Guerrero Hernández Aarón

Robles Ríos Rafael

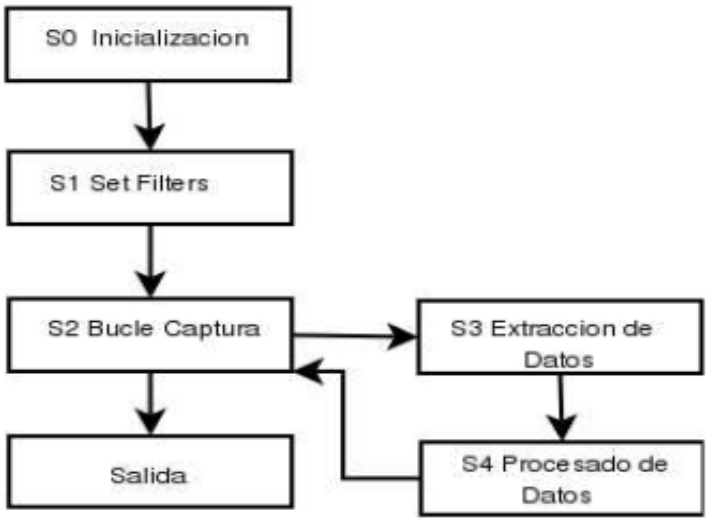
Reporte Practica 04

Redes de Computadoras

Describe cómo funciona la biblioteca pcap

Libpcap es una biblioteca open source escrita en C que ofrece al programador una interfaz desde la que capturar paquetes en la capa de red, además Libpcap es perfectamente portable entre un gran número de sistemas operativos

Esquematización de un programa:



Como puede verse en el gráfico, la primera fase de nuestro programa es la Inicialización. Esta engloba las funciones capaces de obtener información del sistema: Las interfaces de red instaladas, las configuraciones de estas interfaces (Mascara de Red, Dirección de Red) etc., a continuación, se recogen las funciones más relevantes.

Funciones Específicas:

```
char *pcap lookupdev(char *errbuf)
```

Devuelve un puntero al primer dispositivo de red válido para ser abierto para captura, en caso de error se devuelve NULL y una descripción del error en errbuf

.

```
int pcap lookupnet(char *device, bpf u int32 *netp,bpf u int32 *maskp, char errbuf)
```

Una vez obtenido el nombre de una interfaz valida, podemos consultar su dirección de red (que no su dirección IP) y su máscara de subred. Device es un puntero a un array de caracteres que contiene el nombre de una interfaz de red valida, netp y maskp son dos punteros a bpf u int32 en los que la función dejaŕ a la dirección de red y la máscara respectivamente. En caso de error la función devuelve -1 y una descripción del error en errbuf.

```
int pcap findalldevs(pcap if t **alldevsp, char *errbuf)
```

Esta función nos devuelve todas las interfaces de red que pueden ser abiertas para capturar datos, puede que existan mś as interfaces en el sistema pero que por una razón u otra no puedan ser abiertas para captura (falta de permisos.

Ejemplo:

```
/*************************************************************************
*
* Fichero: lsdevs.c
* Fecha:  Alejandro Lopez Monge
* Orignal: Martin Casado http://www.cet.nau.edu/~mc8/Socket/Tutorials/section1.html
*
* Compilacion: gcc lsdevs.c -lpcap
*
* Descripcion:
*  Buscamos la primera interfaz de red disponible y lista su direccion de red
*  (que no su direccion IP) y su mascara de subred.
*
*************************************************************************/

#include <stdio.h>
#include <stdlib.h>
#include <errno.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <pcap.h>                       //include a libpcap

int main(int argc, char **argv)
{
  char *net;                            // direccion de red
  char *mask;                           // mascara de subred
  char *dev;                            // nombre del dispositivo de red
  int ret;                              // codigo de retorno
  char errbuf[PCAP_ERRBUF_SIZE];        // buffer para mensajes de error
  bpf_u_int32 netp;                     // direcion de red en modo raw
  bpf_u_int32 maskp;                    // mascara de red en modo raw
  struct in_addr addr;

  if((dev = pcap_lookupdev(errbuf))== NULL) //conseguimos la primera interfaz libre
  {printf("ERROR %s\n",errbuf);exit(-1);}

  printf("Nombre del dispositivo: %s\n",dev); //mostramos el nombre del dispositivo

  if((ret = pcap_lookupnet(dev,&netp,&maskp,errbuf))==-1) //consultamos las direccion de red y las mascara
  {printf("ERROR %s\n",errbuf);exit(-1);}

  addr.s_addr = netp;                   //Traducimos la direccion de red a algo legible
  if((net = inet_ntoa(addr))==NULL)
  {perror("inet_ntoa");exit(-1);}

  printf("Direccion de Red: %s\n",net);

  addr.s_addr = maskp;                  //Idem para la mascara de subred
  mask = inet_ntoa(addr);

  if((net=inet_ntoa(addr))==NULL)
  {perror("inet_ntoa");exit(-1);}

  printf("Mascara de Red: %s\n",mask);
  return 0;
}
```

Vulnerabilidades del protocolo HTTP

El US-CERT advierte sobre sitios Web que transmiten señales de autenticación sin encriptar. Debido a ello, las cookies utilizadas por algunos servicios de Internet, pueden ser interceptadas por un atacante. Algunos sitios Web transmiten esta autenticación sin ninguna clase de cifrado durante la sesión entera, aún cuando éstas son iniciadas sobre una sesión HTTP encriptada. Este comportamiento, podría permitir a un atacante en la red, usurpar las credenciales de un usuario legítimo. Sitios que aplican cookies de autenticación sobre un inicio de sesión HTTPS (protocolo seguro), y luego transmiten las cookies sobre HTTP (protocolo no seguro), son particularmente vulnerables, puesto que es más probable que los usuarios piensen que la seguridad de la página a la que están conectados se aplica a toda la sesión. Las cookies HTTP son textos que envía el servidor Web al navegador. Las cookies se transmiten después al servidor cuando el navegador tiene acceso al sitio Web. Algunos sitios pueden autenticar a un usuario con un nombre y contraseña y crear una cookie con un identificador único, para responder a las futuras peticiones de autenticación. Para aumentar la seguridad, el sitio Web puede borrar la cookie cuando el usuario abandona la sesión habilitando el atributo opcional "Secure" en el cabezal de respuesta "Set-Cookie", o haciendo que la cookie expire después de un tiempo específico.

Barras de herramientas o extensiones utilizadas por los navegadores, pueden también enviar credenciales de autenticación (cookies) a los sitios o servicios Web. Sitios que utilizan cookies para autenticación sobre protocolos de texto plano tales como HTTP, son vulnerables a que ésta autenticación sea interceptada, simplemente accediendo al tráfico que transporta a la cookie. Luego de ello, el atacante podría utilizarla como credencial de autenticación, tomando cualquier clase de acción en el sitio Web, como si se tratara del propio usuario. Son afectados especialmente aquellos sitios que ofrecen un software como servicio. El cifrado nulo es una opción válida al usar HTTPS, de acuerdo a las especificaciones originales del SSL (Secure Socket Layer, la tecnología que utiliza criptografía para cifrar los datos que se intercambian con un servidor seguro). Para minimizar los riesgos, se aconseja al usuario que inicia sesión en una dirección HTTPS://, observar que la misma se mantenga así y no cambie a HTTP:// mientras se navega. También es aconsejable al abandonar una sesión, utilizar la opción correspondiente en el sitio Web, y no solo cerrar la ventana o acceder a otra dirección (sitio) de Internet. Esto disminuye las posibilidades de que un atacante obtenga las credenciales del usuario y explote la vulnerabilidad.

Ataque a protocolos HTTP

DNS Spoofing

Cambiar la configuración de DNS y definir otro servidor de DNS, uno que las asociaciones entre nombres de dominios e IPs las determine un atacante malintencionado para un fin específico.

Ataque falsificar certificados

En las conexiones HTTPS, un navegador solo mostrará sin alarmas la página Web si ésta incluye un certificado de seguridad firmado por una AC, una agencia independiente que garantiza la autenticidad de la propia página.

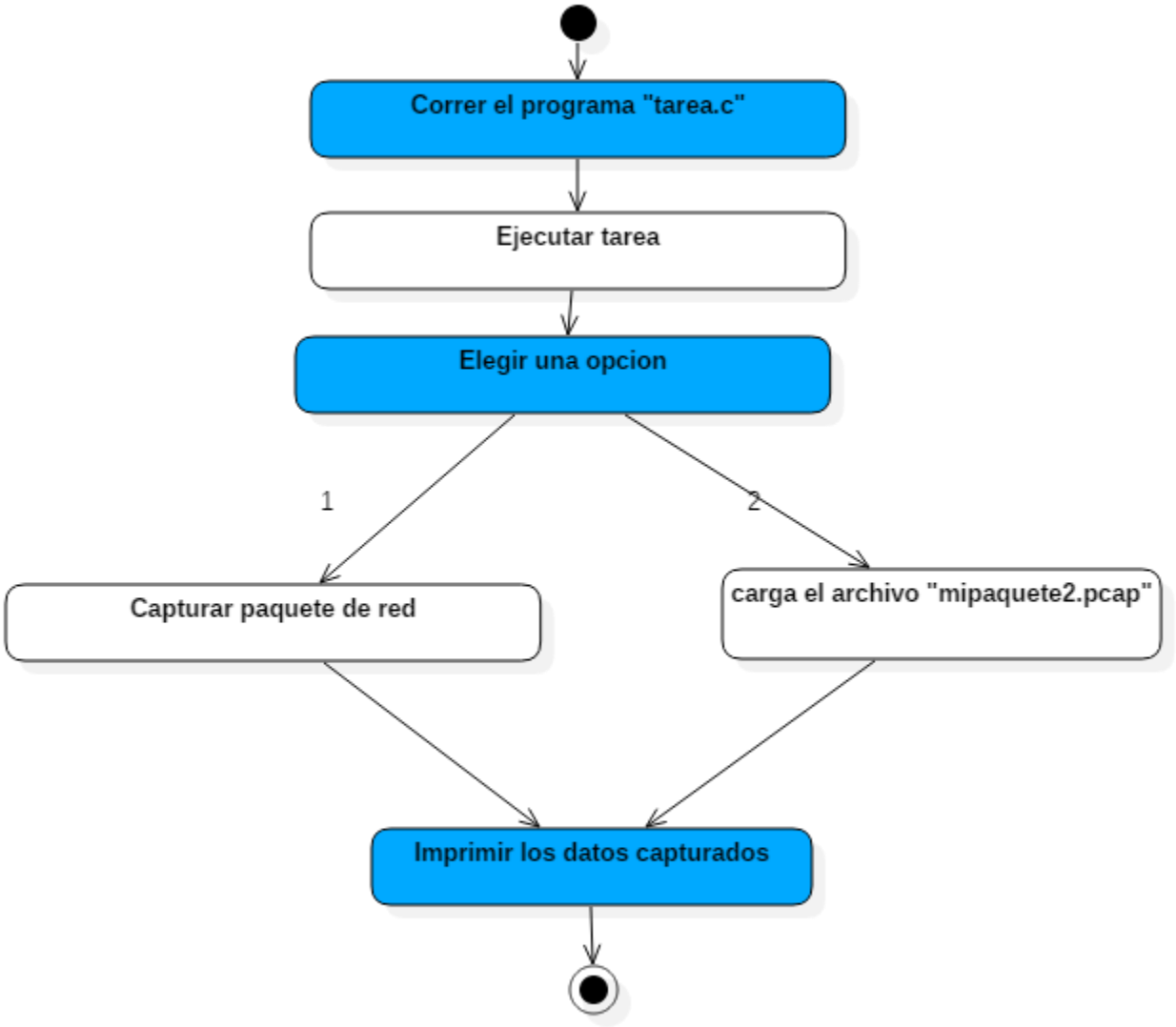
Ataque de dirección falsa

Este ataque es bastante simple e imaginativo. Debido a que la mayoría de los usuarios visitan páginas no protegidas con SSL antes de ser llevados a páginas cifradas con SSL/TLS, un atacante puede evitar fácilmente que éstos usen el cifrado para capturar así todo su tráfico.

Ataque de dirección parecida

Todos hemos ido alguna vez a la página equivocada gogole.com en vez de google.com. En el pasado, las cosas eran relativamente simples: Si un atacante quería registrar un dominio que imitaba a uno real, le bastaba con saltarse o añadir alguna letra.

Flujo del Programa



Principales Problemas y como se solucionaron

Como identificar las cabeceras y se solucionan importando la paquetería

Problemas que no fueron solucionados

No es posible cargar las direcciones correspondientes en todos los puertos