

Seguridad de Datos - Conceptos

Conceptos fundamentales de seguridad

- Qué proteges, y de quién
- Amenazas, vulnerabilidades y ataques
- Autenticación, Privacidad, Integridad, No-repudio

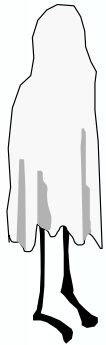


Qué proteges? De quién?

Tus datos

Tus recursos

Tu reputación



Aburridos

Vándalos

Busca-trofeos

Espías

Amenazas, Vulnerabilidades y Ataques

Amenaza (*threat*)::=

cualquier ocurrencia potencial, maliciosa o no, que pueda tener efectos indeseables en los bienes o recursos asociados con un sistema de cómputos
(*algo malo que podría llegar a ocurrir*)



Tipos de amenazas

Revelación (disclosure, leak) ::= diseminación de información hacia alguien que no debía haberla recibido.

Integridad (Integrity) ::= destrucción o alteración, no autorizadas, de información almacenada o en tránsito.

Negación de servicio (Denial of service) ::= bloqueo intencional del acceso a un recurso



Amenazas, Vulnerabilidades y Ataques

Vulnerabilidad::=

Característica desafortunada de los sistemas, que hace posible la existencia de una *amenaza*.

La presencia de vulnerabilidades permite que ocurran las cosas indeseables;

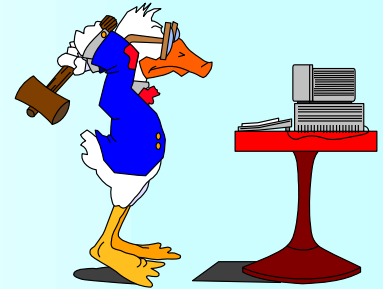
=> las amenazas pueden mitigarse por medio de la identificación y eliminación de las vulnerabilidades



Amenazas, Vulnerabilidades y Ataques

Ataque:

Acción de un intruso malicioso que aprovecha ciertas *vulnerabilidades* del sistema para hacer que ocurra una *amenaza* preexistente.



Ataques: una taxonomía

Substracción externa de información

(ej. mirar la pantalla del sysop)

Abuso externo de recursos

(ej. 220-Base-T)

Enmascaramiento

(ej. grabar y reproducir una transmisión, secuestro de sesión -- hijacking)

Programas infecciosos (virus, trojans, etc)

Saltear controles internos (cracking passwords)

Abuso de autoridad (falsificación de registros)

Abuso por no intervención (mala administración intencional)

Abuso indirecto (usar otro sistema para crear un programa malicioso)

Conceptos

Conceptos fundamentales de seguridad

Qué proteges, y de quién

Amenazas, vulnerabilidades y ataques

Autenticación, Privacidad, Integridad, No-repudio

Trusted paths

Covert channels

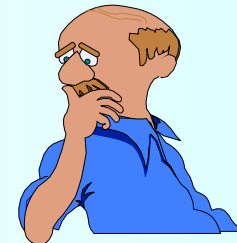
Identificación y autenticación

Asegurar a un sistema quién es el usuario que solicita un recurso

Identificación:

mecanismo por el cual alguien indica su *identidad* al sistema

- ¿Quién eres tú?
- Yo soy Carlos Gardel
- Oh.





Identificación y autenticación

Autenticación::=

mecanismo por el cual se puede asegurar que la identidad indicada es verdadera

Algo que se **posee**

(algo que sólo el agente identificado pueda tener en su poder, ej. smart card)

Algo intrínsecamente **propio**

(característica inherente del agente identificado,
ej. biometrics, voice-print, huella digital, patrones retinales)

Algo **sabido**

(un secreto no compartido:
ej. passwords, esquemas challenge / response)

Protocolos de autenticado

Una autenticación efectiva requiere un procedimiento, que puede involucrar varias transacciones complejas entre dos o más entidades. Cada componente de un dado protocolo de autenticación debe ser resistente a los distintos tipos de ataque.

El autenticado no es un problema trivial: se desaconseja la creatividad *naive*

ej. challenge-response con clave pública: firmar un nonces que no es un nonces



Passwords

Generación automática, o por el usuario; memorizables; diccionarios; atributos personales obvios; secuencias en el teclado; reutilización.

Passwords por única vez

Encriptado; salt:

si el password es	pepe ,
encriptar	<u>K4</u> pepe ,
obteniendo	dj!g%Kda ,
y guardar	<u>K4</u> dj!g%Kda

Privacidad

Mecanismo de ocultamiento selectivo de la información, para evitar la amenaza de revelación (“disclosure”)
debe ser fácil encriptar
debe ser fácil desencriptar si se dispone de la clave
difícil (“imposible”) desencriptar si no se dispone de la clave

Integridad

Mecanismo que permite verificar si un documento ha sido alterado en el intervalo que media entre dos puntos de control

no puede impedir la modificación, pero de existir la detecta.



No repudio

::= Mecanismo por el cual un agente que efectúa una acción no puede posteriormente *negar* haberla efectuado

Requisito clave para posibilitar la digitalización de circuitos administrativos, utilización del documento digital, despapelización de las oficinas, transacciones a distancia

Se requiere:

Identificación autenticada del agente emisor

verificación de integridad del documento (digital)

Cripto-Elementos

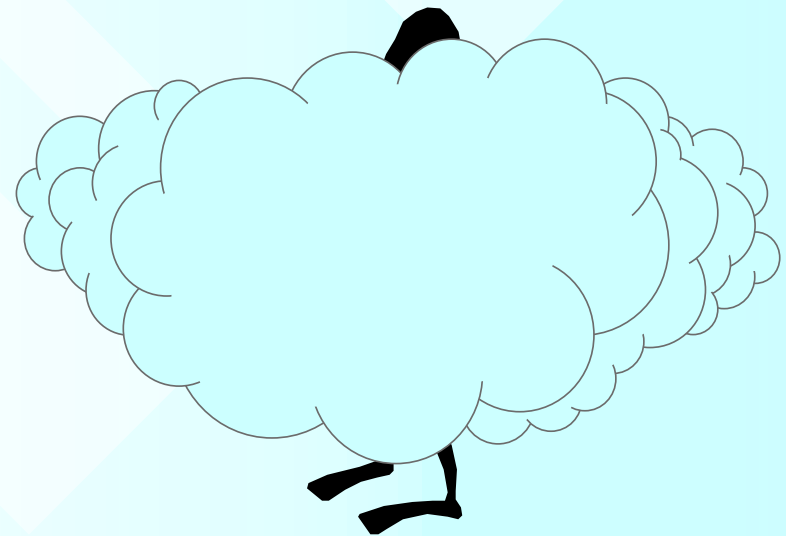
Cripto-Elementos

Hashes

Ciphers

Simétricos (secreto compartido)

Asimétricos (clave pública)



Criptología

Criptografía

Cifrar (*encrypt*) ::=

una transformación **E** que enmascara un mensaje **M**
(*plaintext, cleartext*) para ocultar su substancia. Se produce un
texto cifrado **C** (*ciphertext*)

$$E_{k_1} [M] = C$$

Descifrar (*decrypt*)

$$D_{k_2} [C] = M$$

Espacio de claves (*keyspace*) ::=

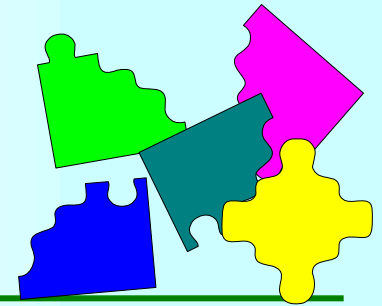
rango de posibles valores de la clave **K**

Criptoanálisis

algoritmos seguros (*secure*) ::=

no descifrables (*breakables*) en la práctica

Elementos



Las aplicaciones criptográficas más usuales se construyen con estos 2 elementos:

Funciones de hash seguras (*Secure Hash Functions*)

Cifradores (*Ciphers*)

Simétricos,

(también llamados convencionales, o de secreto compartido)

Asimétricos o de clave pública

(public key cyphers)

Hash Function

:: = una función **H** que toma un mensaje **M**, y produce un resultado **h**, de tamaño fijo (generalmente mas corto que M)

Ejemplo:

$h :=$ [el resultado de hacer un X-OR
entre todos los bytes del mensaje M]

Al resultado (h) de aplicar una *hash function* a un mensaje se lo suele llamar simplemente *hash* del mensaje M



Secure Hash Functions (shf)

$h = H(M)$ pero con los siguientes requisitos:

Dado M debe ser fácil computar h

Dado h debe ser difícil encontrar el M original

Dado un cierto M , es difícil encontrar otro M' tal que
 $H(M) = H(M')$

2 formas habituales de uso de las s.h.f.:

Sin clave

Con clave (ej. usar una s.h.f sin clave y encriptarla). Para MACs

Aplicaciones: integridad de mensajes o archivos, firma de digestos en vez de mensajes.

Ejemplo: hashes para verificar una distribución de software (pero...)

MD5 (progr1.tar.Z) = 5388b86a01300e7525d3cc1c36aab523

MD5 (progr2.tar.Z) = 850ec13fc8f43daed1a42883fb29577e

MD5 (LICENSE) = 3d50052fb71fec10388fddb918af11c4

MD5 (README) = 2540c145b8ac9a06b0c4807a37d3f62c



Dificultades

Una s.h.f. de 128 bits puede producir 2^{128} posibles hashes. Necesitamos 2^{128} intentos* para hallar algún string cuyo hash sea igual a uno dado.

Sin embargo, sólo necesitamos 2^{64} intentos para hallar un par M, M' donde $H(M) = H(M')$

(birthday attack)

* en el peor caso

Ataques a las SHF

El usuario **A** utiliza una shf para formar un *fingerprint* de su mensaje **M**, luego firma éste.

Si es usuario **B** obtiene otro **M'** que resulte en idéntico fingerprint, entonces **B** puede decir que **A** firmó **M'**

Birthday attack (ejemplo no práctico):

A genera 2 “contratos”, **M** y **M'**, que resulten en un mismo fingerprint.

Envía **M** a **B**, que firma el fingerprint del contrato.

Luego **A** puede decir que **B** firmó **M'**.



“Secure” hashes

SNEFRU (128/256 bits); El SNEFRU de 2 pasos se puede quebrar, usando una PC, en 3 minutos [birthday: dado M , hallar M' cuyo $H(M')=H(M)$], o en 1 hora (hallar un mensaje M dado un hash h)

N-HASH

MD2 (Message Digest-2, Ron Rivest, 128 bits, mas lento, menos seguro que MD4, usado en PEM)

MD4 (Message Digest-4, Ron Rivest, 128 bits, muy usado)

MD5 (Message Digest-5, Ron Rivest, 128 bits, MD4 con mejoras, muy usado, usado en PEM, S/MIME, SSL)

SHA (secure hash algorithm, 160 bits, usado en DSS, SSL, S/MIME)

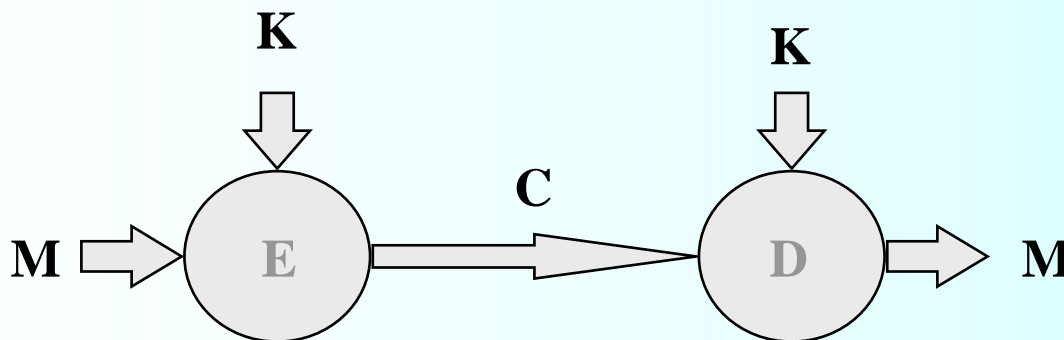
Otros

Criptografía Simétrica (o de secreto compartido)

$C = E_k [M]$ (cifrar)

$M = D_k [C]$ (descifrar)

La clave (K) es la misma para ambas operaciones



Criptografía simétrica

Problemas fundamentales

distribución de la clave: requiere canal seguro independiente

revelación de la clave:

la clave no sólo está en poder de uno.

Conceptualmente imposible el no-repudio

Se necesita una clave por cada par
de usuarios: $n(n-1)/2$ claves
para n usuarios

(para evitar esto puede usarse un *key server*, pero éste
debe ser muy seguro)

Algoritmos simétricos

DES y 3DES

RC2 (block, Ron Rivest, concebido para reemplazar al DES, clave de tamaño variable)

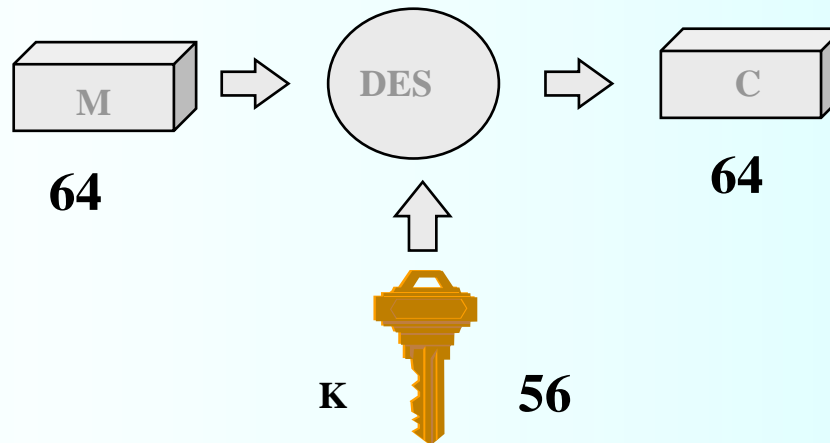
RC4 (stream, Ron Rivest)

IDEA (block, international data encryption std. clave de 128 bits; modos ecb, cbc, ofb, cfb)

AES (Advanced Encryption System)

Otros

DES: Data Encryption Standard



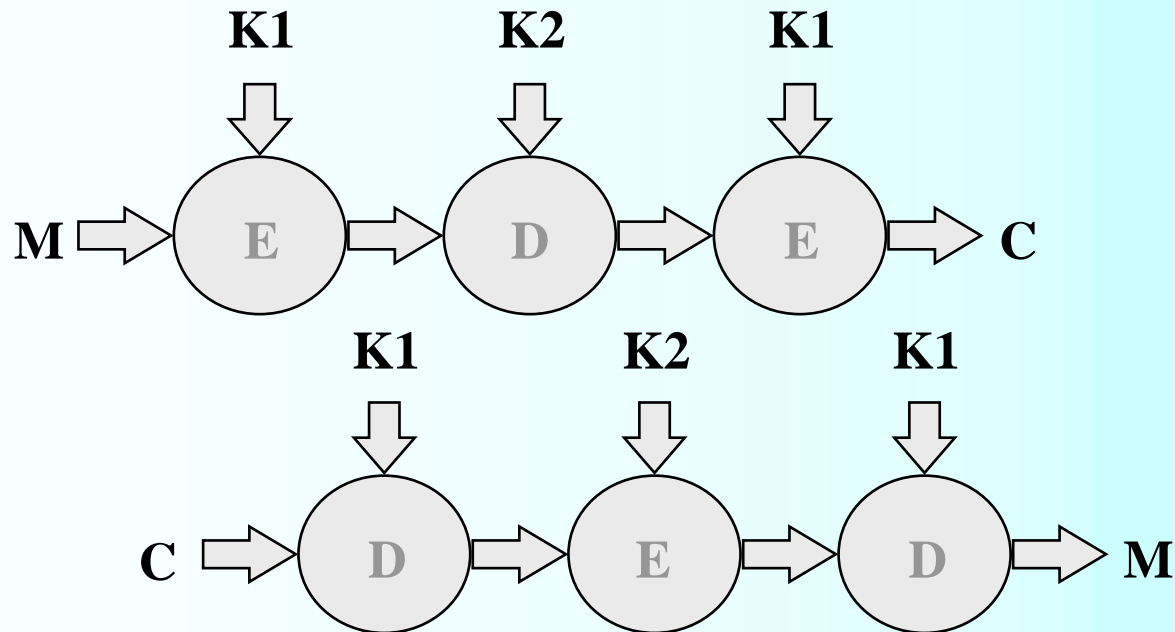
Es un cifrador de *bloques* de 64 bits, con claves de 56 bits.

La clave suele expresarse como un número de 64 bits
(8 veces 7 bits de clave con su bit de paridad)

Triple-DES (modo EDE)

El cipher resultante es mucho más difícil de descifrar (2^{112} en vez de 2^{56})

Es compatible con DES si $K1 = K2$



Block ciphers: 4 modos

4 Modos de operación:

ECB: Electronic Code Book

CBC: Cypher Block Chaining

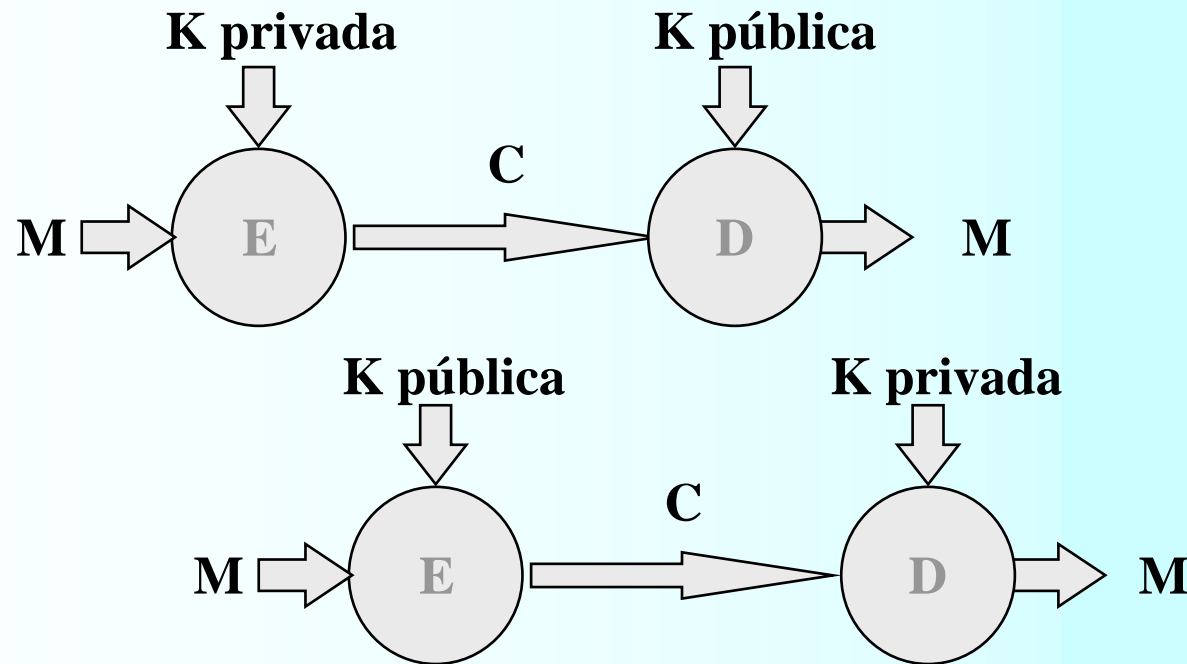
OFB: Output FeedBack

CFB: Cypher FeedBack

Criptografía Asimétrica (Sistemas de Clave Pública)

Mediante un programa, el usuario genera un PAR de claves. Una es la pública, la otra es la privada.

Si se encripta con una, se descrypta con la otra.



Criptosistemas de clave pública

Deducir una clave a partir de la otra es computacionalmente irrealizable

100-1000 veces más lentos que los simétricos

claves mucho más largas que los simétricos

Solución para el problema de distribución de claves, pero no completa:

Ataque: sustitución de clave pública:

man-in-the-middle (MITM) / bucket brigade



Algoritmos de clave pública

Diffie-Hellman (distribución de claves)

RSA (distribución de claves, encriptado)

ElGamal (distribución de claves, encriptado)

DSA (firmas digitales)

La mayor parte basados en uno de estos tres problemas difíciles:

logaritmo discreto:

p, primo: g y M enteros,
encontrar x tal que $g^x = M \pmod{p}$

factorio

knapsack (dado un conjunto de números particulares, encontrar un subconjunto cuya suma sea N)



R S A

Rivest, Shamir, Adleman

Clave pública:

$n = p \cdot q$ (ambos primos, secretos)

e primo relativo a $(p-1)(q-1)$

(sin factores comunes)

Clave privada

$d = e^{-1} \pmod{(p-1)(q-1)}$

Cifrado

$c = m^e \pmod{n}$

Descifrado

$m = c^d \pmod{n}$

Sistemas de clave pública

Firma Digital

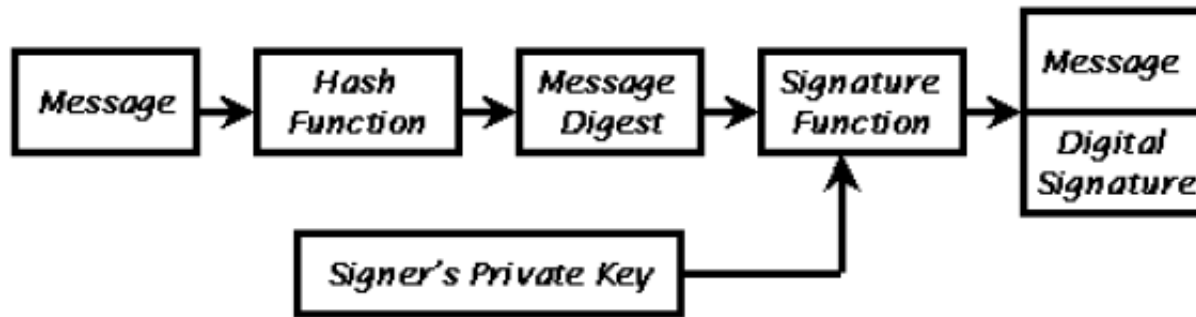


Figure 1: The process used to create a Digital Signature.

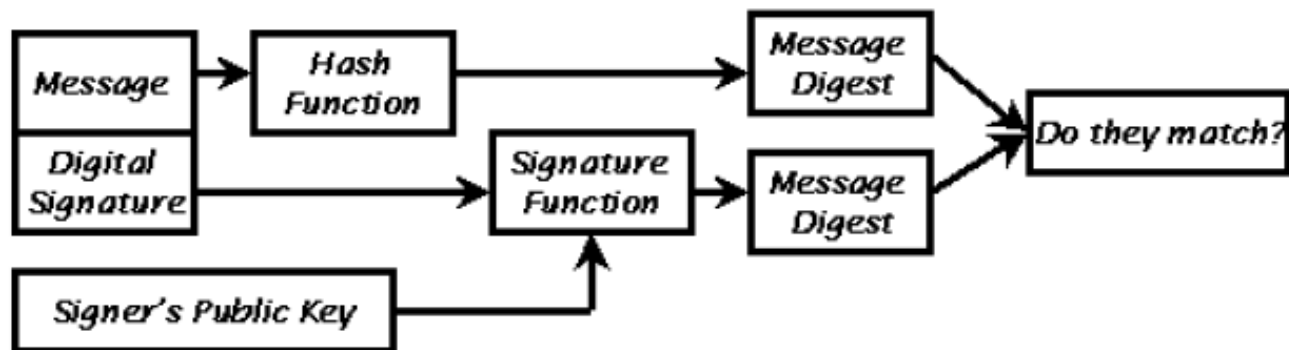


Figure 2: The process used to verify a Digital Signature.

Sistemas de clave pública

Los sistemas de clave pública son mucho más lentos, para servicios de privacidad se usan en combinación con los simétricos, que son más rápidos.

Con el sistema simétrico se cifra el mensaje; con el de clave pública se intercambia la clave que se usó para cifrar el mensaje.

Por ej., mandar el mensaje cifrado con DES usando una clave determinada (*session key*), y enviar esa clave en otro mensaje, ésta vez cifrado con RSA.



Sistemas de clave pública

Sobre Digital

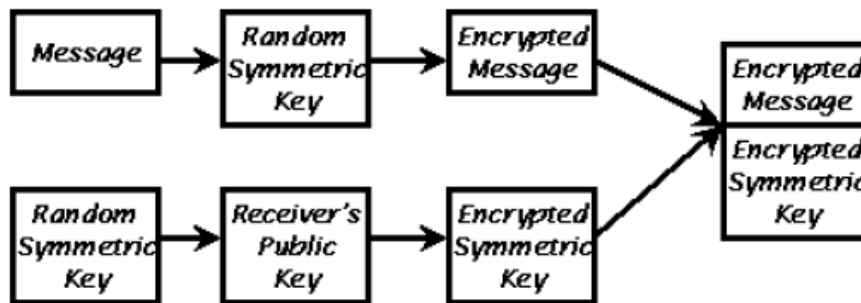


Figure 3: The process used to create a Digital Envelope.

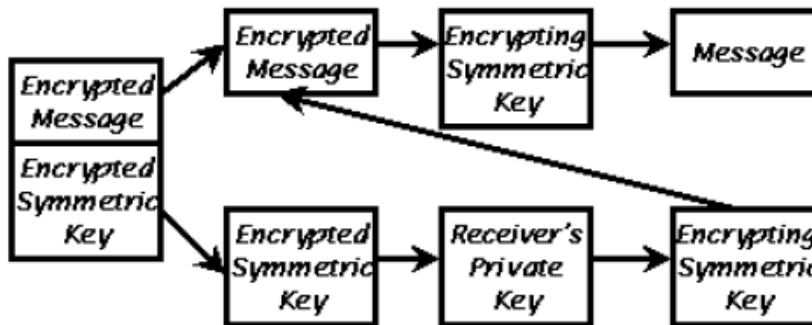


Figure 4: The process used to verify a Digital Envelope.

Debilidades deliberadas

Filtrar un bit de la clave de vez en cuando

Suavizar la clave de manera de que no tenga más de 30 bits efectivos

Encriptar un header conocido al comienzo de cada mensaje (permite ataque *chosen-plaintext*),
o bien encriptar un mensaje corto, (random) y colocar *plaintext* y *cipher* al comienzo de cada mensaje



Seguridad en Internet e Intranet

Seguridad en redes TCP/IP

Ejemplos de Ataques en redes TCP/IP

Host Security

Network Security: Firewalls

Application Security

Acceso Remoto



Network Security: Firewalls

- Justificación, Características, Protección perimetral
- Packet Filters, Application Gateways, Filtrado dinámico
- Protección en routers: Access Lists
- Distintas arquitecturas de firewall
- Servicios Proxy
- Virtual Private Networks (VPN)
- Ej: Firewall-1 de Checkpoint

Firewalls

Protección perimetral

Elementos

Packet filters,

Application gateways

Circuit level gateways

Filtros dinámicos



Dificultades

UDP: difícil establecer contexto

FTP: operación con conexión de control mas conexión para transferencia de datos ésta última entrante hacia el cliente

RPC / portmapper



Transparencia

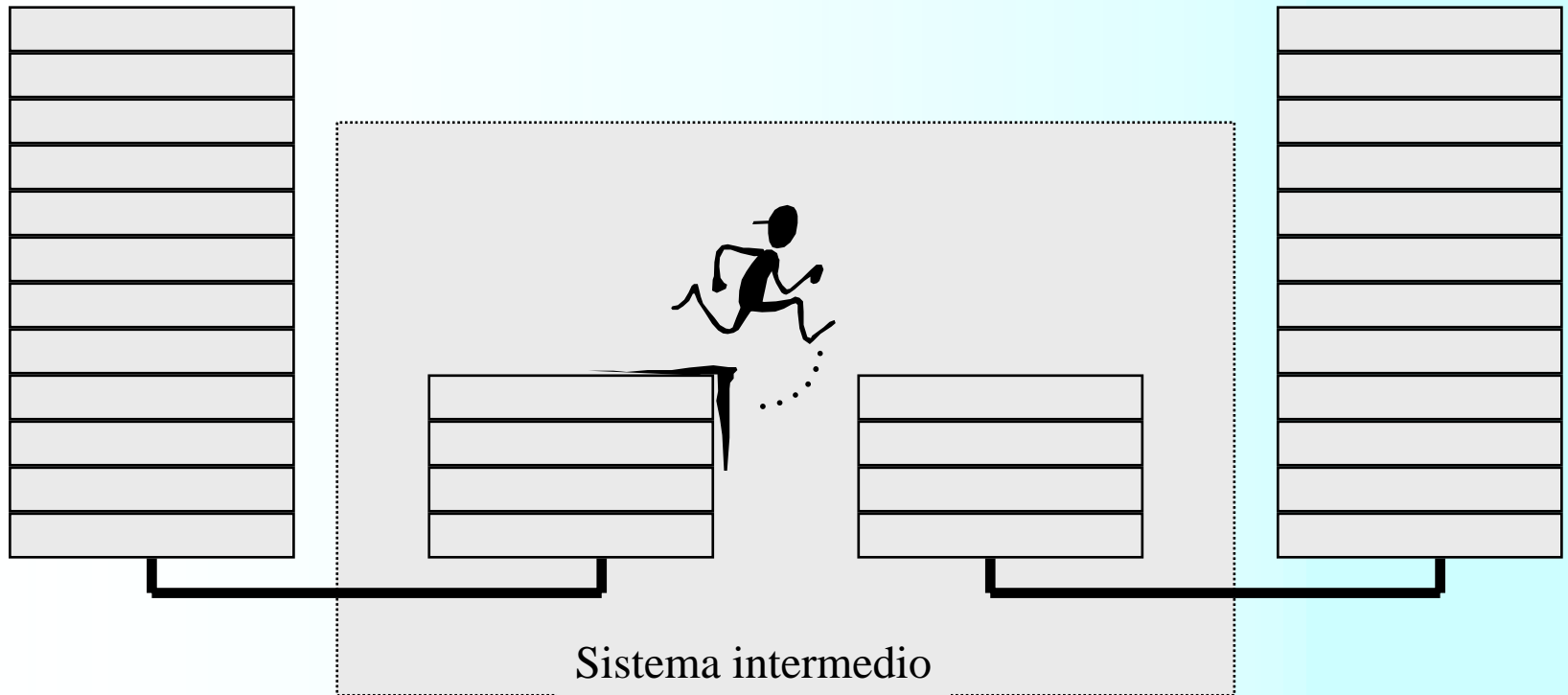
```
telnet firewall.sseg.com.ar
Proxy para telnet de Sistemas Seguros S.A.
Uso exclusivo usuarios autorizados
tn-gw->connect sherlock
Trying 134.172.94.3 port 23...
UNIX System V Release 3.2 (sherlock.sseg.com.ar) (ttyp1)
login: qwatson
Password:
Last successful login for qwatson:Thu Oct 10 20:15:25 BST 1996 on ttyp11
Last unsuccessful login for qwatson:Mon Sep 23 12:43:17 BST 1996 on ttyp0
Terminal type is ansi
320$
...
```



Transparencia (cont.)

```
329$ ftp angela 3213
Connected to angela.sseg.com.ar
220 Proxy para FTP - Sistemas Seguros S.A.
220 Uso exclusivo usuarios autorizados
501 Use user@site to connect via proxy
user: rhood@ftp.sseg.com.ar
331-(----GATEWAY CONNECTED TO ftp.sseg.com.ar----)
331-(220-Usuarios autorizados: Bienvenidos. )
331-(220-Usuarios no autorizados: Canelen esta conexion inmediatamente)
331-(220-)
331-(220 ftp.sseg.com.ar FTP server (Version 2.1WU(1)) ready.)
331 Password required for rhood.
Password:
ftp>
```

Dónde ubicar los puntos de control? (reference monitors)



Filtrado de paquetes (*packet filtering*)

Eficiente

Transparente (no requiere intervención del usuario)

Ubicado estratégicamente, es muy efectivo

No es costoso (ya viene incluido en los routers)

Característica

Anti IP source-address spoofing (no necesariamente un ataque)



Filtrado de paquetes (cont.)

(packet filtering)

Posibilidades de filtrado

dirección de IP - fuente

dirección de IP - destino

protocolo (tcp, udp, icmp)

número de port (tcp o udp) fuente

número de port (tcp o udp) destino

bit ACK del header de TCP

tipo de message ICMP

interface de entrada

interface de salida

Filtrado de paquetes (cont.)

Los criterios para la decisión son limitados por naturaleza, y a veces incompletos por implementación

No se puede decidir en base a operaciones individuales dentro de un servicio

Ciertos protocolos (ej. FTP, “r commands”, RPC) no se adaptan bien a ser tratados por los filtros

Capacidad de *log* muy limitada

La configuración es compleja, y susceptible a errores

Difíciles de probar



Filtrado de servicios (ejemplos)

Sentido	Dir. fuente	Dir. destino	Prot.	Port fuente	Port destino
entrante	externa	interna	PROT	>1023	WKP
saliente	interna	externa	PROT	WKP	>1023
saliente	interna	externa	PROT	>1023	WKP
entrante	externa	interna	PROT	WKP	>1023

SERV.	WKP	PROT
pop2	109	TCP
pop3	110	TCP
smtp	25	TCP
TFTP	69	UDP
telnet	23	TCP
http	80	TCP
gopher	70	TCP

Listas de acceso en routers

(reglas de filtrado de paquetes)

```
access-list 111 permit tcp any 212.20.153.0 0.0.0.255 eq pop3
access-list 111 permit tcp any 212.20.153.0 0.0.0.255 eq www
access-list 111 permit udp any 212.20.153.0 0.0.0.255 eq domain
access-list 111 permit tcp any 212.20.153.0 0.0.0.255 eq smtp
access-list 111 permit ip any host 212.20.153.134
access-list 111 deny ip any 212.20.153.0 0.0.0.255
access-list 111 permit ip any any
...
access-list 112 permit ip any host 212.20.153.134
access-list 112 permit ip host 245.1.134.27 any
access-list 112 permit ip host 245.1.134.81 any
access-list 112 deny tcp any 212.20.153.0 0.0.0.255 eq telnet
access-list 112 deny tcp any 212.20.153.0 0.0.0.255 eq ftp
access-list 112 permit ip any any
...
interface Async3
ip access-group 111 in
...
interface Serial1
ip access-group 112 in
```

Filtrado dinámico

Resuelve las limitaciones de los firewalls de tipo packet filtering tradicionales, por ej:

UDP

FTP

RPC



Filtrado dinámico

UDP

Aplicaciones UDP en Packet Filters tradicionales:

Para permitir sesiones UDP originadas internamente, los packet filters tradicionales deben permitir ingresar los datagramas UDP que van dirigidos a cualquier port efímero (>1024) exponiendo la red interna a posibles ataques.



Filtrado dinámico

UDP (cont.)

Aplicaciones UDP en Packet Filters con filtrado dinámico:

El sistema intermedio registra el intercambio de los datagramas asociados a una sesión de servicio UDP, permitiendo el forwarding de un datagrama entrante sólo si es un response a un request saliente previo.



Filtrado dinámico FTP

Sesiones FTP salientes en Packet Filters tradicionales:

Para permitir sesiones FTP originadas internamente, los packet filters tradicionales deben permitir el establecimiento de conexiones TCP que van dirigidas a cualquier port efímero (>1024) exponiendo la red interna a posibles ataques.

Filtrado dinámico

FTP (cont)

Sesiones FTP salientes en Packet Filters con filtrado dinámico:

El sistema intermedio registra el intercambio de los datagramas asociados a una sesión de servicio FTP, permitiendo una conexión TCP entrante sólo si es una “ftp back-connection” que se corresponde con un ftp-command iniciado por un ftp-client interno, o sea a la dirección del ftp client y al port en el cual espera la back-connection.

Filtrado dinámico

RPC

Sesiones RPC en Packet Filters tradicionales:

En general no es posible filtrar adecuadamente éste tipo de tráfico, o para hacerlo, deben dejar abiertas brechas importantes que exponen demasiado a la red interna.



Filtrado dinámico

RPC (cont)

Sesiones RPC en Packet Filters con filtrado dinámico:

El sistema intermedio registra el intercambio de los datagramas asociados a una sesión de servicio RPC entrante y/o saliente, complementando los mecanismos ya descritos (por ej. para UDP) con la consulta de los portmappers a modo de proxy, dejando pasar solamente el tráfico UDP/TCP asociado a servicios RPC expresamente habilitados.

Arquitecturas de Firewalls

Host con 2 interfaces
(*dual-homed host*)

Host apantallado

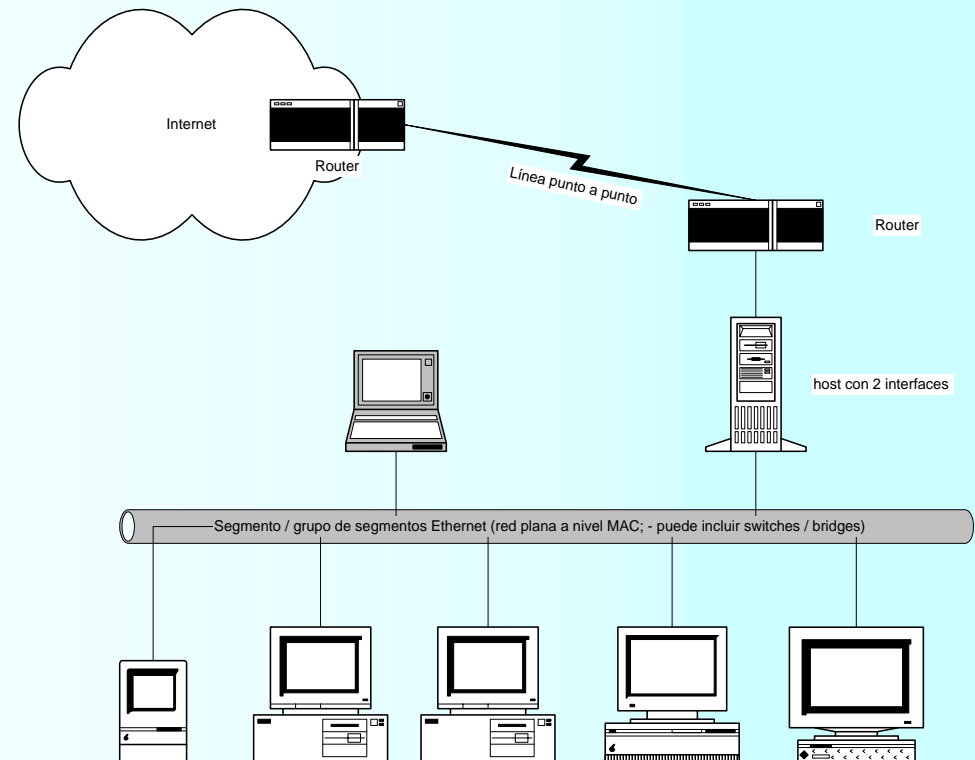
Subnet apantallada

Otros...



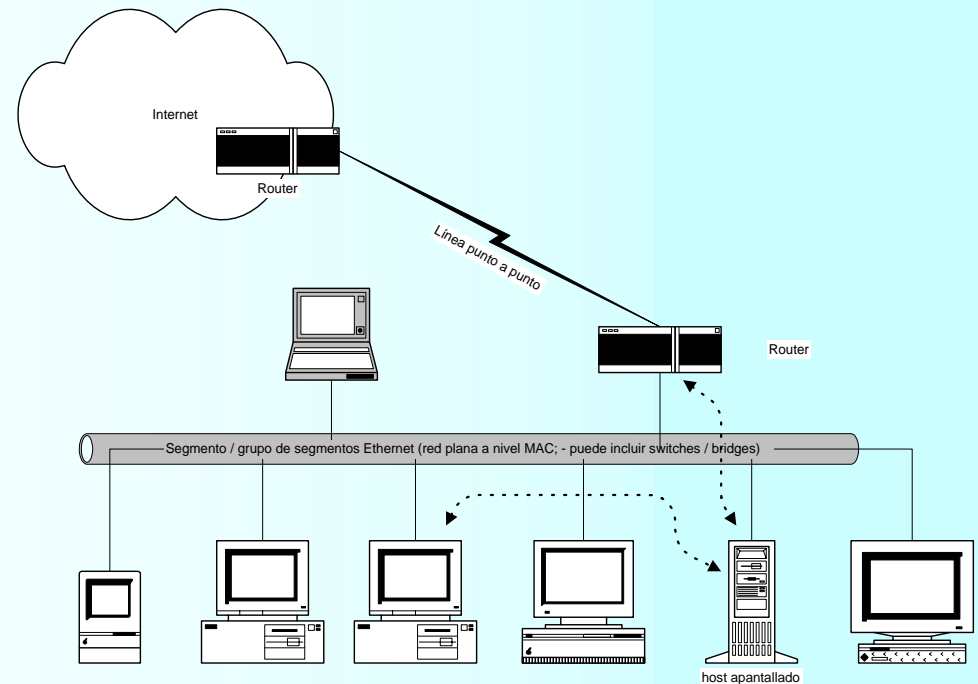
Host con dos interfaces

- routing deshabilitado (*ip forwarding off*)
- provee servicios vía proxies



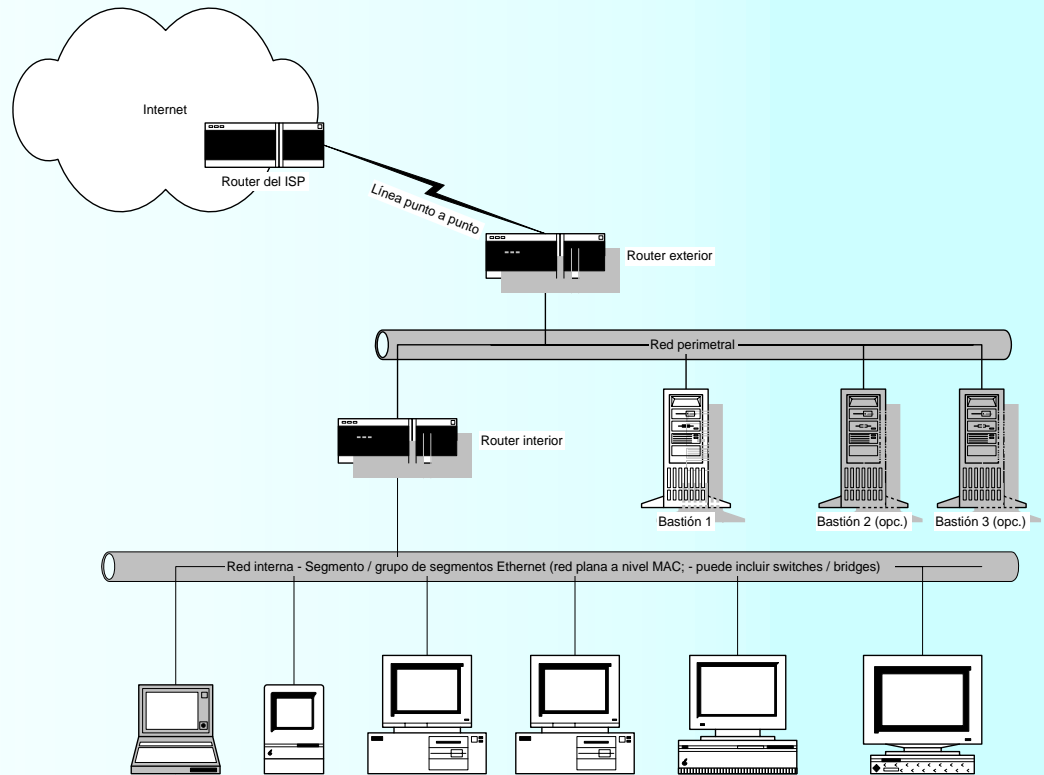
Host apantallado (screened host)

- Seguridad mediante packet filtering
- El router sólo permite acceso desde el exterior a algunos servicios del bastión exclusivamente
- Los clientes pueden tener acceso (restringible) directamente al exterior, o sólo a través del bastión

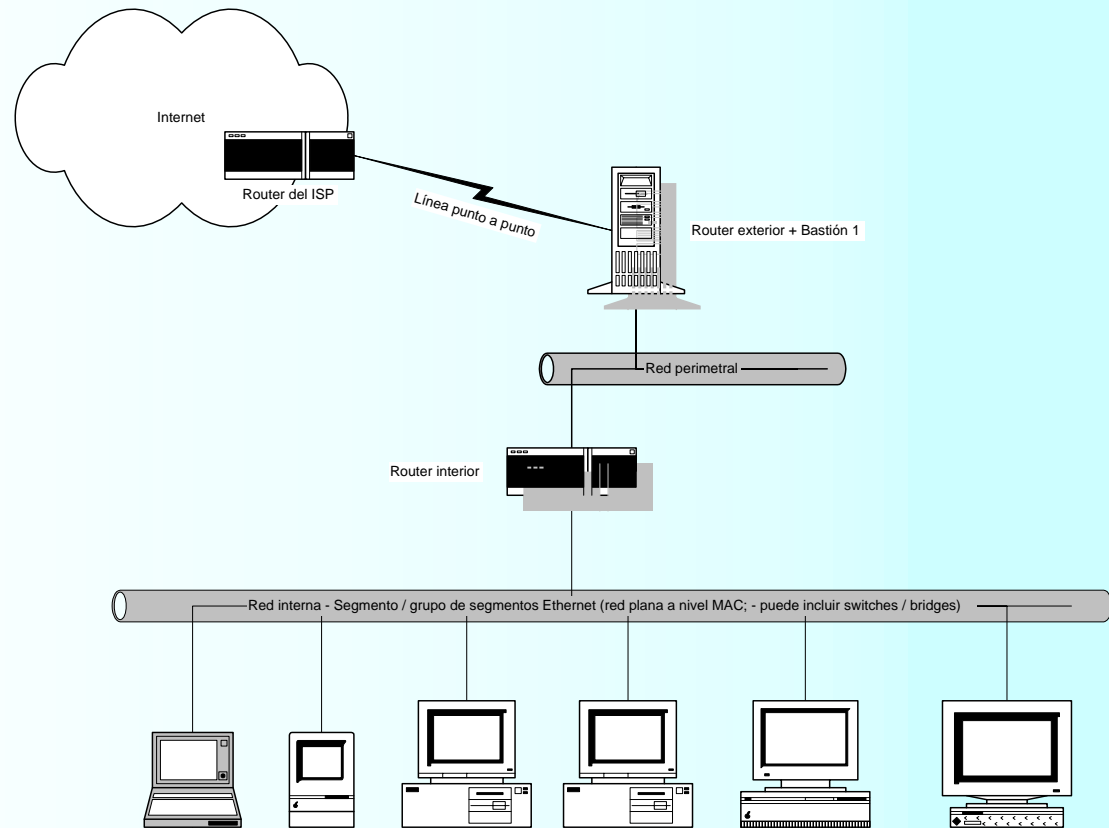


Subnet apantallada con uno o varios bastiones

- Se aísla el bastión sobre una red perimetral
- Más resistente ante eventual compromiso del bastión (ej. snooping)
- Los servicios más vulnerables se ubican afuera
- El router interior efectúa la mayor parte del filtrado

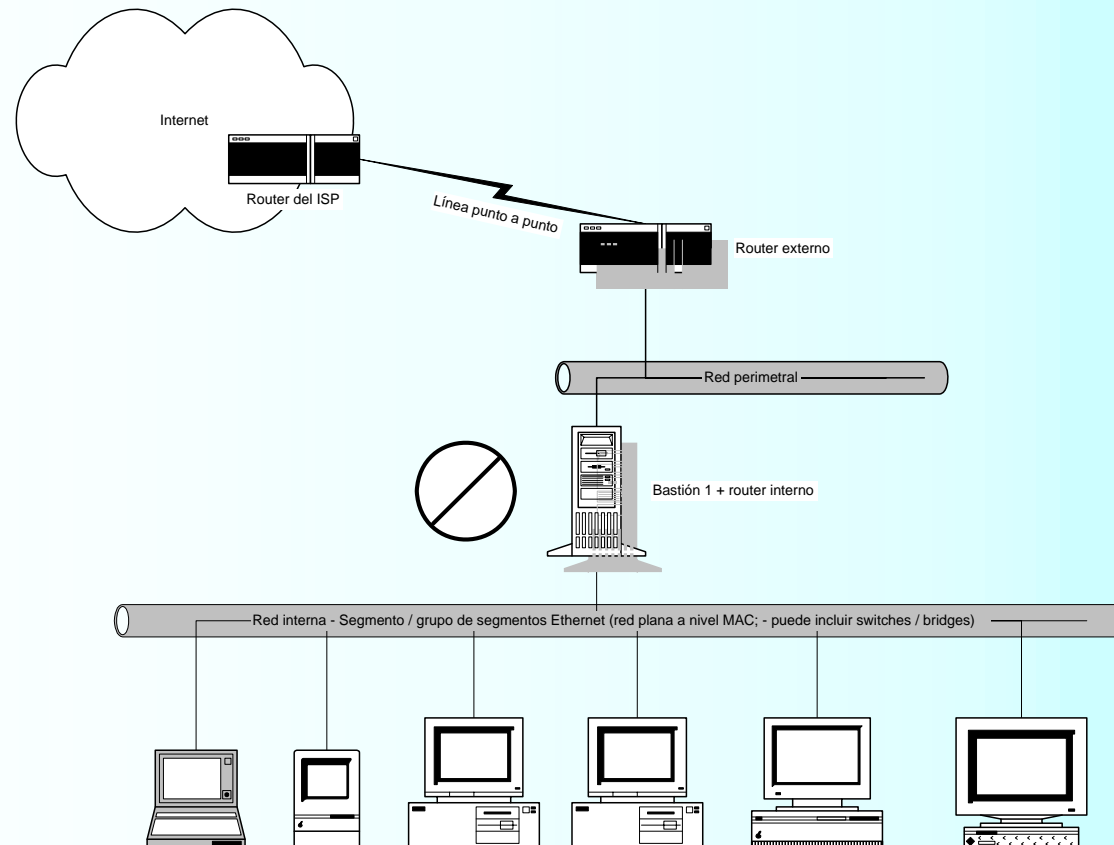


Combinación bastión con router externo

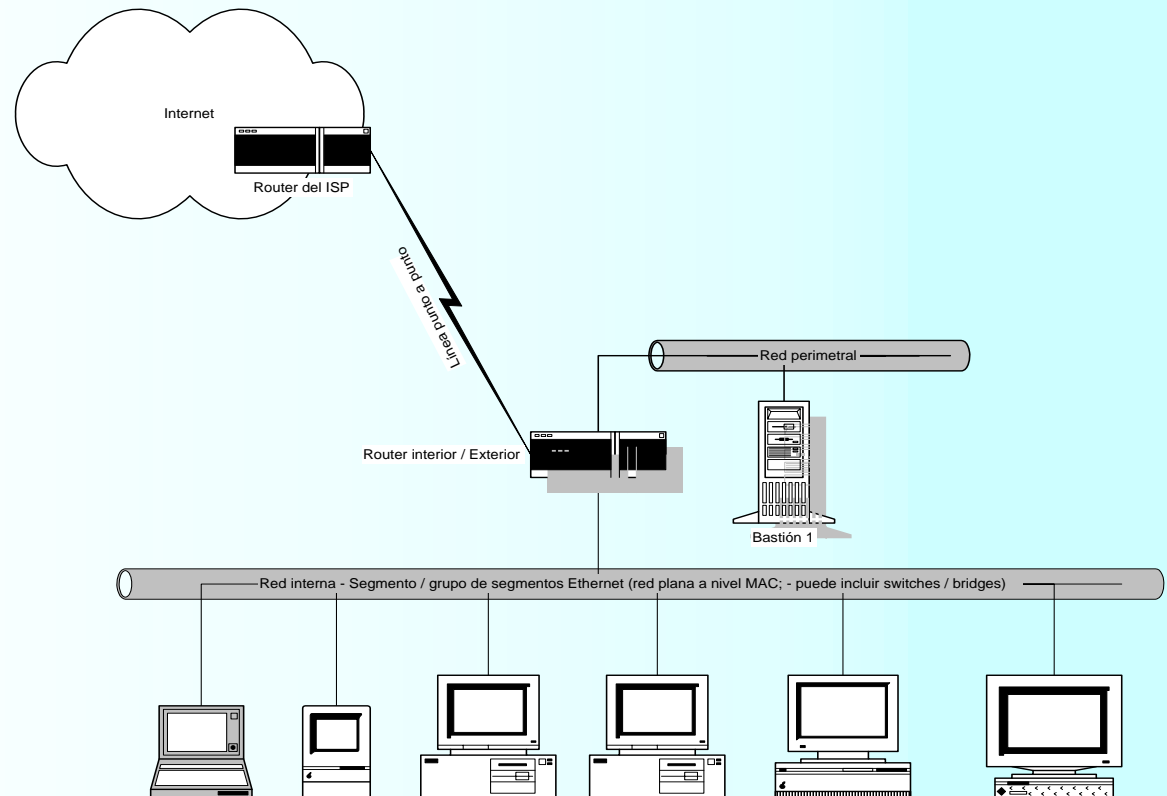


Combinación bastión con router interno

(no recomendado)



Subnet apantallada con un solo router



Proxies

(application and circuit level gateways)

El usuario se comunica con el proxy exclusivamente (no con el host de destino final).

El proxy obtiene el servicio, y lo hace disponible al cliente, quien lo utiliza sólo a través del proxy.

Los *application level proxies* son específicos para cada aplicación (Ej. telnet, ftp, http, etc.).

Circuit level proxy: no interpreta el protocolo aplicativo (es un proxy para TCP)



Proxies (cont.)

Buen nivel de protección. Permiten filtrar con relación a operaciones de la aplicación

(ej. ftp: copiar éste archivo particular)

Permiten *logging* a nivel de cada operación aplicativa

Menos eficientes

A veces requieren procedimientos especiales por parte de los clientes

Se pueden combinar con caching, pattern matching (ej. virus filters), compresión, etc.



Virtual Private Networks (VPN)

Internet: interconectividad “para el resto de nosotros”

Posibilidad de implementación de aplicaciones online sin requerir enormes inversiones en infraestructura

Amenazas:

disclosure: +

integrity: +

denial-of-service: -

Problemas remanentes:

disponibilidad (*)

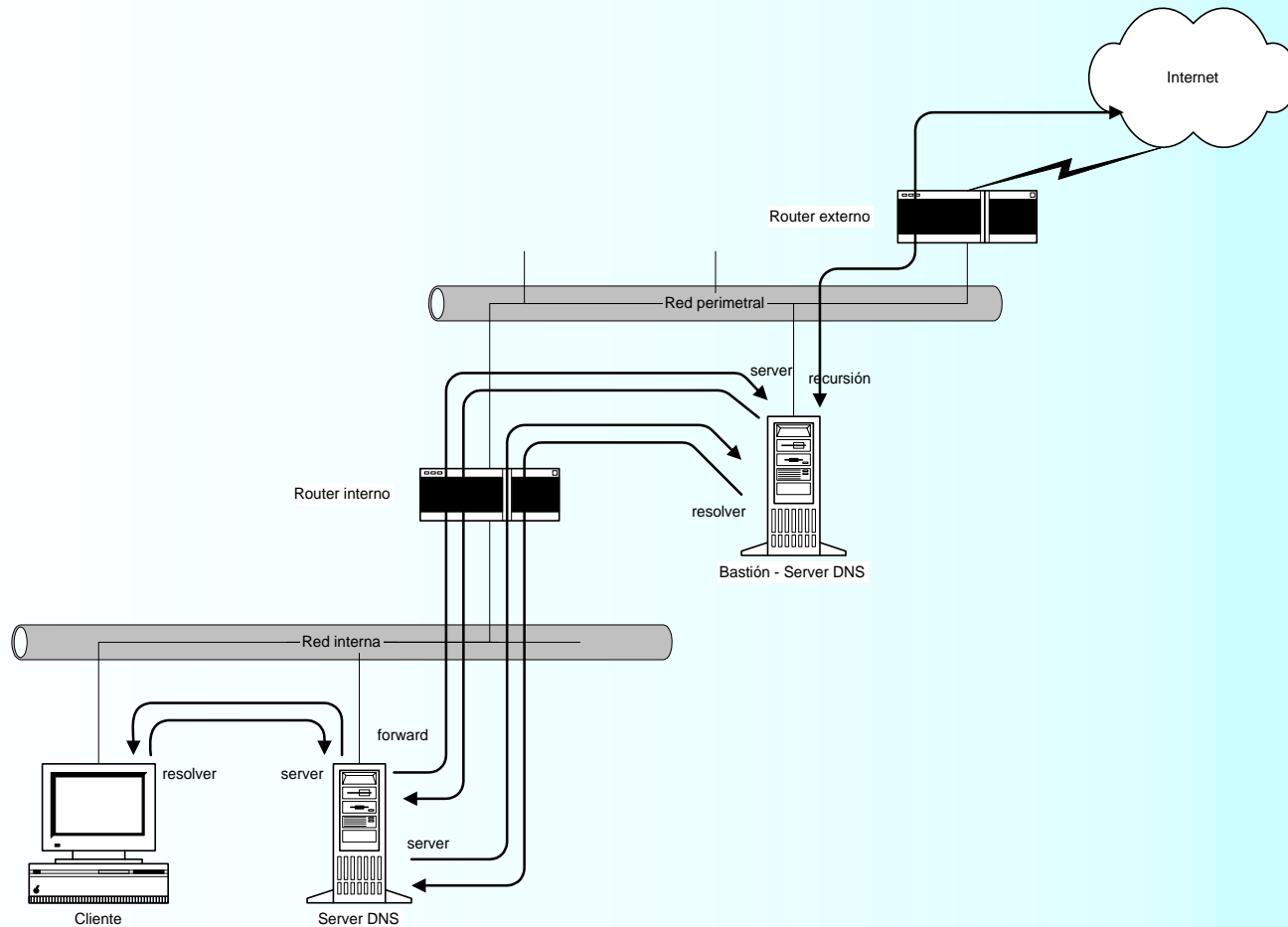
responsabilidad diluida

Otras consideraciones

- No suele ser necesario que los bastiones sean máquinas muy potentes (tener en cuenta vel. del enlace: ej.: max 200 pkts. TCP /s para 64 kbps)
- Empezar con instalaciones frescas, y hacer checklists antes de empezar
- Chequear bug releases
- Quitar TODO lo innecesario (ej. compiladores, intérpretes, editores, libraries, comandos, etc.). Si no fuese conveniente, montar las herramientas a demanda, desde un filesystem read-only o CDROM.
- En el firewall, NUNCA tener cuentas de usuario
- Política de backup cuidadosas, considerar el uso de medios read-only
- Escribir software para monitoreo automático

DNS

ocultamiento de hosts internos



Network Address Translation (NAT)

RFC 1597 - Direcciones de IP explícitamente fuera de la internet:

10.0.0.0 a 10.255.255.255

172.16.0.0 a 172.31.0.0

192.168.0.0 a 192.168.195.255

STATIC & HIDE



Registro de actividades (logging)

El valor está asociado con la relevancia, no necesariamente sólo con la cantidad

Justifica instalaciones especiales (log hosts, dispositivos no-reescribibles)

Combinar con herramientas de post-proceso

Combinar con alarmas automáticas y detectores de patrones

Un log por conveniencia, otro para catástrofes



Applications Security

SNMP y SNMPv2

Secure Sockets Layer SSL para TCP/IP
v.2 y v.3

Web Servers Seguros:

El protocolo HTTPS

Correo Electrónico Seguro:

PEM Privacy Enhanced Mail

PGP Pretty Good Privacy



SNMP y SNMPv2

SNMP no es seguro

ya que el community name
viaja en claro por la red

SNMPv2 ofrece

autenticación

integridad

privacidad

SNMP

Autenticado y control de acceso en SNMP

- Community names

- Authorized IP addresses

- MIB Views

 - Subset of manageable objects

- Community Profiles

 - View + Access Mode



SNMPv2

Derivado del S-SNMP

(Secure SNMP - RFC 1351/52/53)

Modelo administrativo en SNMPv2

(Autenticado , control de acceso, integridad y privacidad)

La aplicación de management en la NMS necesita realizar una determinada operación sobre una colección de objetos y para ello debe definir:

El Nivel de Autenticado y de Privacidad requerido

El Party que mínimamente satisface estos requisitos en relación con las Access Policies

