VIRTUALES EN MULTINUCLEOS



CONFIGURACION

La combinación de CPU's de multinúcleo con M.V, ya que permite configurar un sistema como multicomputadora o multiprocesador virtual según las necesidades.



RECURSOS

Configurar una CPU de escritorio con múltiples núcleos, permite que actúe como una multicomputadora de varios nodos, lo cual aprovecha al máximo los recursos disponibles.



MEMORIA

Surge la posibilidad de que las máquinas virtuales puedan compartir memoria, convirtiendo una sola computadora en un multiprocesador virtual.



POTENCIAL

Los diseñadores de aplicaciones, les permite elegir el número de CPU's que necesitan y diseñar el software en consecuencia.

CUESTIONES SOBRE LICENCIAS



Las implicaciones de las licencias de software en la virtualización, las licencias por CPU pueden complicarse por el uso de máquinas virtuales.

- **Licencias por CPU:** Muchas licencias de software son por CPU, en otras palabras, significa que el software solo puede ejecutarse en una CPU.
- **Licencias por Uso Concurrente:** Son empresas con licencias que permiten un número fijo de máquinas ejecutando el software al mismo tiempo, enfrentando desafíos donde la demanda de máquinas virtuales puede fluctuar.
- **Restricciones en Licencias:** Algunos distribuidores incluyen cláusulas en las licencias que prohíben explícitamente la ejecución del software en máquinas virtuales no autorizadas.
 - **Validez legal:** Se checa si las restricciones serán válidas en los tribunales y cómo responderán los usuarios a ellas.