Resumen para la Prueba Final de Seguridad Ética – Informática

Objetivos

1. Identificar los conceptos y principios fundamentales relacionados con la seguridad informática.
2. Caracterizar los ataques y amenazas a las que se expone la información digital.
3. Identificar los problemas respecto a la seguridad de la información digital a partir del análisis de casos de estudio teniendo en cuenta los valores y principios éticos a cumplir.
4. Caracterizar los sistemas criptográficos simétricos y asimétricos, explicando sus ventajas y desventajas.
5. Caracterizar sistemas de cifrados de firma digital y función resumen a partir de los elementos fundamentales que estos deben cumplir.
6. Identificar los algoritmos, estándares y aplicaciones de la criptografía que se pueden aplicar ante determinado caso para preservar los aspectos de la seguridad de la información.
7. Evaluar el uso de la firma digital, función resumen, PKI y otras aplicaciones de la criptografía a partir de situaciones presentadas.
8. Explicar los elementos a tener en cuenta en las diferentes fases del proceso de desarrollo de software seguro.
9. Identificar principios de la programación que han sido violados y errores cometidos en el ciclo de desarrollo de software que provocan fallas de seguridad en el mismo.
10. Proponer soluciones a las vulnerabilidades del código fuente.
11. Proponer arquitecturas de cortafuegos adecuadas segun características de la red y sistema que se quiere proteger.
12. Corregir scripts con reglas iptables para la configuración de la seguridad perimetral.
13. Caracterizar el funcionamiento de los cortafuegos segun el nivel del modelo de referencia OSI en el que trabajan.
14. Seleccionar el sistema de detección de intrusos adecuado a partir de su clasificación y características, teniendo en cuenta el entorno donde se quiera emplear así como las funciones y limitaciones que este tipo de sistema tiene.
15. Caracterizar los sistemas de detección de intrusiones segun la fuente de datos, el motor de análisis y el tipo de respuesta.
16. Caracterizar los casos especiales de sistemas de detección de intrusiones.
17. Explicar los aspectos teóricos básicos para la definición de un plan de contingencia.
18. Planificar la salva y recuperación de datos a partir de la aplicación de principios y buenas prácticas para su implementación.
19. Seleccionar adecuadamente los tipos de copias a emplear segun el volúmen de datos y la frecuencia de actualización de los mismos.
20. Explicar las definiciones y las buenas prácticas para aplicar los pasos para la investigación forense para el esclarecimiento de los hechos en medios informáticos teniendo en cuenta aspectos éticos, características de la evidencia digital, el orden de volatilidad.
21. Caracterizar los pasos básicos de la investigación forense.

Primer Trabajo de Control Parcial

1-Identificar los conceptos y principios fundamentales relacionados con la seguridad informática.

Los bienes informáticos: lo constituyen los activos informáticos de tipo Hardware, Software y Datos que son los componentes fundamentales de un sistema de computación. Los datos son generalmente el activo informático más preciado para cualquier institución. El hardware y el software pueden ser caros pero fáciles de reponer, en cambio los datos o la información contienen la vida de una institución y su valor a veces es incalculable.

Seguridad Informática: es el conjunto de métodos y herramientas destinados a proteger los bienes (o activos) informáticos de una institución. La información es el activo más preciado y la seguridad en la información tiene el objetivo de garantizar:

-**Confidencialidad**: la información o los activos informáticos son accedidos solo por las personas autorizadas para hacerlo.

-**Integridad**: los activos o la información solo pueden ser modificados por las personas autorizadas y de la forma autorizada.

-**Disponibilidad**: los activos informáticos son accedidos por las personas autorizadas en el momento requerido.

Se considera un **incidente** de seguridad:

1. Un evento adverso en un entorno informático, que puede comprometer o compromete la confidencialidad, integridad o disponibilidad de la información.

2. Una violación o inminente amenaza de violación de una política de seguridad de la información.

Una **vulnerabilidad** es una debilidad en el sistema de seguridad que puede ser explotada para causar algún daño.

Una **amenaza** es un grupo de circunstancias que tienen el potencial para causar algún daño o pérdida. Existe hoy día un gran número de amenazas que pueden afectar la seguridad del equipo. A medida que se desarrollan las tecnologías informáticas y las herramientas de comunicaciones, los piratas disponen de posibilidades mayores para la propagación de amenazas.

Un **control** es una acción, dispositivo o procedimiento que elimina o reduce una vulnerabilidad. De manera general, se puede describir la relación entre amenaza, vulnerabilidad y control de la siguiente manera: **una amenaza puede ser bloqueada aplicando un control a una vulnerabilidad.**

Principios Básicos de Seguridad Informática:

- Mínimo privilegio: se deben otorgar los permisos estrictamente necesarios para efectuar las acciones que se requieran, ni más ni menos de lo solicitado.
- Eslabón más débil: la seguridad de un sistema es tan fuerte como su parte más débil. Un atacante primero analiza cual es el punto más débil del sistema y concentra sus esfuerzos en ese lugar.
- Proporcionalidad: las medidas de seguridad deben estar en correspondencia con lo que se protege y con el nivel de riesgo existente. No sería lógico proteger con múltiples recursos un activo informático que no posee valor o que la probabilidad de ocurrencia de un ataque sobre el mismo es muy baja.
- Dinamismo: la seguridad informática no es un producto, es un proceso. No se termina con la implementación de los medios tecnológicos, se requiere permanentemente monitoreo y mantenimiento.
- Participación universal: la gestión de la seguridad informática necesita de la participación de todo el personal de una institución. La seguridad que puede ser alcanzada mediante medios técnicos es

limitada y debiera ser apoyada por una gestión y procedimientos adecuados, que involucren a todos los individuos.

Las fuentes de amenazas pueden dividirse en tres grupos:

- El factor humano: Este grupo de amenazas incluye las acciones de personas que disponen (o no) de acceso autorizado a la información. Las amenazas de este grupo se subdividen en:

  Externas, que incluyen a los cibercriminales, los piratas, las estafas por Internet, los colaboradores sin escrúpulos y las organizaciones criminales.

  Internas, donde se incluyen las actuaciones del personal de la organización y los usuarios de PC personales. Las acciones tomadas por este grupo pueden ser deliberadas o accidentales.

- El factor tecnológico**:** Este grupo de amenazas se relaciona con problemas de orden técnico, por ejemplo, equipos que se vuelven obsoletos, software y hardware de mala calidad a la hora de procesar información. Todo ello conduce a fallos en los equipos y, a menudo, a pérdidas de datos.
- Los desastres naturales. Este grupo de amenazas incluye cualquier número de eventos de origen natural o independiente de la actividad humana.

2-Caracterizar los ataques y amenazas a las que se expone la información digital

Tipos de amenazas

- Intercepción: Acceso a la información por personas no autorizadas. Cuando se materializa la amenaza estamos en presencia de un ataque contra la confidencialidad de la información. Ejemplo: Robo de contraseñas o la copia ilícita de programas.
- Modificación: Acceso no autorizado a la información en el que se produce una modificación de la misma. Al materializarse la amenaza nos encontramos ante un ataque contra la integridad de los datos. Ejemplo: La desfiguración de un sitio web es un ejemplo de este tipo de ataque (defacement).
- Interrupción: Deja de funcionar total o parcialmente un sistema informático. Cuando esta amenaza se materializa estamos ante un ataque contra la disponibilidad de la información. Ejemplo: El bloqueo de los servicios de servidores web o de correo electrónico.
- Fabricación: Se crean objetos falsificados en un sistema de cómputo. El intruso puede insertar operaciones no esenciales de un sistema de comunicación de red o añadir registros a una base de datos existente. Este es un ataque contra la autenticidad. Ejemplo: Inserción de mensajes espurios en una red o añadir registros a un archivo.

Tipos de ataques:

- Pasivos**:** El atacante no altera la comunicación únicamente la escucha o monitorea para obtener la información que está siendo transmitida.

Clasificacion :

- Monitorización
- Ingeniería social.

Activo**:** Implican algún tipo de modificación del flujo de datos transmitidos o la creación de un falso flujo de datos.

Clasificacion

- Suplantación de identidad (robo de pass)
- Re-actuación(no verificación de los datos)

- Modificación de mansajes
- Degradación de los servicios(interrupción de los servicios)
- Elevación de privilegios

Los mecanismos de defensa de manera general siguen uno de los objetivos siguientes:

- Prevención: aumentar la seguridad del sistema previniendo la ocurrencia de violaciones a la seguridad. La utilización de un muro de seguridad o firewall es un ejemplo de este tipo de control.
- Detección: detectar la ocurrencia de una violación a la seguridad en el momento en que se produce la misma. Un ejemplo de este mecanismo de defensa lo constituyen los sistemas detectores de intrusos.
- Recuperación: retornar el sistema a su normal funcionamiento después de una violación. Las copias de respaldo o backups son el ejemplo más ilustrativo.

> Los ingenieros de software debieran obligarse a hacer del análisis, especificación, diseño, desarrollo, pruebas y mantenimiento del software una profesión respetada y beneficiosa. En concordancia con la obligación con el bienestar, salud y seguridad de la sociedad, los ingenieros del software debieran adherirse a los Ocho Principios siguientes:

2 Identificar los problemas respecto a la seguridad de la información digital a partir del análisis de casos de estudio teniendo en cuenta los valores y principios éticos a cumplir.

1. Sociedad: Los ingenierios de software actuarán de manera coherente con el interés social.

2. Cliente y Empresario: los ingenieros de software actuarán de manera que produzca el mejor resultado para cliente y empresario, y de manera coherente con el interés social.

3. Producto: los ingenieros de software garantizarán que sus productos y las modificaciones correspondientes cumplen los mayores estándares profesionales posibles.

4. Valoración: los ingenieros de software mantendrán la integridad e independencia en sus valoraciones profesionales.

5. Gestión: los líderes y gestores de ingeniería de software suscribirán y promoverán un enfoque ético en la gestión del desarrollo y mantenimiento del software.

6. Profesión: los ingenieros de software avanzarán en la integridad y reputación de la profesión, de manerar consistente con el interés social.

7. Compañeros: los ingenieros del software serán justos y apoyarán a sus compañeros.

8. Personal: los ingenieros del software participarán en el aprendizaje continuo referente a la práctica de su profesión y promoverán un enfoque ético en la práctica de la profesión.

Valores

Responsabilidad-disciplina
Capacidad-Cortesía
Horades-Iniciativa
Honestidad-Colaboración
Justicia-Precisión
Prudencia-Entusiasmo
Superación-Vocación

4- Caracterizar los sistemas criptográficos simétricos y asimétricos, explicando sus ventajas y desventajas.

Criptografía simétrica

Incluye los sistemas clásicos, y se caracteriza porque en ellos se usa la misma clave para cifrar y para descifrar, motivo por el que se denomina simétrica.

Para que un algoritmo de este tipo sea considerado fiable debe cumplir varios requisitos básicos:

- conocido el criptograma (texto cifrado) no se pueden obtener de él ni el texto en claro ni la clave.
- conocidos el texto en claro y el texto cifrado debe resultar más caro en tiempo o dinero descifrar la clave que el valor posible de la información obtenida por terceros.

Ventajas

- Los algoritmos simétricos descifran bloques de texto del documento original, y son más sencillos que los sistemas de clave pública, por lo que sus procesos de cifrado y descifrado son más rápidos.

Desventajas

- Las principales desventajas de los métodos simétricos son la distribución de las claves, el peligro de que muchas personas deban conocer una misma clave y la dificultad de almacenar y proteger muchas claves diferentes.

Criptografía asimétrica o de clave pública:

Se basa en el uso de dos claves diferentes, claves que poseen una propiedad fundamental: una clave puede descifrar lo que la otra ha encriptado.

Las claves pública y privada tienen características matemáticas especiales, de tal forma que se generan siempre a la vez, por parejas, estando cada una de ellas ligada intrínsecamente a la otra. Los algoritmos asimétricos están basados en funciones matemáticas fáciles de resolver en un sentido, pero muy complicadas de realizar en sentido inverso, salvo que se conozca la clave privada, como la potencia y el logaritmo. Ambas claves, pública y privada, están relacionadas matemáticamente, pero esta relación debe ser lo suficientemente compleja como para que resulte muy difícil obtener una a partir de la otra.

Para que un algoritmo de clave pública sea considerado seguro debe cumplir:

- conocido el texto cifrado no debe ser posible encontrar el texto en claro ni la clave privada.
- conocido el texto cifrado (criptograma) y el texto en claro debe resultar más caro en tiempo o dinero descifrar la clave que el valor posible de la información obtenida por terceros.
- conocida la clave pública y el texto en claro no se puede generar un criptograma correcto cifrado con la clave privada.
- dado un texto cifrado con una clave privada sólo existe una pública capaz de desencriptarlo, y viceversa.

Ventajas

- La principal ventaja de los sistemas de clave pública frente a los simétricos es que la clave pública y el algoritmo de cifrado son o pueden ser de dominio público y que no es necesario poner en peligro la clave privada en tránsito por los medios inseguros, ya que ésta está siempre oculta y en poder únicamente de su propietario.

Desventajas

- Los sistemas de clave pública dificultan la implementación del sistema y son mucho más lentos que los simétricos.

Problemas que resuelve la criptografía:

- Confidencialidad: se consigue porque solamente el receptor autorizado conocerá la clave con la que ha sido cifrado el mensaje; otro cualquiera al no conocerla no podría interpretar el mensaje.

- Integridad: se obtiene ya que solamente el emisor autorizado conoce la clave con lo que solo él podrá haber sido el que haya cifrado el criptograma, el receptor puede estar tranquilo y saber que no se trata de un impostor.
- Autenticidad: se refiere a que se pueda confirmar que el mensaje recibido haya sido mandado por quien dice lo mando o que el mensaje recibido es el que se esperaba.
- No rechazo: se refiere a que no se pueda negar la autoría de un mensaje enviado.

Algoritmos:

- Algoritmo DES

Algoritmo DES. (Estándar de encriptación de datos): DES (Data Encryption Standard)

1. Es un esquema de encriptación simétrico desarrollado en 1977 por el Departamento de Comercio y la Oficina Nacional de Estándares de EEUU en colaboración con la empresa IBM
2. Se creó con el objetivo de proporcionar al público en general un algoritmo de cifrado normalizado para redes de ordenadores.
3. DES utiliza una clave simétrica de bloques de 64 bits, de los cuales 56 son usados para la encriptación, mientras que los 8 restantes son de paridad, y se usan para la detección de errores en el proceso.

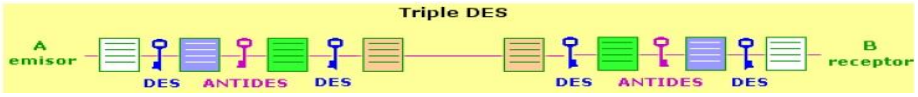
Ventajas:

- Es el sistema más extendido del mundo, el que más máquinas usan, el más barato y el más probado.
- Es muy rápido y fácil de implementar.
- Desde su aparición nunca ha sido roto con un sistema práctico.

Actualmente DES ya no es estándar y fue roto en Enero de 1999 con un poder de cómputo que efectuaba aproximadamente 250 mil millones de ensayos en un segundo.

- Algoritmo TripleDES

Algoritmo Triple DES: Como hemos visto, el sistema DES se considera en la actualidad poco práctico, debido a la corta longitud de su clave. Para solventar este problema y continuar utilizando DES se creó el sistema Triple DES (TDES), basado en tres iteraciones sucesivas del algoritmo DES, con lo que se consigue una longitud de clave de 128 bits, y que es compatible con DES simple. Este hecho se basa en que DES tiene la característica matemática de no ser un grupo, lo que implica que si se cifra el mismo bloque dos veces con dos llaves diferentes se aumenta el tamaño efectivo de la llave. Para implementarlo, se toma una clave de 128 bits y se divide en 2 diferentes de 64 bits, aplicándose el siguiente proceso al documento en claro:

- 

- 1. Se le aplica al documento a cifrar un primer cifrado mediante la primera clave, C1.
- 2. Al resultado (denominado ANTIDES) se le aplica un segundo cifrado con la segunda clave, C2.
- 3. Y al resultado se le vuelve a aplicar un tercer cifrado con la primera clave, C1.
- Actualmente TDES usa 3 claves diferentes, lo que hace el sistema mucho más robusto que el DES, al conseguirse longitudes de clave de 192 bits (de los cuales son efectivos 168).

- Algoritmo AES (algoritmo avanzado de encriptado)

Algoritmo AES- (Algoritmo avanzado de encriptación): Advanced Encryption Standard (AES),

1. También conocido como Rijndael.
2. Es un esquema de cifrado por bloques adoptado como un estándar de cifrado por el gobierno de los Estados Unidos.
3. es un sistema de cifrado por bloques, diseñado para manejar longitudes de clave y de bloque variables, ambas comprendidas entre los 128 y los 256 bits. Opera en una matriz de 4×4 bytes.
4. A partir de ésta base se realiza una serie de bucles de cifrado, cada uno de ellos consistente en las siguientes operaciones:

1. Sustitución de bytes no lineal, operando independientemente sobre cada uno de los bytes del Estado.
2. Desplazamiento de las filas del Estado cíclicamente con offsets diferentes.
3. Mezcla de columnas, que se realiza multiplicando las columnas del Estado módulo x4+1, consideradas como polinomios en GF(28), por un polinomio fijo c(x).
4. Adición de la clave de vuelta, en la que se aplica al Estado por medio de un simple XOR. La clave de cada vuelta se deriva de la clave de cifrado mediante el esquema de clave.

El esquema de clave consiste en dos operaciones, expansión de clave y selección de clave de vuelta de cifrado, y el proceso de cifrado consta de tres pasos: una adición inicial de la clave de vuelta, n-1 vuelta de cifrado y una vuelta final. Como plataforma se especificó el procesador Pentium de Intel (un procesador de 32 bits) y también procesadores de 8 bits (sin especificar alguno en particular) como los que se utilizan en tarjetas inteligentes (smart cards).

Ventajas:

- AES es rápido tanto en software como en hardware.
- Es relativamente fácil de implementar, y requiere poca memoria.

- **IDEA**: Utiliza una clave de 128 bits, lo que por el momento lo hace inmune a los ataques de fuerza bruta así como al criptoanálisis diferencial para su descifrado. Trabaja con bloques de texto de 64 bits. El algoritmo de descifrado es muy parecido al de encriptación, por lo que resulta muy fácil y rápido de programar. No está sometido a restricciones o permisos nacionales, por lo que se ha difundido ampliamente.

Ventajas:

- El espacio de claves es mucho más grande.
- Todas las operaciones son algebraicas.
- No hay operaciones a nivel bit, facilitando su programación en alto nivel.
- Es más eficiente que los algoritmos de tipo Fuiste, porque a cada vuelta se modifican todos los bits de bloque y no solamente la mitad.
- Se pueden utilizar todos los modos de operación definidos para el DES.

- **RC5**: Tiene tamaño variable de bloques (32, 64 o 128 bits), con tamaño de clave (entre 0 y 2040 bits, 128 sugerido) y número de vueltas (entre 0 y 255). La estructura general del algoritmo es una red tipo Feistel. Su funcionamiento se basa en las rutinas de cifrado y descifrado pueden ser especificadas en pocas líneas de código, pero la programación de claves es más complicada. Una de sus principales características es Adecuado para ser implementado en hardware o software, Rápido(Las operaciones básicas se trabajan en palabra completas simultáneamente), Adaptable a procesadores de diferentes tamaños de palabras, Número variable de iteraciones, Clave de longitud variable, Rotaciones con dependencia de datos, Sencillo, Alta seguridad y Bajo consumo de memoria.

Ventaja: Fácil de implementar y analizar, Adaptable a cualquier aplicación, Difícil de descifrar, Seguro.

Desventaja: Distribución de las claves, Dificultad de almacenar y proteger diversas claves diferentes.

Algoritmos asimétricos

Algoritmo RSA

Es un criptosistema de llaves públicas. Las claves pública y privada se calculan a partir de un número que se obtiene como producto de dos primos grandes. El atacante se enfrentará, si quiere recuperar un texto claro a partir del criptograma y la llave pública, a un problema de factorización.

1. Es el algoritmo asimétrico más sencillo de comprender e implementar.
2. sus claves sirven indistintamente tanto para codificar como para autentificar.

3. las primeras versiones de PGP lo incorporaban como método de cifrado y firma digital, pero se desaconsejó su uso a partir de la versión 5 en favor de otros algoritmos.
4. Se basa en la dificultad para factorizar grandes números. Las claves pública y privada se calculan a partir de un número que se obtiene como producto de dos primos grandes.
5. se considera segura una clave RSA con una longitud de n de al menos 768 bits.

Diffie-Hellman:

Se emplea fundamentalmente para acordar una clave común entre dos interlocutores, a través de un canal de comunicación inseguro. La ventaja de este sistema es que no son necesarias llaves públicas en el sentido estricto, sino una información compartida por los dos comunicantes.

1. Fue el punto de partida para los sistemas asimétricos, basados en dos claves diferentes, la pública y la privada.
2. Matemáticamente se basa en las potencias de los números y en la función mod (módulo discreto).
3. Permite el intercambio secreto de claves entre 2 partes que no han tenido previo contacto, utilizando un canal inseguro, y sin previa autenticación.
4. Su seguridad radica en la dificultad de calcular logaritmos discretos en un campo finito.
5. Su importancia se debe sobre todo al hecho de ser el inicio de los sistemas asimétricos, ya que en la práctica sólo es válido para el intercambio de claves simétricas, y con esta funcionalidad es muy usado en los diferentes sistemas seguros implementados en Internet, como SSL (Secure Socket Layer) y VPN (Virtual Private Network).

El Gamal:

Fue diseñado en un principio para producir firmas digitales, pero posteriormente se extendió también para codificar mensajes. Se basa en el problema de los logaritmos discretos, que está íntimamente relacionado con el de la factorización, y en el de Diffie-Hellman. Para generar un par de llaves, se escoge un número primo n y dos números aleatorios p y x menores que n.

1. Es usado en software libre en GNU Privancy Guard, versiones recientes del PGP y otros sistemas.
2. La seguridad se basa en la actual incapacidad computacional de solucionar el problema discreto del logaritmo.
3. Diseñado previamente para producir firmas digitales pero se extendió posteriormente para codificar mensajes. Se basa en el problema de los logaritmos discretos que está relacionado con la factorización de números enteros.

Rabin:

Se basa en el problema de calcular raíces cuadradas módulo de un número compuesto. Este problema se ha demostrado que es equivalente al de la factorización de dicho número.

Algunos protocolos que usan los algoritmos antes citados son:

DSS ("Digital Signature Standard") con el algoritmo DSA ("Digital Signature Algorithm")

- PGP se usa en el algoritmo ElGamal.
- SSH gestionar claves RSA
- SSL, ahora un estándar del IETF se usa:
  - Para criptografía de clave pública: RSA, Diffie-Hellman.
  - Para cifrado simétrico: RC2, RC4, IDEA (International Data Encryption Algorithm), DES (Data Encryption Standard), Triple DES o AES
- TLS emplea el intercambio de claves de Diffie-Hellman.

5- Caracterizar sistemas de cifrados de firma digital y función resumen a partir de los elementos fundamentales que estos deben cumplir.

PKI *puede definirse como el conjunto de componentes y políticas necesarias para crear, gestionar y revocar* ***certificados digitales*** *que pueden ser utilizados para autenticar cualquier aplicación, proceso u*

organización. El principal propósito de una PKI es la protección de datos sensibles a través de técnicas de encriptación.

¿Cómo funciona una PKI?

Cada dispositivo de usuario final posee un software de encriptación y un par de claves: pública para distribuirla a otros usuarios y, otra privada, guardada y protegida por su propietario. Por ejemplo, si un usuario quiere enviar un correo electrónico a otro usuario, el usuario emisor cifra el mensaje utilizando la clave pública del receptor; cuando el mensaje se recibe, el receptor lo descifra con su clave privada.

Se pueden tener múltiples pares de claves para mantener comunicaciones distintas con grupos diferentes. Por tal motivo, dado el elevado número de claves que intervienen en las comunicaciones, resulta crucial contar con algún método para administrarlas y controlar su utilización. Aquí es donde una PKI entra en juego, permitiendo la creación, distribución, seguimiento y revocación centralizada de claves, siendo este el método de seguridad más completo que existe hoy en día.

Resumiendo, PKI se basa en identidades digitales conocidas como "certificados digitales", que actúan como "pasaportes electrónicos", y vinculan la firma digital del usuario a su clave pública.

Ventajas de la PKI

- *Las claves no viajan a través de la red desde el cliente al servidor, dado que los certificados constituyen información pública.*
- *Los certificados basados en tecnología de clave pública proveen un mecanismo de autenticación más fuerte. Sólo el usuario conoce la forma de acceder a su clave privada. Simplificación en la administración y disminución de costos.*

Desventajas de la PKI

- *La seguridad de un sistema basado en CA consiste en varios eslabones, algunos de los cuales no son criptográficos: la gente también forma parte de esta infraestructura.*
- *Un conflicto que permanece sin resolver es la seguridad de la clave privada con respecto al almacenamiento de la misma.*

***La firma digital** garantiza integridad mientras que el cifrado garantiza confidencialidad, indirectamente la criptografía asimétrica garantiza autenticidad pero es realmente el **certificado digital** quien garantiza la autenticidad del emisor ante el receptor.*

El uso de certificados digitales y de claves públicas y privadas para realizar informáticas hoy día solo tiene éxito cuando existe transparencia entre las aplicaciones y los mecanismos que PKI utiliza para garantizar la seguridad.

Certificado Digital

Un certificado digital es un documento que contiene diversos datos, entre ellos el nombre de un usuario y su clave pública, y que es firmado por una Autoridad de Certificación (AC).Como el emisor y receptor confiarán en esa AC, el usuario que tenga un certificado expedido por ella se autenticará ante el otro, en tanto que su clave pública está firmada por dicha autoridad. Una de las certificaciones más usadas y un estándar en la actualidad en infraestructuras de clave pública PKIs (Public-KeyInfrastructure) es X.509, el cual está basado en criptografía asimétrica y firma digital.

Firma Digital: El cifrado con clave pública permite generar firmas digitales que hacen posible certificar la procedencia de un mensaje, en otras palabras, asegurar que proviene de quien dice. ‘’ Autenticidad o No Repudio ‘

- La firma se puede aplicar a un mensaje completo o puede ser algo añadido al mensaje.
- Las firmas son especialmente útiles cuando la información debe atravesar redes sobre las que no se tiene control directo.

- La firma digital es una herramienta tecnológica que permite garantizar la autoría e integridad de los documentos digitales.
- Una firma digital es un conjunto de datos asociados a un mensaje digital que permite garantizar la identidad del firmante y la integridad del mensaje.
- La firma digital no implica asegurar la confidencialidad del mensaje; un documento firmado digitalmente puede ser visualizado por otras personas.

En la práctica, debido a que los algoritmos de clave pública son muy ineficaces a la hora de cifrar documentos largos, los protocolos de firma digital se implementan junto con funciones unidireccionales de resumen (hash), de manera que en vez de firmar un documento, se firma un resumen del mismo. Este mecanismo implica el cifrado, mediante la clave privada del emisor, del resumen de los datos, que serán transferidos junto con el mensaje.

De esta forma se ofrecen conjuntamente los servicios de no repudio, ya que nadie excepto A podría haber firmado el documento, y de autenticación, ya que si el documento viene firmado por A, podemos estar seguros de su identidad, dado que sólo él ha podido firmarlo. En último lugar, mediante la firma digital se garantiza asimismo la integridad del documento, ya que en caso de ser modificado, resultaría imposible hacerlo de forma tal que se generase la misma función de resumen que había sido firmada.

Función Resumen: (Hash) ‘’ Integridad o Contraseñas ‘’

Contraseña: La contraseña es una frase que el usuario elije y que se utiliza para proteger su clave privada. De otro modo, alguien podría robar la clave privada de otro usuario y usarla para descifrar sus mensajes. La contraseña es requerida cada vez que se va a utilizar la clave privada. Se utilizan para cifrar ficheros de manera convencional. La función resumen es la aplicación que tiene que ver con las contraseñas.

MD5: es más lento pero con mayor nivel de seguridad.

SHA-1: más seguro y resistente a ataques por fuerza bruta que el algoritmo MD5. Aunque es más lento que MD5, hoy es el estándar como función hash.

1. Explicar los elementos a tener en cuenta en las diferentes fases del proceso de desarrollo de software seguro

Fase de Diseño:

- **Definir los objetivos de seguridad del producto:** Es necesario determinar de forma temprana quién es el cliente objetivo y cuáles son sus requisitos de seguridad. Para definir objetivos de seguridad se debe tener en cuenta quién va a ser el usuario final de la aplicación, qué significa la seguridad para el cliente, si cambian los requisitos de seguridad en dependencia del cliente, dónde se ejecuta la aplicación, qué se está tratando de proteger, cuáles son las implicaciones para los usuarios si los objetos que está protegiendo están en peligro, etc.
- **Modelado de amenazas potenciales para un Diseño Seguro:** Los modelos de amenaza ayudan a formar la base de las especificaciones de diseño. Sin modelos de amenaza, no se puede construir sistemas seguros, porque los sistemas seguros lo necesitan para comprender sus amenazas.
- **Construir Planes de finalización en características inseguras:** Un Software nunca muere, sino que sólo se vuelve inseguro debido a que la industria encuentra nuevas vulnerabilidades. Debido a esto, es necesario tener al final de su vida los planes para cambiar las viejas funcionalidades. Los clientes por lo general no les gustan las sorpresas, y esto es una gran manera de ayudarles a prepararse para el cambio.
- **Configuración de la barra de error:** Hay que ser realista y pragmático cuando se determina cuáles errores corregir y cuáles no para arreglar antes del envío. La seguridad es una parte, aunque sea una parte muy importante, de las ventajas y desventajas que van en el diseño y desarrollo de una aplicación. Muchos otros criterios deben ser evaluados al momento de decidir el modo de reparar un defecto.
- **Revisión del equipo de seguridad:** Equipo externo que revisa el plan de amenazas desarrollado.

Fase de Desarrollo

- **Definir directrices de codificación segura:** Usted debe definir y evangelizar a un conjunto mínimo de directrices para el equipo de codificación. Informar a los desarrolladores de cómo se deben manejar los buffers, cómo se deben tratar los datos que no son de confianza, cómo deben cifrar los datos, y así sucesivamente.
- **Revise Defectos Viejos:** Debe aprender de los errores pasados para que no sigan cometiendo los mismos errores de seguridad.
- **Examen de la Seguridad Externa:** Vale la pena contar con una entidad externa, como una empresa de consultoría de seguridad, revisar su código y planes.
- **Sea consciente de sus cuentas de errores:** Usted encontrará los errores de seguridad si se centran en busca de ellos, pero asegúrese de que su número de errores no se convierta en inmanejable. Una regla utilizada por algunos grupos es permitir a los desarrolladores no tienen más de cinco errores activos a la vez
- **Lleve un registro de métricas de errores:** Cuando una falla de seguridad se encuentra en el diseño o en el código, debe iniciar una entrada en la base de datos de seguimiento de errores, como lo haría normalmente. Sin embargo, se debe añadir un campo extra a la base de datos para que pueda definir qué tipo de amenaza de seguridad plantea el error.

Fase de prueba

A los probadores se les deben enseñar cómo los atacantes operan y deben aprender las mismas técnicas de seguridad tales como desarrolladores. El papel de las pruebas de seguridad es verificar que el diseño del sistema y el código pueden resistir el ataque. La determinación de que las características funcionan como se anuncia es todavía una parte sumamente importante del proceso.

Despliegue y mantenimiento

- Proceso de Respuesta
- Responsabilidad de la persona que despliega.

9-Identificar principios de la programación que han sido violados y errores cometidos en el ciclo de desarrollo de software que provocan fallas de seguridad en el mismo.

Fallas en productos de software.

- Inyección de código SQL: Burlar la información a entrar, entrando comandos de código SQL para apoderarse de datos privados de la aplicación (datos no confiables).

Ocurre cuando datos no confiables son enviados a un intérprete como parte de un comando o consulta SQL, los datos hostiles del atacante pueden engañar al intérprete en ejecutar comandos no intencionados o acceder datos no autorizados.

Para prevenir la inyección se requiere mantener los datos no confiables separados de comandos y consultas. Una de las opciones sería utilizar una API segura que evite el uso del intérprete o provea una interface parametrizada. Si una API parametrizada no se encuentra disponible, se deben escapar los n caracteres especiales utilizando una sintaxis de escape especial para dicho intérprete.

- Desbordamiento de buffer: se produce cuando un programa no controla adecuadamente la cantidad de datos que se copian sobre un área de memoria reservada a tal efecto (buffer), de forma que si dicha cantidad es superior a la capacidad preasignada los bytes sobrantes se almacenan en zonas de memoria adyacentes, sobrescribiendo su contenido original.

Para prevenir el buffer overflow se necesita controlar de manera adecuada la cantidad de datos que se copian en un área de memoria reservada.

- Secuencia de comandos en sitios cruzados: Las fallas XSS ocurren cada vez que una aplicación toma datos no confiables y los envía al navegador web sin una validación y codificación apropiada. XSS permite a los atacantes ejecutar secuencia de comandos en el navegador de la víctima los cuales pueden secuestrar las sesiones de usuario, destruir sitios web, o dirigir al usuario hacia un sitio malicioso.

Para prevenir XSS requiere mantener los datos no confiables separados del contenido activo del navegador. Una de las opciones es escapar todos los datos no confiables basados en el contexto HTML donde los mismos serán ubicados. Una validación de entrada positiva con apropiada

canonicalización y decodificación es también recomendable ya que ayuda a proteger contra XSS pero no es una defensa completa.

Proponer arquitecturas de cortafuegos adecuadas según características de la red y sistema que se quiere proteger

Arquitecturas de cortafuegos (topologías):

Filtrado de paquetes

Se trata de la arquitectura de cortafuegos más antigua, basada simplemente en aprovechar la capacidad de algunos routers para hacer un encaminamiento selectivo, es decir, para bloquear o permitir el tránsito de paquetes mediante listas de control de acceso en función de ciertas características de las tramas, de forma que el router actúe como pasarela de toda la red.

El principal problema es que no disponen de un sistema de monitorización sofisticado, por lo que muchas veces el administrador no puede determinar si el router está siendo atacado o si su seguridad ha sido comprometida.

Dual-homed Gateway (o de dos bases)

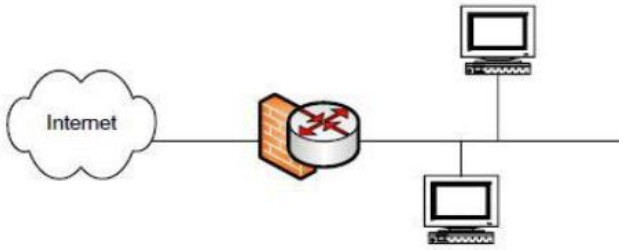


Figura 2. Arquitectura Filtrado de Paquetes.

Se trata de un host con dos tarjetas de red, cada una de ellas conectada a una red diferente. El sistema ha de ejecutar al menos una aplicación proxy para cada uno de los servicios que se desee pasar a través del cortafuego, y también es necesario deshabilitar la función de enrutamiento. La ventaja de estos sistemas es su sencillez, pues sólo requieren un ordenador. La desventaja es que sólo soportan servicios mediante proxy y no por filtrado de paquetes, ya que al tener la función de enrutamiento deshabilitada, se fuerza a que el tráfico deba ser tratado por una aplicación en el propio host.

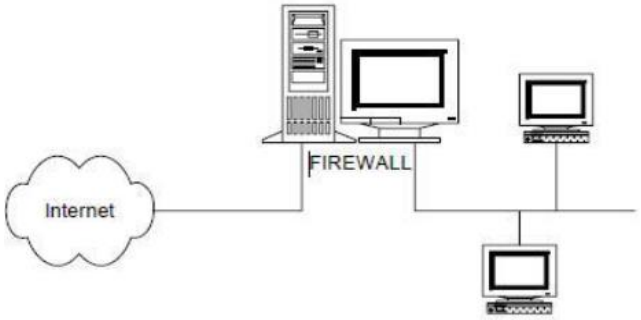
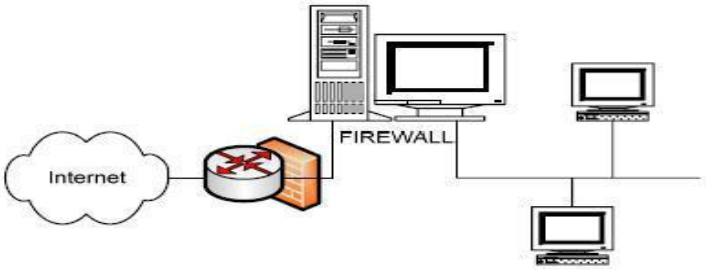


Fig 3. Arquitectura Dual-homed gateway.

Screened host

En este modelo la conexión entre las dos redes se produce mediante un router configurado para bloquear todo el tráfico entre la red externa y todos los hosts de la red interna, excepto un único bastión, donde se instala todo el software necesario para la implementación del firewall. Esta topología nos permite soportar servicios tanto mediante proxy (en el bastión) como mediante filtro de paquetes (en el router). El problema de esta topología es que no hay nada previsto a nivel de seguridad entre el bastión y el resto de hosts internos, de modo que si un atacante logra entrar en el bastión, puede atacar la red interna, al igual que pueden producirse ataques internos hacia el host bastión.

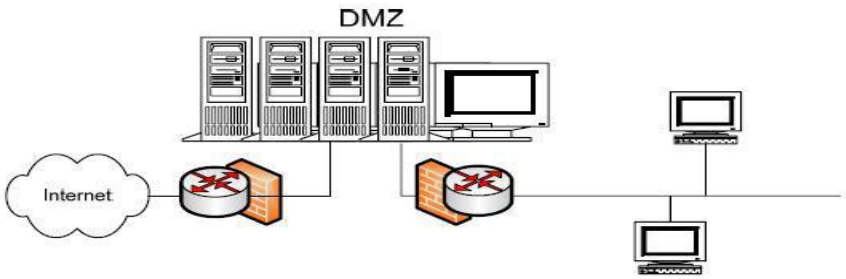


Esta arquitectura es un paso más en términos de seguridad de los cortafuegos al combinar un router con un host bastión, el principal nivel de seguridad proviene del filtrado de paquetes, es decir, el router es la primera y más importante línea de defensa.

En la máquina bastión, único sistema accesible desde el exterior, se ejecutan los proxies de las aplicaciones, mientras que el router se encarga de filtrar los paquetes que se puedan considerar peligrosos para la seguridad de la red interna, permitiendo únicamente la comunicación con un reducido número de servicios.

Screened subnet
En este modelo se sitúa una red entre las dos redes a conectar. A ésta red se le conoce como red perímetro o zona desmilitarizada (DMZ), y se conecta a las otras dos mediante sendos routers. La DMZ añade un nivel de seguridad en las arquitecturas de cortafuegos, de forma que se consiguen reducir los efectos de un ataque exitoso al host bastión.

Como la máquina bastión es un objetivo interesante para muchos piratas, la arquitectura DMZ intenta aislarla en una red perimétrica de forma que un intruso que accede a esta máquina no consiga un acceso total a la subred protegida.



Los routers se configuran, mediante reglas de filtrado, para que tanto los nodos de la red interna como los de la externa, sólo puedan comunicarse con máquinas de la red perímetro. Esto permite a la red interna ser efectivamente invisible a la externa. Si un atacante lograra entrar a alguno de los bastiones de la red perímetro, aún estaría el router interno protegiendo las máquinas de nuestra red privada. Screened subnet es la arquitectura más segura, pero también la más compleja; se utilizan dos routers, denominados exterior e interior, conectados ambos a la red perimétrica.

El router exterior tiene como misión bloquear el tráfico no deseado en ambos sentidos (hacia la red perimétrica y hacia la red externa), mientras que el interior hace lo mismo pero con el tráfico entre la red interna y la perimétrica. De esta forma, un atacante necesitaría romper la seguridad de ambos routers para acceder a la red protegida.

Un cortafuego o muro de seguridad, es un sistema que hace cumplir una política de control de acceso entre dos redes. Se utiliza para separar, en cuanto a seguridad se refiere, una máquina o subred del resto, protegiéndola así de servicios y protocolos que puedan suponer una amenaza a la seguridad.

Es importante recalcar que un cortafuego es parte de una política de seguridad, que protege de un gran número de amenazas creando un perímetro de defensa en la red, pero solo es capaz de detectar y bloquear los ataques que pasen por él. Son necesarios además otros mecanismos de seguridad que se complementen en la defensa de los sistemas informáticos.

Corregir scripts con reglas iptables para la configuración de la seguridad perimetral.

Iptables:

Iptables [-t tabla] comando [comparación] [-j objetivo/salto]

Ejercicio 1. Haga un script para configurar el firewall iptables en un servidor que posee dos interfaces de red, una para la red LAN y la otra para el exterior. La dirección de la red LAN es *192.168.1.0/24* (clase C privada) y las direcciones IP de las interfaces de red del servidor son *eth0: 192.168.1.1* y *eth1: 200.55.134.15*. Las reglas del filtrado del firewall serían las siguientes:

- Permitir el acceso al servicio WEB (puerto 80) solo desde la LAN.
- Permitir el acceso al servicio Proxy (8080) solo desde la LAN.
- Permitir el acceso al servicio IMAP (143) solo desde la LAN.
- Permitir el acceso al SMTP (25) solo desde la LAN
- Permitir el acceso al SSH (22) solo desde de la IP 192.168.1.20 y desde la IP 213.195.64.45
- Permitir el acceso al FTP (20, 21) solo desde la IP 80.37.45.194
- Por último cerrar todos los puertos bien conocidos TCP y UDP (1 al 1024).

Debe desarrollar dos variantes para las políticas por defecto DENEGAR y ACEPTAR.
Respuesta:

 # Variante 1: Política por defecto DENEGAR
 # Eliminando todas las reglas existentes (FLUSH de reglas)
iptables –F
 # Estableciendo política por defecto DENEGAR
iptables -P INPUT DROP
iptables -P OUTPUT ACCEPT
iptables -P FORWARD DROP
 # El localhost siempre se deja abierto para las conexiones locales a diversos programas
iptables -A INPUT -i lo -j ACCEPT
 # Acceso al servicio WEB solo desde la LAN
iptables -A INPUT -i eth0 -s 192.168.1.0/24 -p TCP --dport 80 -j ACCEPT
 # Acceso al Proxy desde la LAN
iptables -A INPUT -i eth0 -s 192.168.1.0/24 -p TCP --dport 8080 -j ACCEPT
 # Acceso al IMAP desde la LAN
iptables -A INPUT -i eth0 -s 192.168.1.0/24 -p TCP --dport 143 -j ACCEPT
 # Acceso al SMTP desde la LAN
iptables -A INPUT -i eth0 -s 192.168.1.0/24 -p TCP --dport 25 -j ACCEPT
*# Acceso al SSH desde una PC de de la red loca*l
iptables -A INPUT -i eth0 -s 192.168.1.20 -p TCP --dport 22 -j ACCEPT
 # Acceso al SSH desde una dirección de la red externa
iptables -A INPUT -i eth1 -s 213.195.64.45 -p TCP --dport 22 -j ACCEPT
 # Acceso al FTP desde una PC
iptables -A INPUT -i eth0 -s 80.37.45.194 -p tcp --dport 20:21 -j ACCEPT
 # No es necesario cerrar explícitamente los puertos bien conocidos porque la política por defecto es
 # DENEGAR y le será aplicada a todos los paquetes que no cumplan con las condiciones anteriores

 # Variante 2: Política por defecto ACEPTAR
 # Eliminando todas las reglas existentes (FLUSH de reglas)
iptables –F
 # Estableciendo política por defecto ACEPTAR
iptables -P INPUT ACCEPT
iptables -P OUTPUT ACCEPT
iptables -P FORWARD ACCEPT
 # El localhost siempre se deja abierto para las conexiones locales a diversos programas
iptables -A INPUT -i lo -j ACCEPT
 # Acceso al acceso al servicio WEB solo desde la LAN, denegar cualquier otro rango IP
iptables -A INPUT -i eth0 -s ! 192.168.1.0/24 -p TCP --dport 80 -j DROP
 # Acceso al Proxy desde la LAN
iptables -A INPUT -i eth0 -s ! 192.168.1.0/24 -p TCP --dport 8080 –j DROP
 # Acceso al IMAP desde la LAN
iptables -A INPUT -i eth0 -s ! 192.168.1.0/24 -p TCP --dport 143 -j DROP
 # Acceso al SMTP desde la LAN
iptables -A INPUT -i eth0 -s ! 192.168.1.0/24 -p TCP --dport 25 -j DROP

```
# Acceso al SSH desde una PC de de la red local
iptables -A INPUT -i eth0 -s ! 192.168.1.20 -p TCP --dport 22 -j DROP
# Acceso al SSH desde una dirección de la red externa
iptables -A INPUT -i eth1 -s ! 213.195.64.45 -p TCP --dport 22 -j DROP
# Acceso al FTP desde una PC
iptables -A INPUT -i eth0 -s ! 80.37.45.194 -p tcp --dport 20:21 -j DROP
# Cerrando los puertos bien conocidos
iptables - A INPUT -- dport 1:1024 - j DROP
```

Caracterizar el funcionamiento de los cortafuegos según el nivel del modelo de referencia OSI en el que trabajan.

Tipos de *Firewalls*

Los cortafuegos pueden dividirse en cuatro grandes categorías, basados en el nivel en que lleva a cabo las acciones de filtrado:

1. Filtrado a nivel de Paquetes.
    a) Estático
    b) Dinámico
    c) Completo (*stateful*)
2. Filtrado a nivel de Circuito.
3. Filtrado a nivel de Aplicación.
4. Inspección de paquetes completa (*Stateful Packet Inspection*).

Cortafuegos de capa de red o filtro de paquetes

Trabajaban en el nivel de red del modelo OSI (o el nivel de IP de TCP/IP). Usualmente son parte de un router. En estos cortafuegos, se examina cada paquete, comparándolo con un grupo de reglas o listas de acceso (ACLs) y en dependencia de estas se puede aceptar el paquete, bloquearlo o enviarle un mensaje al que lo originó. Las reglas pueden incluir las direcciones IP de origen y destino, puertos de origen y destino y los protocolos del nivel de transporte usados. La mayoría de los equipos de interconexión (routers y switches) soportan el filtrado de paquetes. Incluso si se usan otros tipos de cortafuegos, implementar el filtrado de paquetes a nivel de router agrega un grado inicial de seguridad a un nivel mas bajo. Un cortafuego de filtrado a nivel de red puede utilizar alguna de las siguientes tecnologías:

- Filtrado estático de paquetes: reglas establecidas manualmente y los puertos permanecen abiertos o cerrados hasta que se cambie manualmente.
- Filtrado dinámico de paquetes: más inteligente que el anterior, en que las reglas se cambian dinámicamente dependiendo de sucesos o condiciones, y los puertos se mantienen abiertos mientras es necesario y luego se cierran.
- Filtrado a nivel de datos o estado (stateful): se usa una tabla para mantener los estados de conexión de sesiones como el orden en que los paquetes deben pasar en una secuencia autorizada por las políticas de filtrado.

Entre las ventajas de los *firewalls* de nivel de red están que son económicos, muy rápidos y con un bajo impacto en el rendimiento de la red. Sus desventajas vienen dadas porque es una tarea muy compleja definir las reglas de filtrado, sobre todo en entornos muy heterogéneos, y una vez definidas son muy laboriosas de comprobar. Sus capacidades de auditoria y registro de actividades suelen ser limitadas y no soportan políticas de seguridad complejas como autentificación de usuarios y control de accesos con horarios prefijados.

Cortafuegos de filtrado a nivel de Circuito (*Circuit Level Gateway*)

Un cortafuego de este tipo, también denominado pasarela a nivel de circuito, trabaja en la capa de sesión del modelo OSI o en una “ranura” entre las de transporte y aplicación de TCP/IP. En lugar de examinar los paquetes, monitorea las sesiones TCP o UDP en el proceso de *handshaking* entre dos máquinas. Una vez establecida la sesión los puertos permanecen abiertos para permitir el paso del resto de los paquetes que pertenecen a esa sesión. Los puertos son cerrados al terminar la sesión. La información que llega al destino pasando a través de este tipo de cortafuego aparece como si fuera originada en la pasarela, similar al funcionamiento de un Proxy, lo cual es muy útil a la hora de ocultar información sobre la red protegida.

Cortafuego de capa de aplicación (*Aplication level Gateway*)
También llamados *Proxy*, son similares a las pasarelas a nivel de circuito, excepto que son específicos para una aplicación. Trabajan el último nivel de la suite TCP/IP (aplicación) y pueden examinar el contenido de los datos, de manera que los filtrados se pueden adaptar a características propias de los protocolos de este nivel. Este filtrado permite controlar el acceso basado en la identidad del usuario o en la tarea concreta que el usuario está intentando llevar a cabo. Por ejemplo, si se trata de tráfico HTTP se pueden realizar filtrados según la URL y según la identidad del usuario.

Estos cortafuegos ofrecen un alto nivel de seguridad, debido principalmente a que los criterios que se utilizan para el filtrado, abarcan mucho más que los métodos anteriores, pero tienen un impacto negativo en el rendimiento de la red, debido a que el análisis que se lleva a cabo hace un poco más lento el proceso de conexión. Debe existir además una aplicación Proxy para cada servicio que el cortafuego debe soportar, lo cual supone un mayor trabajo de configuración; sin embargo, esta debilidad es también un punto fuerte que se añade a la seguridad del cortafuego. Por otro lado hay que agregar que no son transparentes a los usuarios, pues requieren ser configurados manualmente, en principio, en cada computadora de los clientes.

Inspección de paquetes completa (*Stateful Packet Inspection*)
La inspección completa de paquetes combina los aspectos de los tres tipos de cortafuegos descritos anteriormente. Cuando los paquetes llegan al firewall, la información de la cabecera es examinada y almacenada en una tabla dinámica de estados. Se compara luego con un grupo de reglas de filtrado pre-configuradas y se aceptan o se deniegan en base al resultado de la comparación.
Este método puede tomar decisiones basado en las direcciones IP de origen y destino, los tipo de protocolo, los puertos de origen y destino, el estado de la conexión, además de evaluar el contenido de los paquetes a nivel de aplicación usando algoritmos que reconocen y procesan los datos del nivel de aplicación en vez de ejecutar los proxies correspondientes a estas aplicaciones. El estado de la conexión se obtiene de la información de los paquetes previos, chequeando la tabla de estado dinámica para verificar que los paquetes son parte de una conexión establecida válida. Esto es un factor esencial para tomar la decisión con los nuevos intentos de comunicación.

Ventajas:

- Al igual que los cortafuegos por filtrado de paquetes, tienen un pequeño impacto en el rendimiento de la red.
- Son implementados de forma transparente para los usuarios finales.
- Son independientes de las aplicaciones, no requiere de proxy.
- Son más seguros que los sistemas de filtrado de paquetes básicos, puesto que usan una
- mayor cantidad de información de los paquetes para determinar el estado de la conexión y por tanto están mejor equipados para proteger contra accesos no deseados o no autorizados.
- Generalmente tienen alguna capacidad de almacenar registros, que permiten identificar y dar seguimientos a los diferentes tipos de tráficos que pasan por el cortafuego.

Desventajas:

- Permiten que se establezcan conexiones directas pues al igual que los cortafuegos por filtrado de paquete, no rompe el modelo cliente/servidor.
- Las reglas de filtrado pueden ser complejas, difíciles de administrar, con posibilidades de estar erróneas y difíciles de comprobar.

Caracterizar los sistemas de detección de intrusiones segun la fuente de datos, el motor de análisis y el tipo de respuesta.

Clasificación de los IDS y dónde colocarlos

Un IDS es necesario porque:

- Debemos tener conocimiento de todos los intentos de ataques para penetrar nuestro sistema, con el

fin de no caer en una falsa sensación de seguridad.

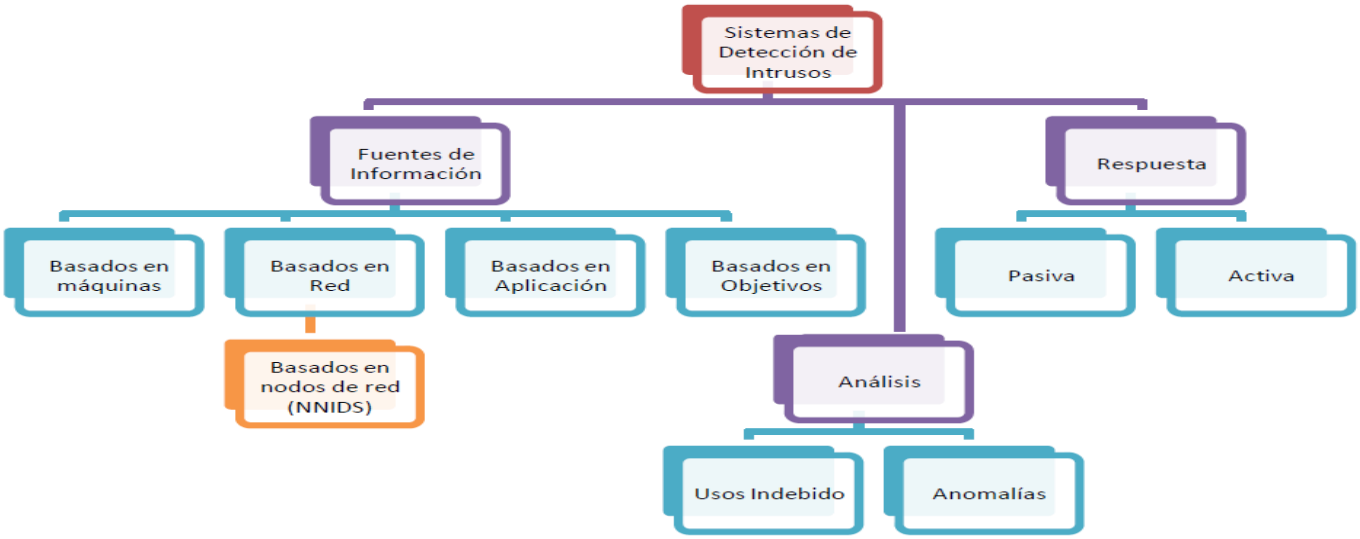
- Comprender que en cualquier momento alguien conseguirá romper la seguridad de nuestro entorno informático y debemos ser capaces de detectar el problema en el menor tiempo posible.

IDS 3 elementos funcionales básicos:

1. Una fuente de información, que proporciona eventos de sistema.

2. Un motor de análisis, que busca evidencia de intrusiones.

3. Un mecanismo de respuesta, que actúa según los resultados del motor de análisis.



Según la fuente de datos:

Fuente de información IDS basados en máquinas (host based)

- Monitorizan archivos (logs) generados por una máquina en busca de posibles modificaciones no autorizadas.
- Detectan ataques o intrusiones contra la máquina sobre la que se ejecutan.
- Logs: (registro de todas las aplicaciones del sistema).
- Pueden registrar abundante información acerca de eventos relacionados con el sistema que lo genera.
- Permiten detectar incidentes de seguridad.
- Contienen información muy útil para la recuperación ante incidentes de seguridad.

Caso especial: analizadores de integridad.

- Detecta intrusiones mediante los cambios encontrados en ficheros.
- En general, su búsqueda se enfoca en puertas traseras dejadas por un intruso.
- Son de gran ayuda en el análisis forense, facilitando la identificación del atacante y el método empleado.

Sistemas de decepción o honeypots (tarros de miel)

- Diseñados para recibir ciertos tipos de ataques.
- Son capaces de detectar una actividad hostil y aplicar una estrategia de respuesta.
- Ofrecen conocimiento negativo, información falsa que el atacante creerá real.
- Pueden simular vulnerabilidades conocidas para que el atacante piense que ha tenido éxito y continúe la actividad mientras es monitorizado por el IDS.

IDS basados en red (network based)

Capaces de detectar ataques contra diferentes sistemas de una □□□□□IDS□□Permiten prevenir violaciones de seguridad por actividades maliciosas en la red, tales como el engaño de direcciones IP, ataques de negación de servicio, etc.

Nodos de red (network node based):

Se sitúan en un host y monitorizan los paquetes destinados a u originados desde la máquina anfitriona.

Honeynets (redes de miel):

Redes diseñadas para ser comprometidas, formadas por sistemas de todo tipo que, una vez penetrados, permiten capturar y analizar la actividad (tácticas) del atacante.

En este caso no hay simulación de vulnerabilidades, sino que se ejecutan aplicaciones típicas.

En objetivo en este caso no es la decepción, sino aprender los posibles movimientos de un pirata en un entorno semireal, de manera que vulnerabilidades y configuraciones incorrectas puedan ser detectadas y corregidas en el entorno real.

IDS basados en objetivos (target based):

Similar a los IDS basados en máquina.

Generan sus propios registros, pues graban periódicamente el estado de una serie de recursos valiosos del sistema. Este estado es luego comparado con las políticas de seguridad y se registran posibles incongruencias.

Según el tipo de análisis:

Detección de usos indebidos:

Se comparan patrones de ataques conocidos, con la información de la fuente de datos en busca de coincidencias.

Detección de anomalías:

Se manejan técnicas estadísticas que definen de forma aproximada lo que es el comportamiento usual o normal. Cada intrusión es manejada como una anomalía (** Actividad poco común).

Según el mecanismo de respuesta:

IDS de respuesta pasiva:

El IDS no toma acciones que puedan cambiar el curso de un ataque, sino que se limita a enviar o registrar la alarma correspondiente al responsable cualificado.

Ejemplos: Emisión de un sonido de alerta, emisión de un evento al log del sistema, envío de mensajes de correo, etc.

IDS de respuesta activa:

Además de generar la alarma correspondiente, reaccionan modificando el entorno.

Ejemplos: Cierre de la sesión del usuario sospechoso, salva de paquetes con evidencias, ejecución de programas, etc.

Explicar los aspectos teóricos básicos para la definición de un plan de contingencia.

Un plan de contingencia es un programa alternativo para que una empresa pueda recuperarse de un desastre informático y restablecer sus operaciones con rapidez. Estos planes también se conocen por la sigla DRP, del inglés Disaster Recovery Plan.

El Plan de Contingencia para la Seguridad Informática se establece como una exigencia para todas las entidades, con el fin de garantizar la continuidad de los procesos informáticos ante cualquier desastre que pueda ocurrir. Contendrá las medidas que permitan, en caso de desastres, la evacuación, preservación y traslado, de los medios y soportes destinados al procesamiento, intercambio y conservación de información clasificada o sensible. Así mismo, contemplará las medidas pertinentes para la conservación y custodia de los ficheros creados con fines de salvaguarda.

Para la elaboración del PRD se deben tener en cuenta las siguientes 10 (diez) partes:

1) Determinación del escenario considerado
2) Definición del tipo de operación en una contingencia
   - Operación normal inicial
   - Operación alternativa
   - Operación normal en desastre
   - Operación normal restablecida
3) Establecimiento de criticidades *(mostrar tabla de Criticidades por equipos y la de Criticidades por Servicios)*
4) Determinación de prestaciones mínimas
5) Análisis de riesgos *(mostrar tablas para análisis de los riesgos y amenazas)*
6) Estrategias de recuperación
7) Requerimientos para llevar a cabo el plan
8) Pruebas de PRD
9) Revisión del PRD
10) Cierre del proyecto

Planificar la salva y recuperación de datos a partir de la aplicación de principios y buenas prácticas para su implementación.

Prácticas recomendadas

- Desarrolle estrategias de copia de seguridad y restauración, luego pruébelas correctamente.
- Proporcione entrenamiento al personal correspondiente.
- En redes con nivel de seguridad mínimo o medio, asigne derechos de copia de seguridad a un usuario y derechos de restauración a otro diferente. Entrene al personal que tenga derechos de restauración para realizar las tareas de restauración en caso de que sean necesarias en ausencia del administrador.
- En una red de alta seguridad, asegúrese de que sólo los administradores puedan restaurar archivos.
- Haga copia de seguridad de todos los datos de los volúmenes de sistema y de inicio, así como del Estado del sistema.

- Haga copia de seguridad de los datos de todos los volúmenes y de los datos del Estado del sistema al mismo tiempo. Esta precaución le permitirá estar preparado en el improbable caso de que se produzca un error de disco.

• Cree un conjunto de copia de seguridad de Recuperación automática del sistema.

- Cree siempre un conjunto de copia de seguridad de Recuperación automática del sistema (ASR) cuando se realicen cambios en el sistema operativo; por ejemplo, cuando instale nuevo hardware o nuevos controladores, o cuando aplique un Service Pack. Al disponer de un conjunto de copia de seguridad ASR, será más sencillo recuperarse de un error del sistema. También debe hacer copia de seguridad de todos los volúmenes de datos al mismo tiempo; ASR sólo protege el sistema, por lo que debe hacerse copia de seguridad de los volúmenes de datos de forma independiente.

• Cree un registro de copia de seguridad.

- Elija siempre que se cree un registro de copia de seguridad para cada copia e imprímalo como referencia.
- Mantenga un libro de registros que facilite la búsqueda de determinados archivos. El registro de copia de seguridad es útil para restaurar los datos, y puede imprimirlo o leerlo con cualquier editor de texto. Además, si resulta dañada la cinta que contiene el catálogo del conjunto de copia de seguridad, el registro impreso puede ayudarle a encontrar un archivo.

• Conserve copias.

- Mantenga al menos tres copias de los medios. Guarde como mínimo una copia fuera del emplazamiento, en un entorno correctamente controlado.

• Realice restauraciones de prueba.

- Realice una restauración de prueba de forma periódica para comprobar que la copia de seguridad de los archivos haya sido correcta. La restauración de prueba puede descubrir problemas de hardware no detectables mediante comprobaciones de software.

• Proteja los dispositivos y los medios.

- Proteja los dispositivos de almacenamiento y los medios de copia de seguridad. Una persona que tenga acceso a los datos de un medio robado podría restaurarlos en otro servidor en el que tenga derechos de administrador.

Seleccionar adecuadamente los tipos de copias a emplear segun el volúmen de datos y la frecuencia de actualización de los mismos.

- Tipos de copias de seguridad

En función de la cantidad de archivos que se salvaguardan a la hora de realizar la copia de seguridad, podemos distinguir varios tipos de copia entre estos:

Copia de seguridad diaria

Copia todos los archivos seleccionados que se hayan modificado el día en que se realiza la copia diaria. Los archivos incluidos en la copia de seguridad no se marcan como copiados (es decir, no se desactiva el atributo de modificado).

Copia incremental

En un proceso de copia de seguridad incremental, se hace una copia de seguridad sólo de los archivos que han cambiado desde la última copia de seguridad realizada. Ejemplo, si hacemos copia de seguridad total el día 1 de cada mes y copia de seguridad incremental el resto de los días, cada copia incremental solo guardará los archivos que se hayan modificado ese día. Si tenemos que realizar la restauración de archivos

ante un desastre, debemos disponer de la copia total y de todas las copias incrementales que hayamos realizado desde la copia total.

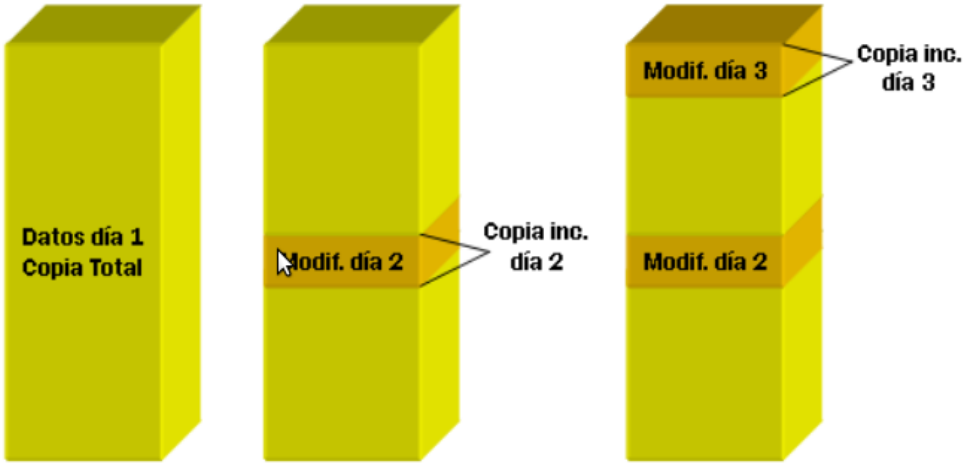


Ilustración 1 Las copias incrementales guardan solo los archivos modificados desde la última copia incremental

Copia diferencial

Una copia de seguridad diferencial es una copia de todos los archivos que han cambiado desde la última copia de seguridad total que hayamos hecho. Ejemplo, si hacemos copia de seguridad total el día 1 de cada mes y copia de seguridad diferencial el resto de los días, cada copia diferencial guardará los archivos que se hayan modificado desde el día 1. La ventaja es que se requiere menos espacio que la copia total y que **en el proceso de restauración únicamente necesitaremos la última copia total y la última copia diferencial. Una copia diferencial anula a la copia diferencial anterior**. Por el contrario, se consume más tiempo en realizar la copia y también más espacio que en el caso de copia incremental.

Copia normal o copia total

Una copia de seguridad normal, es una copia de seguridad total de todos los archivos y directorios seleccionados.

En las copias de seguridad normales sólo se necesita la copia más reciente del archivo o la cinta que contiene la copia de seguridad para restaurar todos los archivos. Las copias de seguridad normales se suelen realizar al crear por primera vez un conjunto de copia de seguridad.

La combinación de copias de seguridad normal e incremental utiliza el mínimo espacio de almacenamiento posible y es el método de copia de seguridad más rápido. Sin embargo, la recuperación de archivos puede ser difícil y laboriosa ya que el conjunto de copia de seguridad puede estar repartido entre varios discos o cintas.

Si realiza una copia de seguridad de sus datos empleando una combinación de copias de seguridad normal y diferencial consumirá más tiempo, especialmente si los datos sufren cambios frecuentes, aunque será más fácil restaurar los datos ya que el conjunto de copia de seguridad sólo estará repartido en unos pocos discos o cintas.

- **Recomendación sobre el tipo de copia a efectuar**

Si el volumen de da tos de nuestra copia de seguridad no es muy elevado (menos de 4 GB), lo más práctico es realizar siempre copias totales ya que en caso de desastre, tan solo debemos recuperar la última copia.

Si el volumen de datos de nuestra copia de seguridad es muy elevado (mayor de 50 GB) pero el volumen de datos que se modifican no es elevado (sobre 4 GB), lo más práctico es realizar una primera copia total y posteriormente realizar **siempre copias diferenciales**. Así, en caso de desastre, tan solo debemos recuperar la copia total y la última diferencial. Periódicamente debemos realizar una copia total y así empezar de nuevo.

Si el volumen de datos de nuestra copia de seguridad es muy elevado (mayor de 50 GB) y el volumen de datos que se modifican también lo es, las copias diferenciales ocuparán mucho espacio, por lo tanto en este caso lo más práctico será realizar una primera copia total y posteriormente realizar **siempre copias incrementales** ya que son las que menos espacio ocupan. El problema es que en caso de desastre debemos recuperar la última copia total y todas las incrementales realizadas desde que se hizo la última copia total. En estos casos, conviene hacer copias totales más a menudo para no tener que mantener un número muy elevado de copias incrementales.

En grandes compañías donde la realización de copias de seguridad está perfectamente planificada, se suelen utilizar sistemas mixtos. Por ejemplo en un caso típico se realizarían las siguientes tareas:

- Todos los días 1 de cada mes, a las 23:00 horas: copia de seguridad total
- Todos los viernes a las 23:00 horas: copia de seguridad diferencial desde la copia de día 1
- Todos los días (excepto los viernes y el día 1) a las 23:00 horas: copia de seguridad incremental desde la copia del día anterior.

Con ésta planificación nos aseguramos disponer de copia de seguridad diaria. En caso de desastre deberíamos recuperar la copia total, la última diferencial y todas las incrementales desde la última diferencial.

En una política de este tipo se pueden utilizar por ejemplo 5 juegos diferentes de cintas de forma que se almacenen las copias de seguridad diarias de los últimos 3 meses. Luego se van reutilizando pero no más de 20 veces ya que las cintas se deterioran y la fiabilidad disminuye.