I

Introducción al Curso

II

Introducción a la Seguridad Informática y PenTesting

¿Qué es la Seguridad Informática?

Medidas y controles que garantizan la confidencialidad, integridad y disponibilidad de los activos del sistema de información, incluyendo hardware, software, firmware e información procesada, almacenada y comunicada.

Seguridad de la información vs Seguridad Informática

Seguridad de la Información

La protección de los sistemas de información y de información contra el acceso, uso, divulgación, interrupción, modificación o destrucción no autorizados a fin de proporcionar confidencialidad, integridad y disponibilidad.

¿Seguridad informática es igual a Seguridad de la información?

¿Seguridad informática mas importante que Seguridad de la información?

¿Seguridad informática es menos importante que Seguridad de la información?

Ética y Hacking

¿Ser «Hacker» es ético?

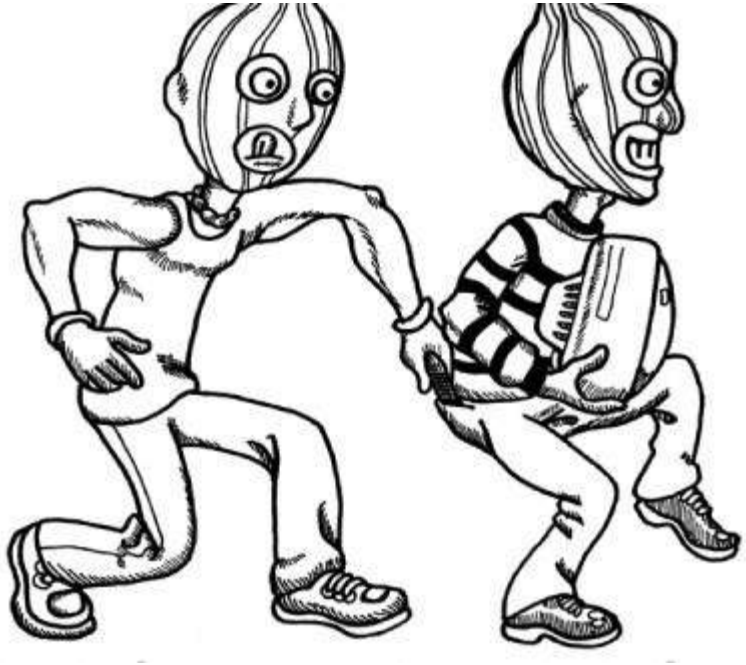
¿Qué es un Hacker?

¿Se emplea correctamente el termino hacker?

¿Ser «Hacker» es ético?

¿La Ética? ¿Ética profesional?

La Gran diferencia: **la autorización.**



¿Pero qué es un «Hacker»?

¿Pirata Informático?

¿Experto en Algo?

¿Cualquiera que programa es Hacker?

¿y los que no programan?

Hacker

significa apreciar la inteligencia juguetona.

‘‘Alguien puede programar sin ser juguetonamente inteligente y puede ser juguetonamente inteligente en otros campos sin programación.’’ - RMS

Hacktivismo

"El matrimonio del Hacking y el Activismo".

''sé curioso , lee mucho, Trata nuevas cosas. creo que lo que mucha gente llama inteligencia solo se reduce a la curiosidad''- Aaron Swartz

Security Hacker

Alguien que busca y explota las debilidades en un sistema informático o red informática.

Clasificación:

Hacker de Sombrero Negro

Hacker de Sombrero Gris

Hacker de Sombrero Blanco

Otras Clasificaciones:

- ***Hacker Suicida***
- ***Script Kiddie***
- ***Hacker espía***
- ***Terrorista cibernético***
- ***Hacker patrocinado por el estado***

Penetration Testing

También conocido como:

- *Pen Testing*
- *PT*
- *Hacking*
- *Ethical Hacking*
- *White Hat Hacking*

Fases de un Pentest

- *Reconocimiento*
- *Escaneo*
- *Ganando Acceso (Explotación)*
- *Manteniendo el Acceso*
- *Borrado de Huellas*

Metodologías de un Pentest

- *Black Box*
- *White Box*
- *Gray Box*

Vectores de Ataques

- *APT (Amenaza avanzada persistente)*
- *Botnets*
- *Cloud Computing*
- *Ataques internos*
- *Amenazas Mobiles*

Vocabulario

Términos Importantes

- **¿Qué es un activo?**
- **¿Qué es una amenaza?**
- **¿Qué es una vulnerabilidad?**
- **¿Qué es un riesgo?**
- **¿Qué es un Exploit?**

III

Kali Linux

IV

Configuración del laboratorio de pruebas de concepto

V

Bases de Networking y GNU/Linux

Introducción a GNU/Linux

El Shell: Comando esenciales

Moviéndonos por el sistema de ficheros:
mkdir, ls, cd, pwd

Manipulación:
rm, rmdir, cp, mv

La ayuda del sistema
man, apropos, info, whatis

Introducción a GNU/Linux

El Shell: Comando esenciales

Patrones

Metacarácter	Descripción
?	Comodín a cualquier carácter simple
*****	Iguala la secuencia de 0 o mas caracteres
[]	Designa un carácter o rango de caracteres que son igualados por un simple carácter.
{}	Abreviar conjuntos de palabras que comparten partes comunes
~	Ruta absoluta directorio home

Introducción a GNU/Linux

Búsqueda
find, locate, whereis

Tipos, contenidos y comparación de ficheros
file, cat, less, more, hexdump, od, grep, cut, paste, head, tail, wc, diff ,cmp, comm

Permisos
chown, chgrp, unmask

Empaquetado y compresión de archivos
tar, gzip, bzip2

Introducción a GNU/Linux

Metacaracteres Sintácticos

Metacarácter	Descripción
;	Separador entre órdenes que se ejecutan secuencialmente
|	Separación entre órdenes que forman parte de un cauce (pipeline).
( )	Se usan para aislar ordenes separadas por ; ó |. Las ordenes dentro de los paréntesis, ejecutadas en su propio shell, son tratadas como una única orden.
&	Indicador de trabajo en segundo plano (background).
||	Separador entre órdenes, donde la orden que sigue al || sólo se ejecuta si la orden precedente falla.
&&	Separador entre ordenes, en la que la orden que sigue al && se ejecuta sólo si la orden precedente tiene éxito.

Órdenes para el control de trabajos

Jobs, fb, gb, %, kill

Redirecciones y Tuberías

Metacaracteres de entrada/salida o de dirección

Metacarácter	Descripción
< nombre	Redirecciona la entrada de una orden para leer del archivo nombre.
> nombre	Redirecciona la salida de una orden para escribir en el archivo nombre. Si nombre existe, lo sobrescribe.
2> nombre	Redirecciona el error (stderr) a un fichero. Si nombre existe, lo sobrescribe
>& nombre	La salida de stderr se combina con stdout, y se escriben en nombre.
>> nombre	La salida de la orden se añade al final del archivo nombre.
2>> nombre	La salida de stderr se añade al final del archivo nombre
>>& nombre	Añade la salida de stderr, combinada con stdout y las añade al final de nombre.
|	Crea un cauce entre dos órdenes.
|&	Crea un cauce entre dos ordenes, con las salidas de stderr y stdout de la orden de la izquierda combinadas y conectadas con la entrada de la orden de la derecha.

Bash Scripting

Nuestro primer shell script

Hola.sh

#!/bin/bash
echo "hola"

¿Qué significa #!?

Variables

Tienen un nombre y un valor

FECHA="29/12/2016"

echo "hoy es $FECHA"

Se asigna valores con "="

Variables

Exportación de Variables

Export FECHA="29/12/2016"

```
#!/bin/bash
echo "hoy es $FECHA"
FECHA="30/12/2016"
echo "la nueva fecha $FECHA"
```

Variables

Declaración de variables

```
#!/bin/bash
declare -i inttest
inttest=123
echo $inttest
inttest=12.3
echo $inttest
declare -r rotest=281
rotest=212
```

-r lectura única
-i variable entera
-u convertir a mayúscula
-l convertir a minúscula
-x declara y exporta

Variables

Variables Globales y Locales

Variables Globales llamadas Variables de Entorno

printenv *se utiliza para mostrar todas las variables de entorno*

```
#!/bin/bash
VAR="variable global"
bash() {
   local VAR="variable local"
   echo $VAR
}
bash
echo $VAR
```

Variables

Interactividad

```
#!/bin/bash
echo "Dime tu nombre"
read NOMBRE

echo "Tu nombre es $NOMBRE"
```

Variables

Argumentos

#!/bin/bash

echo "Tu nombre es $1"

$1,$2,$3,$4……${10} argumentos
$0 nombre del script

shift

Variables

Argumentos Especiales

$# número de argumentos que nos han pasado
$* todos los argumentos
$@ todos los argumentos
$_ comando anteriormente ejecutado
$$ PID del propio proceso shell

Variables

Sustitución de comandos

Dos Sintaxis:

LISTADO=`ls`

LISTADO=$(ls)
LISTADO=$(ls $(cat directorios.txt))

Variables

Operaciones aritmeticas con expr

SUMA=`expr 7 + 5`
echo $SUMA

Variables

Control de flujo
Condiciones: test ó []

test "$NOMBRE" == "COKO" (==,!=,>,<,>=,<=)
["$DINERO" –eq "1000"] (-eq,-ne,-gt,-lt,-ge,-le)
test –f /etc/passwd (-f,-d,-L,-r,-w,-x)

Modifica el valor de $?

cero = verdadero
No cero = falso

Declaraciones condicionales

```
if comando_if
then
  comandos_then
elif comando_elif
then
  comandos_elif
else
  comandos_else
fi
```

Declaraciones de casos

```
case $VARIABLE in
  “VALOR1”) comandos_valor1
    ;;
  “VALOR2”) comandos_valor2
    ;;
  *) comandos_default;
esac
```

Bucles while, until y for

While

while comando
do
  comandos
done

Bucles while, until y for

Until

until comando
do
  comandos
done

Bucles while, until y for

for

for VARIABLE in LISTA
do
  comandos
done

Bucles while, until y for

For estilo C

for ((VARIABLE_INICIADA ; MIENTRAS; CONTADOR))
do
  comandos
done

select

select VARIABLE in LISTA
do
  comandos
done

Funciones

- *Podemos modularizar los scripts agrupando tareas en funciones.*
- *Es necesario que una función esté definida ANTES de que sea llamada.*
- *Dentro de una función, $1, $2, $3, etc. serán los parámetros pasados a la función, no al script en sí*

source

source funciones.sh
. funciones

Modelo OSI

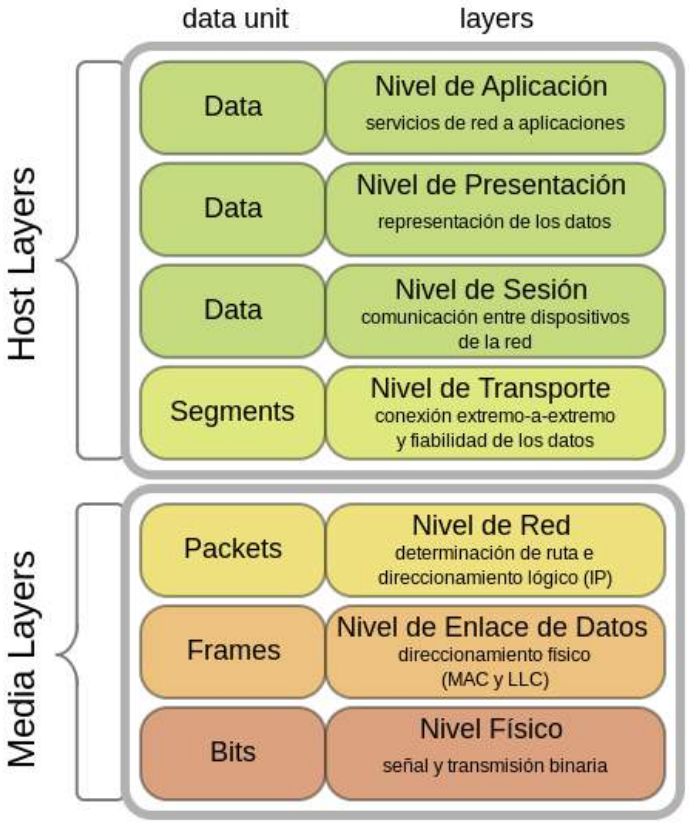


Figura 1. fuente Wikipedia

Transport Control Protocol /Internet Protocol

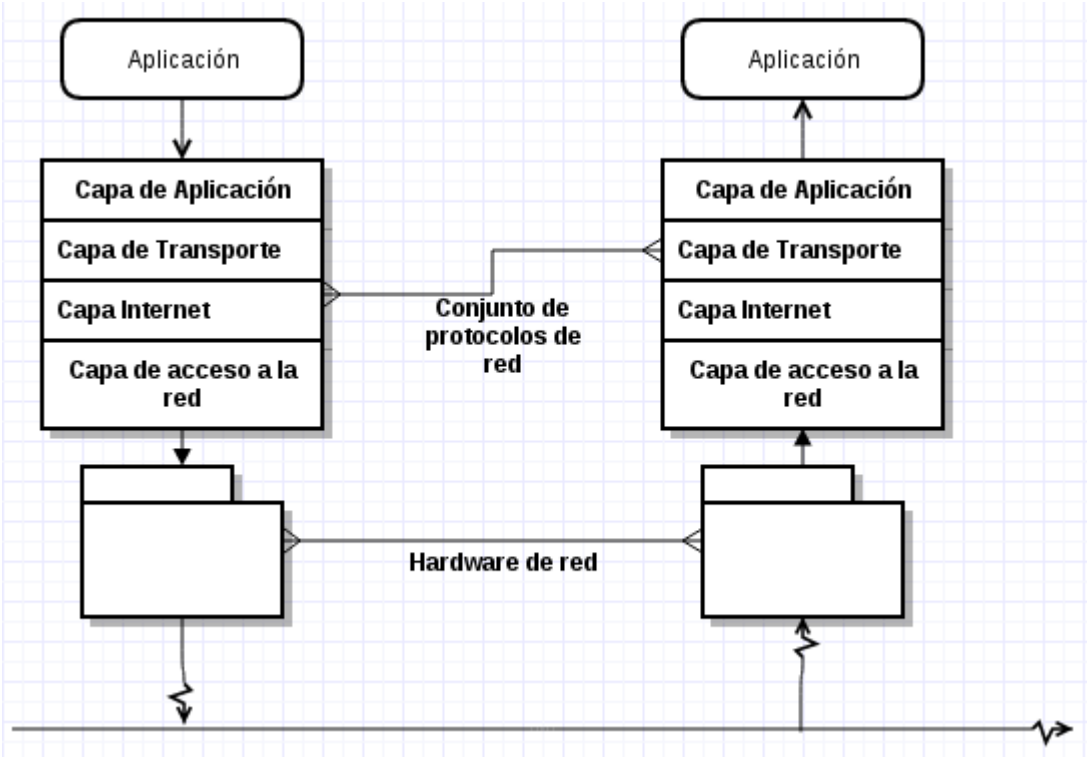
Una colección de protocolos que soporta comunicaciones en Red

Redes Y Protocolos

Una **Red** es una computadores of dispositivos que pueden comunicarse a través de un medio de transmisión común

Un **Protocolo de Red** es un sistema de reglas comunes que ayuda a definir el complejo proceso de comunicación en red.

El papel de una suite de protocolo de red.



Internet Protocol (IP)

- Protocolo de capa de red cuyos trabajos son enviar paquetes o datagramas de un punto a otro
- Cada destino se especifica mediante una dirección IP
- Direcciones IP: Estáticas o Dinámicas

Características IP

- IP es un protocolo sin conexión
- IP es un protocolo poco fiable
- Los paquetes IP no se identifican como parte de una secuencia o pertenecen a un determinado trabajo

Composición Cabecera Paquete IP

Formato de la Cabecera IP (Versión 4)

0-3	4-7	8-15	16-18	19-31
Versión	Tamaño Cabecera	Tipo de Servicio	Longitud Total	
Identificador			Flags	Posición de Fragmento
Time To Live		Protocolo	Suma de Control de Cabecera	
Dirección IP de Origen				
Dirección IP de Destino				
Opciones				Relleno

Dirección IP

- ¿Qué es una dirección IP?

La máscara de red y su aplicación en redes

- Las máscaras de red principales son:

- Para redes de clase C: 255.255.255.0
- Para redes de clase B: 255.255.0.0
- Para redes de clase A: 255.0.0.0

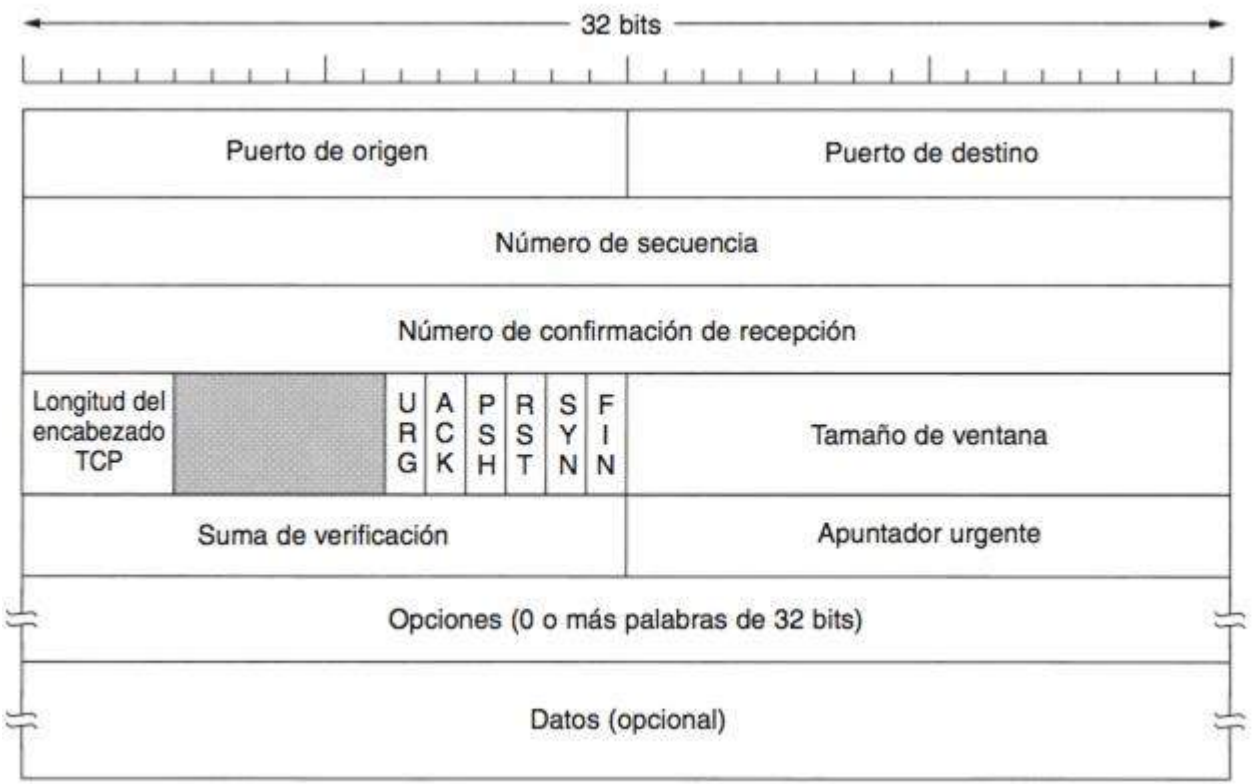
ARP & ICPM: Protocolos de acompañamiento

- Address Resolution Protocol (ARP) detecta la dirección física que corresponde a una dirección IP
- Internet Control Message Protocol: Define el formato de los mensajes de control que se envían al remitente indicando que se ha producido un problema

Transmission Control Protocol

- Tcp es un protocolo orientado a la conexión
- Tcp proporciona fiabilidad
- Tcp garantiza que los datos que llegan fuera de secuencia se vuelven a poner en orden
- Tcp también implementa control de flujo, por lo que un remitente no puede abrumar a un receptor con datos

Composición del segmento TCP



¿Qué es un puerto?

- Múltiples aplicaciones o protocolos de capa superior pueden usar tcp simultáneamente

- Puertos típicos:
  - 20/21 FTP
  - 22 SSH
  - 23 Telnet
  - 25 Simple Mail Transfer Protocol
  - 37 Time
  - 53 Domain Name System
  - 80 HTTP
  - 110 POP3
  - 443 HTTPS

Comunicación TCP: Hacer una conexión

- Se debe establecer una conexión antes de enviar cualquier dato
- Los segmentos sólo se envían entre cliente y servidor si hay datos para fluir.

Comunicación TCP: Transmisión de datos

- TCP es un protocolo de ventana deslizante, y no espera a que se reconozca
- Para evitar el desbordamiento del búfer del receptor…
- Para la eficiencia…

Comunicación TCP: Error de corrección

- En situaciones de error TCP puede…
- Datos perdidos o dañados.
- Bloqueo del flujo

Comunicación TCP: Cierre de la comunicación

- Cada dirección del flujo de datos debe cerrarse por separado.

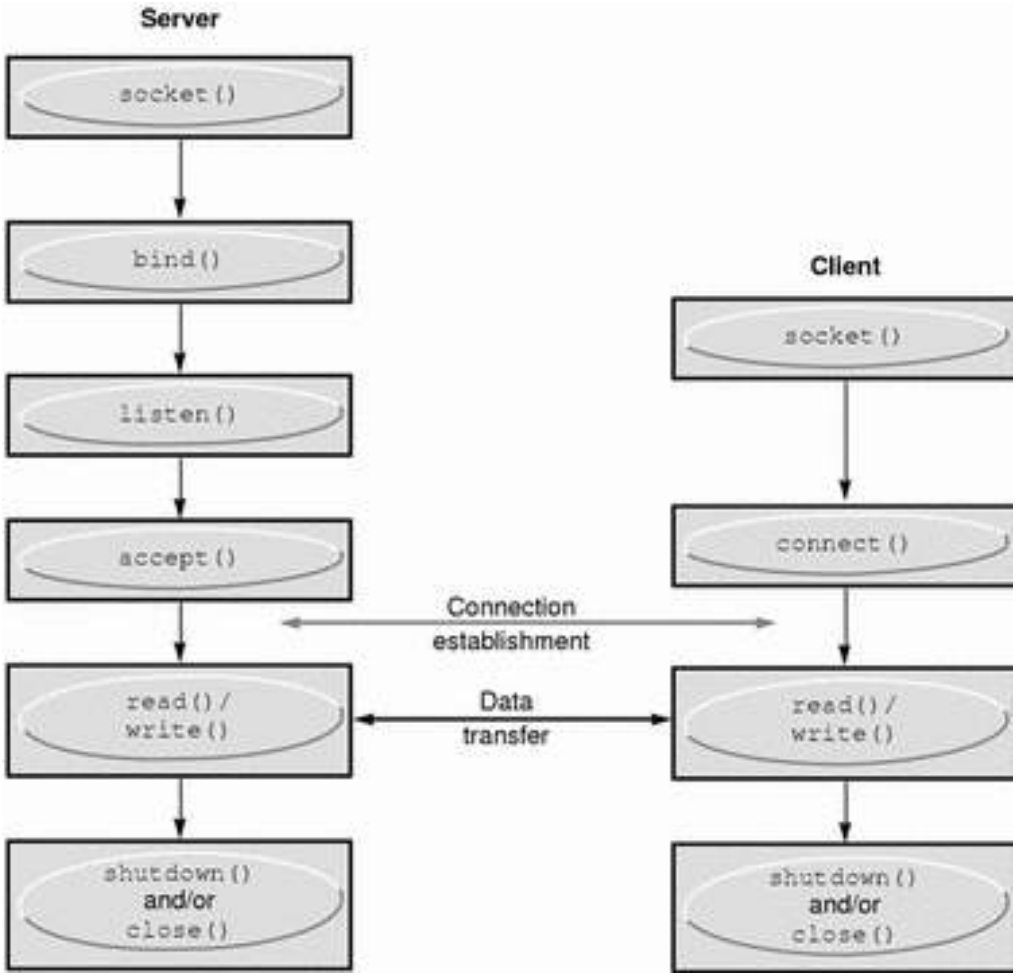
User Datagram Protocol (UDP)

- UDP es un protocolo simple
- UDP no es confiable y sin conexión
- La función principal es especificar los protocolos de la capa superior
- Útil para broadcasting ya que no requiere una conexión

Composición del segmento UDP

bits	0 – 7	8 – 15	16 – 23	24 – 31
0	Dirección Origen			
32	Dirección Destino			
64	Ceros	Protocolo	Longitud UDP	
96	Puerto Origen		Puerto Destino	
128	Longitud del Mensaje		Suma de verificación	
160	Datos			

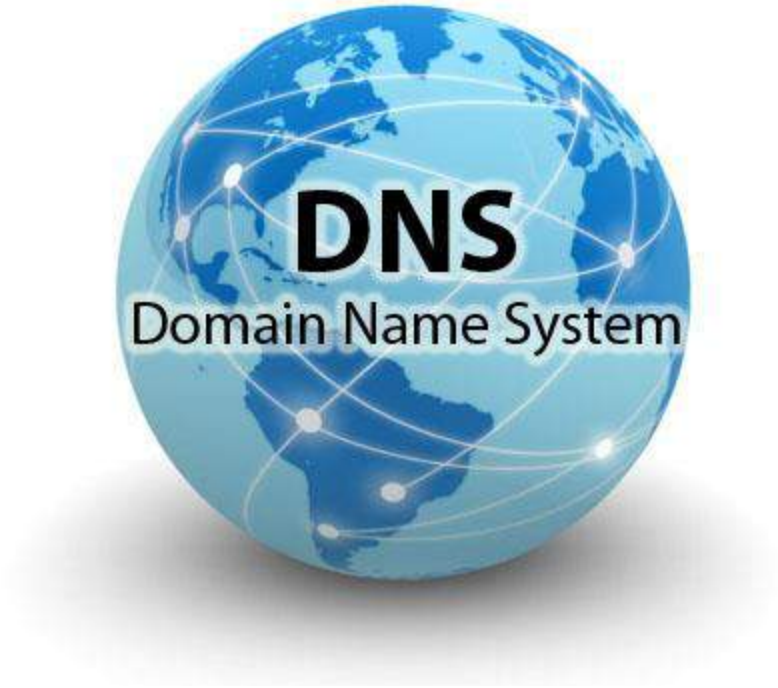
El interfaz Socket



Paradigma de E/S Unix y E/S de la RED

LA ABSTRACCIÓN DE SOCKET

SISTEMA DE NOMBRE DE DOMINIO (DNS)



VI

Herramientas Esenciales

VII

Recopilación de Información

VIII

Pruebas Básicas de Penetración de Red