Capítulo 3
Un esquema integral de seguridad: ASIS

"No se puede confiar en código no creado en su totalidad por uno mismo. (Sobre todo en código de compañías que emplean a personas como yo.)"

Ken Thompson. Creador de UNIX

En este Capítulo se describe la síntesis de este trabajo: por un lado *el Hacker's work bench* como esquema de detección de vulnerabilidades de la red UDLA y por otro, el *Administrator 's work bench* como esquema de prevención de ataques. Se explican las partes que conforman el esquema de seguridad propuesto, su implantación, sus objetivos,y las necesidades que cubren.

Diseño y **A**plicación de un **S**istema **I**ntegral de **S**eguridad informática para la UDLA.

Después de un extenso análisis de las áreas que componen la seguridad informática, ASIS toma cuerpo con aquellas áreas que cubren la necesidad de un esquema integral para la UDLA. ASIS se forma de dos partes principales: ***hacker´s work bench*** y ***administrator´s work bench***. El primero es un conjunto de herramientas que se utilizan para conocer las vulnerabilidades de la red UDLA. El segundo es un conjunto de herramientas para la prevención y corrección de dichas vulnerabilidades a nivel Internet e Intranet. (Ver Figura 3.1)

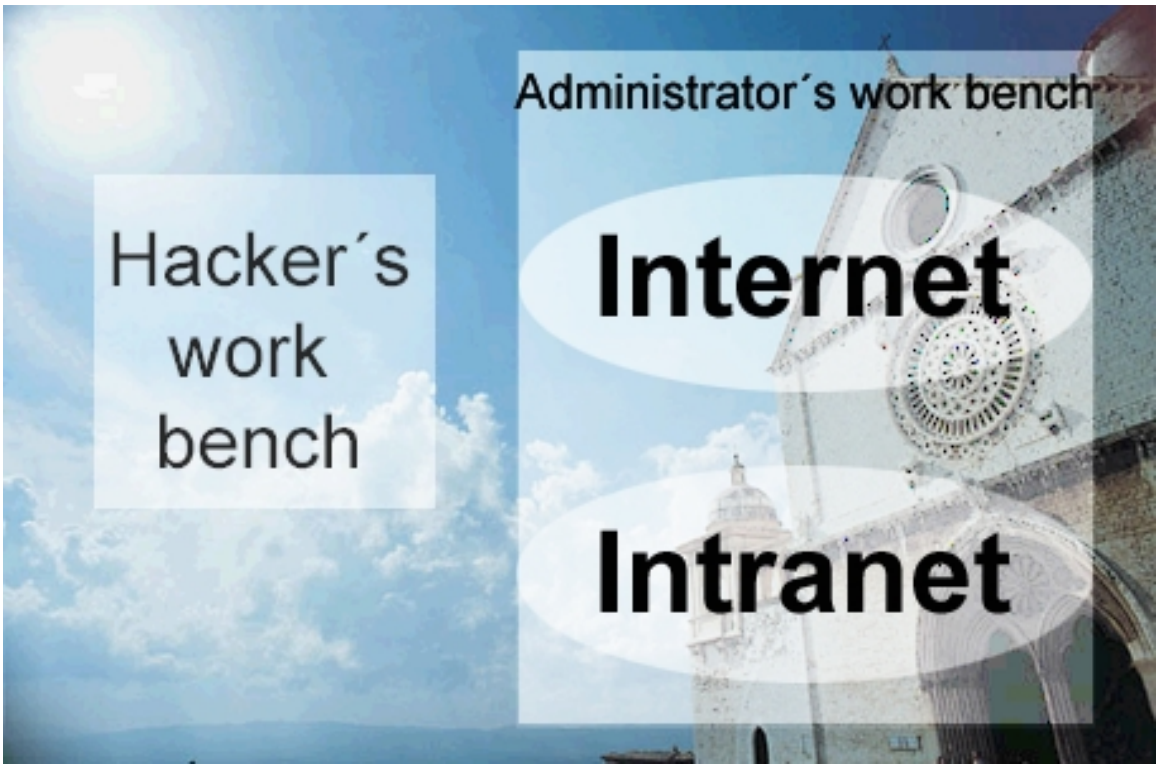

Figura 3.1 Esquema general de ASIS.

3.1 Hacker's work bench.

Hacker's work bench es un conjunto de herramientas para detectar vulnerabilidades en nodos y en la red UDLA en general. Está compuesto por los siguientes módulos:

- Crack_UDLA & YPX.
- Nessus_UDLA.
- Ataca.

3.1.1. Crack_UDLA & YPX

Crack, desarrollado por el experto en seguridad Alec Muffet [VIL00], es el "adivinador" de passwords más utilizados en entornos Unix; actualmente se encuentra en su versión 5, que funciona en la mayoría de los clones del sistema operativo. Ejecutar Crack sobre el archivo de passwords es recomendable para cualquier administrador mínimamente preocupado por la seguridad, sin importar que se utilicen mecanismos que obliguen a los usuarios a elegir passwords aceptables.

Este programa realiza una primera pasada sobre el archivo de passwords intentando adivinarlos con base a la información de cada usuario almacenada en el archivo; la cual es utilizada frecuentemente como password (nombre, apellido).
Posteriormente entran en juego diccionarios para continuar adivinando. Este no es más que un archivo con posibles passwords, generalmente uno por línea. El propio programa se distribuye con algunos de ellos, especialmente con aquellos que contengan palabras relacionadas al área de trabajo de la máquina (en español si se está en México, palabras de informática, de biología, de contabilidad).

Con estos diccionarios y los que el administrador añada, Crack construye una base de datos con la que empieza a trabajar. La primera pasada consiste en probar palabras con todas las letras en minúscula, después mezclando con mayúsculas, después con caracteres alfanuméricos. Habitualmente en las primeras pasadas se obtienen algunos passwords, suficientes para comenzar un ataque.

La forma en que Crack intenta adivinar los passwords es la siguiente: en primer lugar se ordenan y se agrupan las entradas del archivo de passwords con base a su *salt*. Una vez clasificadas, para cada grupo de salts diferentes se selecciona una entrada de diccionario convenientemente tratada, se cifra usando la salt (esto es lo que consume mayor tiempo de CPU) y se compara con el password cifrado de cada miembro del grupo; si coinciden, se ha adivinado un nuevo password.

Para utilizar Crack es necesario contar con el archivo de passwords a atacar; el cual se puede conseguir de la máquina a atacar si no está debidamente protegido. Crack_UDLA está personalizado para obtener periódicamente la lista de passwords vulnerables de los usuarios de la red UDLA.

A continuación se muestra una salida en la ejecución del programa:

```
laboper# ps -ef
   root  5757     1 99   Mar 21 ?       18431:44 cracker -kill run/Klaboper.5683

laboper# ps -ef -o user,pid,pcpu,etime,comm | more
   USER   PID    %CPU    ELAPSED      COMMAND
   root     5757    94.0        13-03:35:35     cracker

laboper# date
Wed Apr  4 00:02:23 CDT 2001
```

Salida:

```
986295333:Guessed edcs [vict0ri4]  Decanatura de Ciencias Sociales      [passwd.acadaplic /bin/csh]
986295388:Guessed vfuente [Tricky.23] Valentin Fuentes                  [passwd.acadaplic /bin/csh]
986297224:Guessed nt200424 [amanda]  PLOTTO LISE                        [passwd.acadapl ic /bin/csh]
```

Si el administrador no suele utilizar este tipo de herramientas por no considerse un atacante, alguien más lo hará. Y lo más importante no es solo correr este programa, sino saber interpretarlo y tomar las medidas pertinentes. En este caso, cambiar inmediatamente el password de la persona para evitar posibles violaciones al mismo.

A través del programa **YPX** es posible obtenetr la tabla de passwords de cualquier máquina que no tenga seguras sus tablas de NIS.

3.1.2 Nessus_UDLA

Una de las herramientas de seguridad más utilizadas durante años en diversos entornos de Unix ha sido SATAN (Security Analysis Tool for Auditing Networks), de Dar Farmer y Wietse Venema . Su tarea consistía en detectar vulnerabilidades de seguridad en Sistemas Unix y redes.

En 1995 SATAN no se había actualizado. Para una herramienta de seguridad esto es demasiado tiempo, por lo que en 1998 surgió Nessus, un analizador de vulnerabilidades gratuito, de código fuente libre y de más fácil uso que su predecesor.

La distribución de Nessus consta de cuatro archivos básicos: librerías del programa, NASL (Nessus Attack Scripting Language), del núcleo de la aplicación y sus plugins. Se requieren de aplicaciones adicionales como GMP (para cifrado) y GTK (para manejo de ambiente gráfico).

Está formado esencialmente por una parte servidora y un cliente gráfico, la cual puede ser invocada y obtener información de forma gráfica o en línea. Este programa fue adaptado para analizar automáticamente sólo a nodos de la familia UDLA (140.148.X.X), bajo un solo administrador

Una salida típica de Nessus_UDLA es la siguiente: (Fig 3.2)

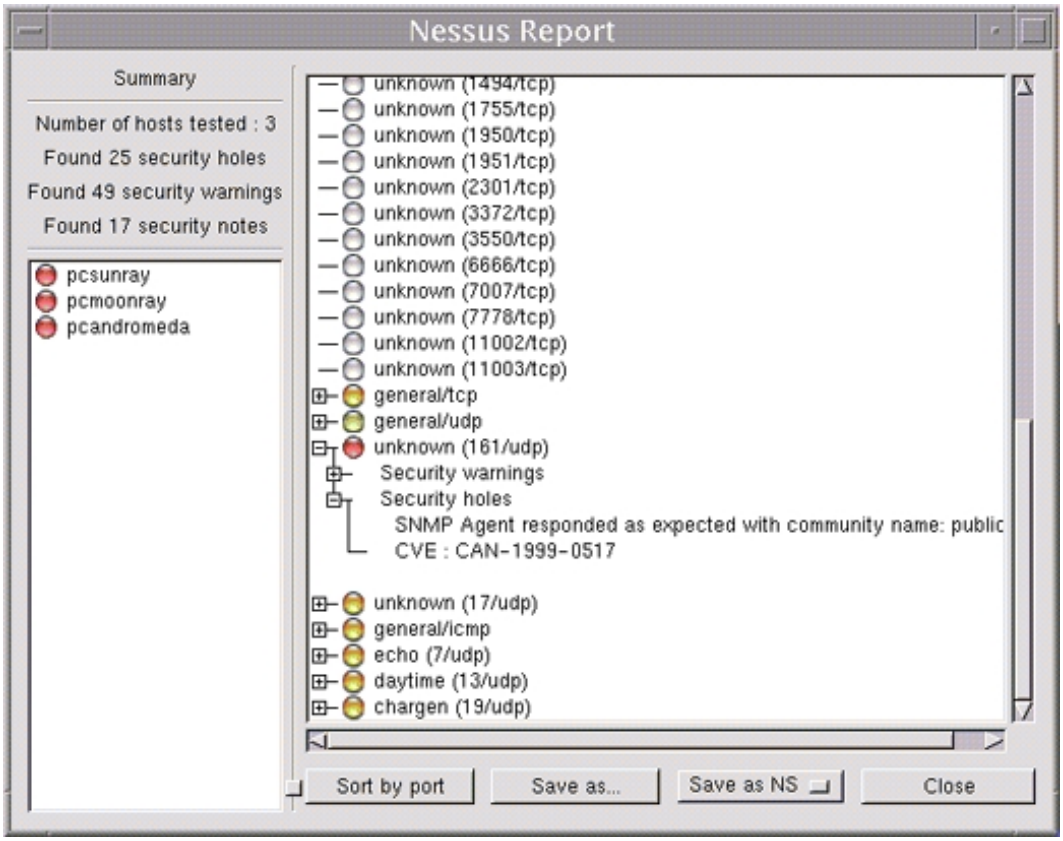

Figura 3.2 Salida de Nessus_UDLA

De la parte izquiera se muestran los nombres de la(s) máquina(s) escaneada(s) y del lado izquierdo todas sus vulnerabilidades, desde simples *warnings* hasta alertas peligrosas. Para mayor detalle sobre su uso ver apéndice A

3.1.3 Ataca.

Ataca (Basado en las ideas de *ISS (Internet Security Scanner)*) es un conjunto de programas en lenguaje C y archivos de datos, los cuales son usados para realizar una

auditoria de seguridad en sistemas UNIX. Los resultados obtenidos se presentan en formato html para fácil interpretación. El programa busca en un determinado rango de direcciones IP de la familia 140.148 fallos de seguridad como lo son: passwords sencillos, particiones NFS con acceso público, y problemas en el daemon sendmail, entre otros.

Está implantado para ejecutarse en web solo con permisos de root. Ver un ejemplo en las Figuras 3.3 y 3.4

3.2 Administrator 's work bench.

Administrator 's work bench es un conjunto de herramientas que permite prevenir o reparar algunas de las vulnerabilidades detectadas en la red UDLA. Está formado por los siguientes módulos:

- Kerberos
- Secure Shell
- Wrappers.
- Monitoreo automático de actividad sospechosa.
- ASIS_ASET

3.2.1 Kerberos.

Kerberos es un sistema de autentificación y distribución de llaves para ser utilizado en redes potencialmente inseguras (como es el caso de Internet).

Debido a su gran éxito, este sistema se extendió a otras redes y otros proyectos, llegando a ser hasta la fecha, la mejor forma de autentificación y protección en la transmisión de datos sobre Internet, basada en llaves simétricas de encriptamiento.

Hasta que se diseño Kerberos, la autenticación en redes se realizaba principalmente de dos formas: o bien se aplicaba la autenticación por declaración (el usuario es libre de indicar el servicio al que desea accesar, esto con un cliente determinado) o se utilizaban passwords para cada servicio de red. Kerberos mejora estos esquemas haciendo que el cliente necesite autorización para comunicarse con el servidor

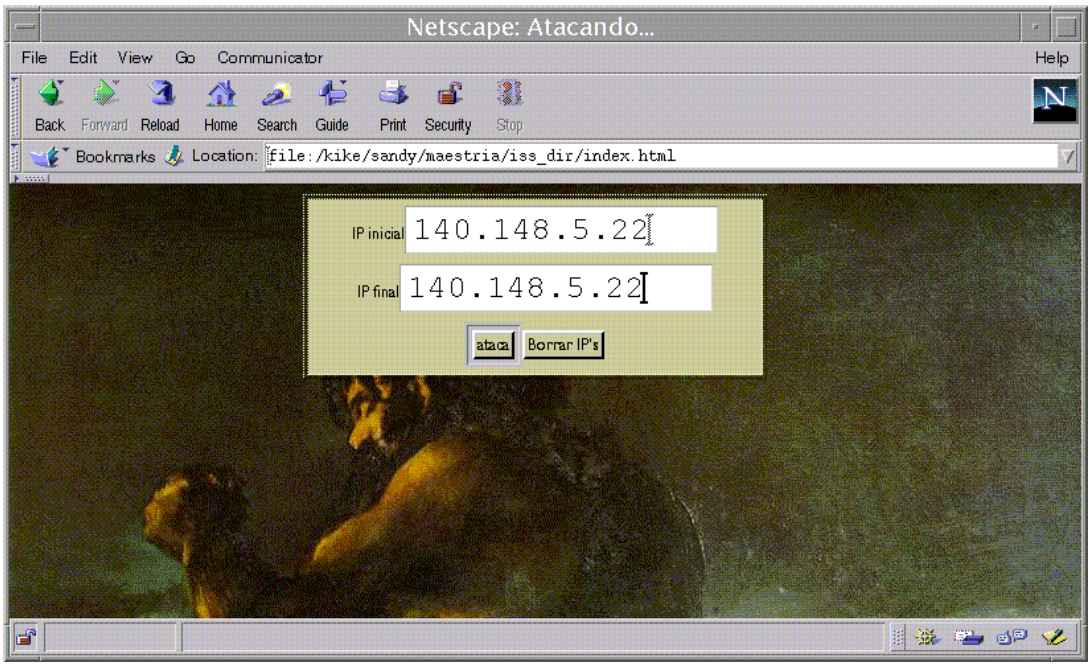

Figura 3.3 Definición de IP's a "atacar"

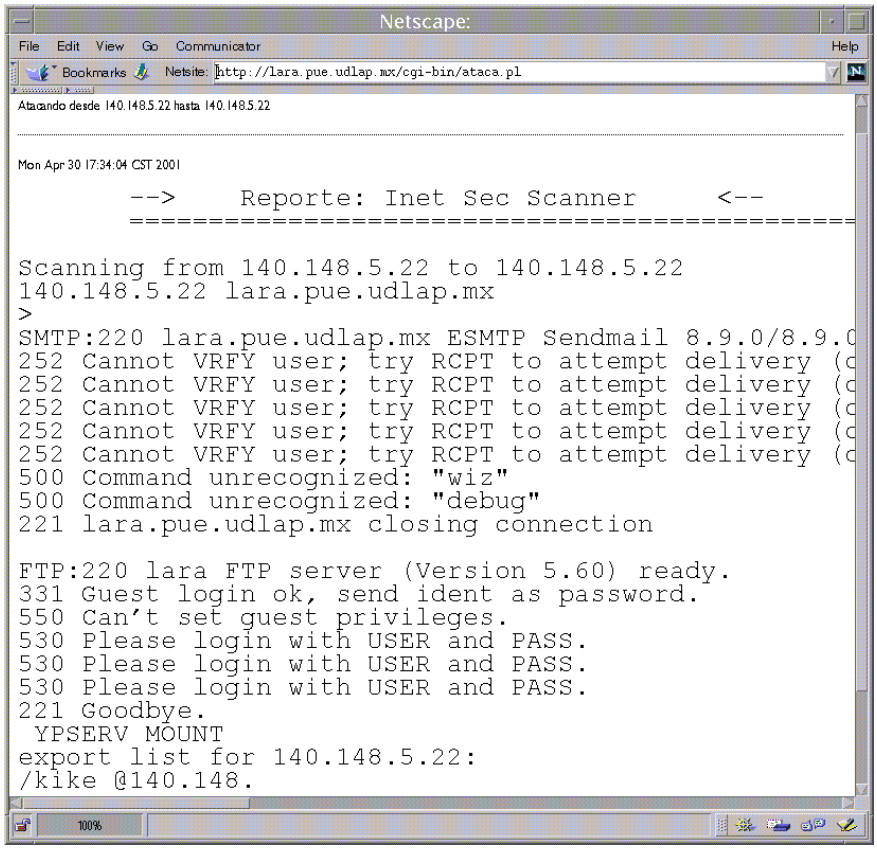

Figura 3.4 Resultados del "ataque"

(y que esa autorización la dé una máquina confiable), y por otro, eliminando la necesidad de mostrar el password del usuario, divulgando dicha información.

3.2.1.1 Autentificación.

Autenticación es la forma en la que se le dice a la máquina quien es el usuario. Esto se hace con un login y un password. Con Kerberos el sistema no cambia para el usuario , y dependiendo de la configuración se puede tener un sistema de autentificación sólo autorizado para usuarios registrados en Kerberos, o bien mixto (dentro y fuera de Kerberos). Internamete, el programa *login* envía el nombre de usuario al servidor de autentificación de Kerberos para solicitar un ticket que le permita comunicarse posteriormente con el servidor de tickets.

3.2.1.2 Tickets.

Una vez autentificado, el cliente de Kerberos pide al servidor de Kerberos un ticket para poder tener acceso a la máquina y subsecuentemente a la red. Si el servidor de Kerberos contiene en su base de datos el nombre del usuario toma un ticket nuevo, lo encripta con la llave del usuario que se encuentra en la base de datos y lo envía encriptado al cliente. Este ticket es desencriptado localmente en la máquina cliente, con el password que introduce el usuario, logrando de esta forma que el password no viaje por la red.

Una vez con el ticket el usuario puede utilizar servicios (telnet, ftp, rlogin, rsh, etc.) seguros, y habrá autentificado en la máquina local eficazmente.

Para ver mayores detallas de su instalación y uso ver el apéndice B.

Aspectos interesantes de los servicios

Primero que nada hay que aclarar que con Kerberos se pueden tener los servicios de encriptamiento habilitados o bien deshabilitados, así como los servicios de autentificación de igual forma habilitados o no. Obteniendo de esta manera una flexibilidad que permite migrar poco a poco todos los servicios. Cuando todos los servicios sean migrados y los usuarios dados de alta en la base de datos de Kerberos, podrán deshabilitarse algunos servicios y podrá darse la opción en cada servidor de que sólo acepten servicios encriptados y usuarios autentificados por Kerberos, con sus respectivos tickets de acceso.

Todos los servicios del cliente se pueden encontrar en /usr/local/bin, y los guardianes del servidor en **/usr/local/sbin.**

3.2.1.3 Telnet con Kerberos.

Cuando se requiere la versión encriptada de telnet por medio de la opción -x, y el usuario cuenta con el ticket que le asigna el servidor de Kerberos, el servicio es transparente, (no requiere username ni password) , el servicio de terminal remota entra automáticamente como si fuera un "rlogin" autorizado:

Por ejemplo:

```
lara :sandra > telnet -x acadaplic
Trying 140.148.155.177...
Connected to acadaplic (140.148.155.177).
Escape character is '^]'.
Waiting for encryption to be negotiated...[ Kerberos V5 accepts you as ``sandra@UDLAP.MX" ]
done.
Last login: Thu Apr  5 12:37:22 from dns-sec.pue.udla
Sun Microsystems Inc.   SunOS 5.6       Generic August 1997

acadaplic:sandra >
```

En ningún momento pide el password, de esta forma no viaja sobre la red.

Enseguida se presenta lo que pasa a nivel red y como viajan los datos cuando se hace una conexión con y sin Kerberos:

Primero sin Kerberos:

```
Using device /dev/le (promiscuous mode)
lara.pue.udlap.mx -> bell TELNET C port=33312
lara.pue.udlap.mx -> bell TELNET C port=33312 s
bell -> lara.pue.udlap.mx TELNET R port=33312 s
lara.pue.udlap.mx -> bell TELNET C port=33312
lara.pue.udlap.mx -> bell TELNET C port=33312 a
bell -> lara.pue.udlap.mx TELNET R port=33312 a
lara.pue.udlap.mx -> bell TELNET C port=33312
lara.pue.udlap.mx -> bell TELNET C port=33312 n
bell -> lara.pue.udlap.mx TELNET R port=33312 n
lara.pue.udlap.mx -> bell TELNET C port=33312
lara.pue.udlap.mx -> bell TELNET C port=33312 d
bell -> lara.pue.udlap.mx TELNET R port=33312 d
lara.pue.udlap.mx -> bell TELNET C port=33312
lara.pue.udlap.mx -> bell TELNET C port=33312 r
bell -> lara.pue.udlap.mx TELNET R port=33312 r
```

```
lara.pue.udlap.mx -> bell TELNET C port=33312
lara.pue.udlap.mx -> bell TELNET C port=33312 a
bell -> lara.pue.udlap.mx TELNET R port=33312 a
lara.pue.udlap.mx -> bell TELNET C port=33312
lara.pue.udlap.mx -> bell TELNET C port=33312
bell -> lara.pue.udlap.mx TELNET R port=33312
lara.pue.udlap.mx -> bell TELNET C port=33312
bell -> lara.pue.udlap.mx TELNET R port=33312 Password for sandra:
lara.pue.udlap.mx -> bell TELNET C port=33312
lara.pue.udlap.mx -> bell TELNET C port=33312 J
bell -> lara.pue.udlap.mx TELNET R port=33312
lara.pue.udlap.mx -> bell TELNET C port=33312 *
bell -> lara.pue.udlap.mx TELNET R port=33312
lara.pue.udlap.mx -> bell TELNET C port=33312 %
bell -> lara.pue.udlap.mx TELNET R port=33312
lara.pue.udlap.mx -> bell TELNET C port=33312 R
bell -> lara.pue.udlap.mx TELNET R port=33312
lara.pue.udlap.mx -> bell TELNET C port=33312 F
bell -> lara.pue.udlap.mx TELNET R port=33312
lara.pue.udlap.mx -> bell TELNET C port=33312 G
bell -> lara.pue.udlap.mx TELNET R port=33312
lara.pue.udlap.mx -> bell TELNET C port=33312 w
bell -> lara.pue.udlap.mx TELNET R port=33312
lara.pue.udlap.mx -> bell TELNET C port=33312 N
bell -> lara.pue.udlap.mx TELNET R port=33312
lara.pue.udlap.mx -> bell TELNET C port=33312
bell -> lara.pue.udlap.mx TELNET R port=33312
lara.pue.udlap.mx -> bell TELNET C port=33312
bell -> lara.pue.udlap.mx TELNET R port=33312 login:
lara.pue.udlap.mx -> bell TELNET C port=33312
bell -> lara.pue.udlap.mx TELNET R port=33312 Sun Microsystems
```

Aquí toda la comunicación viaja en modo texto y puede ser fácilmente capturada por alguna estación con tarjeta de red en modo promiscuo.

Después con Kerberos:

```
Using device /dev/hme (promiscuous mode)
lara.pue.udlap.mx -> bell        TELNET C port=33285
      bell -> lara.pue.udlap.mx TELNET R port=33285
lara.pue.udlap.mx -> bell        TELNET C port=33285
lara.pue.udlap.mx -> bell        TELNET C port=33285
      bell -> lara.pue.udlap.mx TELNET R port=33285
      bell -> lara.pue.udlap.mx TELNET R port=33285
lara.pue.udlap.mx -> bell        TELNET C port=33285
      bell -> lara.pue.udlap.mx TELNET R port=33285
lara.pue.udlap.mx -> bell        TELNET C port=33285
```

```
lara.pue.udlap.mx -> bell        TELNET C port=33285
      bell -> lara.pue.udlap.mx TELNET R port=33285
lara.pue.udlap.mx -> bell        TELNET C port=33285
      bell -> lara.pue.udlap.mx TELNET R port=33285
lara.pue.udlap.mx -> bell        TELNET C port=33285
      bell -> lara.pue.udlap.mx TELNET R port=33285
lara.pue.udlap.mx -> bell        TELNET C port=33285
      bell -> lara.pue.udlap.mx TELNET R port=33285
lara.pue.udlap.mx -> bell        TELNET C port=33285
      bell -> lara.pue.udlap.mx TELNET R port=33285
lara.pue.udlap.mx -> bell                              TELNET    C    port=33285
w\371\212o\212|\274%\313=1\227\227*N\232[>\341L
      bell -> lara.pue.udlap.mx TELNET R port=33285 \275CM
lara.pue.udlap.mx -> bell        TELNET C port=33285 G\24v
      bell -> lara.pue.udlap.mx TELNET R port=33285 "\356=
lara.pue.udlap.mx -> bell        TELNET C port=33285
      bell -> lara.pue.udlap.mx TELNET R port=33285
lara.pue.udlap.mx -> bell        TELNET C port=33285
      bell -> lara.pue.udlap.mx TELNET R port=33285
lara.pue.udlap.mx -> bell        TELNET C port=33285
```

Aquí no se entiende y no se ve absolutamente nada de la sesión.

3.2.1.4 FTP.

Kerberos permite una comunicación segura para ftp, negando el acceso a usuarios no autorizados (si el servicio está configurado de forma exclusiva en inetd.conf) y encriptando toda la comunicación. Este ftp kerberizado utiliza una aplicación para la autentificación que se llama GSSAPI [GAR91]. Una vez que el usuario teclea su username la comunicación se establece sin necesidad de teclear el password, en caso de que el usuario introduzca un username diferente del suyo, el servidor pide un password para el username apropiado, pero es importante notar que dicho password y el username , sea el que sea, viaja encriptado.

3.2.1.5. Instalador Automático.

Antes de esta tesis, kerberos se tenía sólo en los servidores centrales, pues el tiempo que se necesitaría para su instalación sería considerable, además no se cuenta con personal que tenga los conocimientos para hacerlo y el resto de las actividades los absorben. Para ello se realizó un sistema automático de instalación, con ello solo se ejecutan unos scritps y en menos de 5 minutos es posible hacer uso de los servicios kerberizados. El archivo instalador se tiene en un nodo central y se envía a aquellas máquinas con permisos de root que lo necesiten. Este script se ha probado para las

versiones 2.6, 2.7 y 2.8 de Solaris y ha funcionado sin ningún problema. Para ver detalles de instalación, ver Apéndice B.

3.2.2 Secure Shell.

Cuando se realiza una conexión a un servidor remoto usando por ejemplo el comando telnet o ftp, el login(usuario) y password(contraseña) son transmitidos en la red de forma clara, lo cual representa un gran riesgo si llega a existir sobre la red un programa que capture la información, basándose en el modo promiscuo de las redes ethernet (comúnmente llamado sniffer), pues se puede obtener tanto el login como el password y posteriormente irrumpir en el servidor con esta información.

Este tipo de problemáticas ha llevado al diseño de herramientas que permitan evitar estas situaciones siendo el caso de Secure Shell.

Secure Shell (ssh) es un programa que permite realizar conexiones entre máquinas a través de una red abierta, así como ejecutar programas en una máquina remota y copiar archivos de una máquina a otra. Proporciona una exhaustiva autenticación y comunicaciones seguras en redes no seguras.

Ssh provee fuerte autenticación y comunicación segura sobre un canal inseguro y nace como un reemplazo a los comandos telnet, ftp, rlogin, rsh, y rcp, los cuales proporcionan gran flexibilidad en la administración de una red, sin embargo, presenta grandes riesgos en la seguridad de un sistema. Adicionalmente, ssh provee seguridad para conexiones de servicios X Windows y envío seguro de conexiones arbitrarias TCP.

La ventaja más significativa de ssh es que no modifica mucho las rutinas. En todos los aspectos, iniciar una sesión de ssh es tan sencillo como (y similar a) iniciar una sesión de telnet. Tanto el intercambio de llaves, la autenticación, así como el posterior cifrado de sesiones son transparentes para los usuarios.

Antes de este trabajo NO se había probado este métedo de seguridad en la UDLA. En esta tesis no solo se adapta para las necesidades de la Universidad, sino que se instala automáticamente como parte de la instalación remota de máquinas Solaris 2.8

Para ver más detalles de configuración e instalación, ver Apéndice C.

Sesión "telnet" entre máquinas en UNIX

Ejemplos:

$ ssh -l micuenta *maquina.remota*
En este ejemplo utilizamos la opción -l para proporcionar el login con el que tendremos acceso a la máquina remota.
En este caso la cuenta es micuenta y la máquina es maquina.remota.

$ ssh *micuenta@maquina.remota*
También podemos hacer uso del formato descrito arriba para entrar a una cuenta dentro de una máquina remota.

$ ssh *maquina.remota*
En caso de poseer el mismo nombre de cuenta en ambas máquinas (local y remota) es posible tener acceso a la máquina remota proporcionando solamente el nombre de la máquina.

Transferencia de archivos

Ejemplos:

$ scp micuenta@*máquina.remota*:/tmpu/archivo /copias
En este caso se copiará /tmpu/archivo localizado en la máquina remota maquina.remota al directorio /copias en la máquina local. Se utilizó la cuenta micuenta para acceder al servidor.

$ scp /copias/archivo *micuenta@máquina.remota*:~/bck

En este caso se realizará la copia del archivo /copias/archivo localizado en la máquina local a la máquina remota máquina.remota colocando el archivo en el directorio bck del home de micuenta.

A continuación se presenta un ejemplo de la comunicación de ssh. La máquina *lara* pide acceso a la máquina *acadaplic*. El password viaja a través de la red en forma encriptada y no es posible obtenerlo de la red:

```
lara# ssh acadaplic
root's password:
Authentication successful.
Last login: Thu May 10 2001 02:26:50
Sun Microsystems Inc.   SunOS 5.6       Generic August 1997
```

Al espiar el tráfico de la red entre las máquinas *lara* y *acadaplic* solo es posible obtener lo siguiente:

```
#lara> snoop acadaplic
acadaplic -> lara.pue.udlap.mx TCP D=46634 S=22 Syn Ack=3433417090 Seq=336326
8323 Len=0 Win=64240
lara.pue.udlap.mx -> acadaplic    TCP D=22 S=46634     Ack=3363268324 Seq=343341
7090 Len=0 Win=8760
  acadaplic -> lara.pue.udlap.mx TCP D=46634 S=22     Ack=3433417090 Seq=336326
8324 Len=49 Win=64240
lara.pue.udlap.mx -> acadaplic    TCP D=22 S=46634     Ack=3363268373 Seq=343341
7090 Len=0 Win=8760
lara.pue.udlap.mx -> acadaplic    TCP D=22 S=46634     Ack=3363268373 Seq=343341
7090 Len=49 Win=8760
  acadaplic -> lara.pue.udlap.mx TCP D=46634 S=22     Ack=3433417139 Seq=336326
8373 Len=0 Win=64240
  acadaplic -> lara.pue.udlap.mx TCP D=46634 S=22     Ack=3433417139 Seq=336326
8373 Len=712 Win=64240
lara.pue.udlap.mx -> acadaplic    TCP D=22 S=46634     Ack=3363269085 Seq=343341
7139 Len=0 Win=8760
lara.pue.udlap.mx -> acadaplic    TCP D=22 S=46634     Ack=3363269085 Seq=343341
7139 Len=280 Win=8760
  acadaplic -> lara.pue.udlap.mx TCP D=46634 S=22     Ack=3433417419 Seq=336326
9085 Len=0 Win=64240
lara.pue.udlap.mx -> acadaplic    TCP D=22 S=46634     Ack=3363269085 Seq=343341
7419 Len=144 Win=8760
  acadaplic -> lara.pue.udlap.mx TCP D=46634 S=22     Ack=3433417563 Seq=336326
9085 Len=0 Win=64240
  acadaplic -> lara.pue.udlap.mx TCP D=46634 S=22     Ack=3433417563 Seq=336326
9085 Len=640 Win=64240
lara.pue.udlap.mx -> acadaplic    TCP D=22 S=46634     Ack=3363269725 Seq=343341
7563 Len=0 Win=8760
  acadaplic -> lara.pue.udlap.mx TCP D=46634 S=22     Ack=3433417563 Seq=336326
9725 Len=16 Win=64240
lara.pue.udlap.mx -> acadaplic    TCP D=22 S=46634     Ack=3363269741 Seq=343341
7563 Len=0 Win=8760
```

De la salida anterior no es posible obtener información en texto plano. Así el password tecleado viajo de forma segura.

Se propone estandarizar el uso de ssh en la UDLA, y hasta cierto punto erradicar el esquema de Kerberos. Esto debido a lo transparente que resulta el uso de ssh y también porque las últimas versiones de Kerberos no se pueden exportar de Estados Unidos a México y la UDLA estaría expuesta a ataques externos fácilmente.

3.2.3 Wrappers

Hay una serie de servicios como telnet o ftp que habitualmente no se pueden cerrar ya que los usuarios necesitan conectarse al servidor para trabajar en él o para transferir archivos; en estos casos es peligroso permitir que cualquier máquina en Internet tenga la posibilidad de acceder a los recursos.

TCP Wrappers son herramientas para definir redes o máquinas autorizadas a conectarse a una determinada máquina. Para este trabajo, se configuró a las necesidades de la UDLA la versión 7.6.

Tras compilar el software se habrán generado una serie de archivos ejecutables. Posteriormente se edita el archivo /etc/inetd.conf para indicarle al proceso inetd que ha de utilizar el denominado tcpd (la parte más importante de TCP Wrapper) al momento de servir peticiones, es decir, una entrada en el archivo inetd.conf de la forma:

```
telnet  stream  tcp     nowait  root    /usr/sbin/in.telnetd    in.telnetd
```

se convertirá en una como:

```
telnet  stream  tcp     nowait  root    /usr/local/sbin/tcpd /usr/local/sbin/telnetd -a none
```

Como se ve, en lugar de que inetd ejecute directamente el daemon correspondiente a cada servicio, ejecutará el wrapper, y es éste el encargado de controlar la ejecución del daemon real.

Hasta aquí se han configurado los servicios que se ofrecen a diferentes máquinas, aún así se sugiere la postura que todo lo que no esté explicitamente permitido, esté negado. Para ello, en el archivo /etc/hosts.allow se indica que servicios se ofrecen y a quienes, esto de la siguiente forma:

daemon:máquinas

Donde *daemon* es el nombre del daemon encargado de atender el servicio correspondiente y máquinas es la especificación de los hosts (IP o subdominios) permitidos para conectarse a cada servicio. Por ejemplo:

```
maquina# more /etc/hosts.allow
ALL : 140.148.5.35,140.148.7.177
telnetd : 140.148.
ftpd : 140.148.
```

ftpd : 200.34.
telnetd : 200.34.:bin/sh /etc/ADM/esl/cambia_shell ${ARGV}

Deliberadamente se puede especificar los hosts que se deniegan para cualquier servicio en el archivo /etc/hosts.deny
Ejemplo:

acadaplic:> more /etc/hosts.deny
ALL : ALL
acadaplic:>

Aquí se indica que por default TODAS las máquinas están denegadas.
Cuando alguien desde una máquina que está autorizada solicita un servicio no notará ningún cambio, pero para aquellas denegadas la conexión se cerrará, mostrando el mensaje:

Connection closed by foreign host.

Una vez configurado todo, es necesario reiniciar al inetd para que tome su archivo con modificaciones (esto con un kill -HUP número_de_proceso_inetd)

Para ver detalles de la instalación, ver Apéndice D

3.2.4 Monitoreo automático de actividad sospechosa.

Deteccion de IP's no validos.

Dada la cantidad de actividades que se realizan en el centro de cómputo, es muy práctico que de forma automática se reporten aquellas actividades consideradas como "sospechosas". Una de ellas es que ningún nodo externo (es decir que no pertenezca a la familia de IP's 140.148.) debe estar conectado directamente en alguno de los nodos de la red UDLA.

Para tales fines, se creó un proceso que detecta si están o intentan conectarse máquinas externas.

Por ejemplo:
De la siguiente lista:

Mar 14 13:05:13 lara telnetd[11263]: connect from ssray3.pue.udlap.mx
Mar 14 13:14:48 lara telnetd[11412]: connect from superacadaplic
Mar 14 13:41:52 lara telnetd[11507]: connect from agora.agora.com.mx
Mar 14 15:49:13 lara telnetd[11708]: connect from bell

reportará que la máquina agora.agora.com.mx no pertenece a la UDLA, y envía una lista de los usuarios que estan conectados en esa máquina:

Date: Mon, 26 Feb 2001 12:23:06 -0600 (CST)
From: inspector_intrusos@mail.udlap.mx
Subject: posibles_intrusos

----- Finger a primer nivel----
Feb26_12:17:42
[200.34.71.96]

Login	Name	TTY	Idle	When	Where
setup	Netra Administrator	pts/2	5	Thu 11:33	lara.pue.udlap.mx
rubli	Alex Rubli	pts/0	2	Mon 12:23	rubli.pue.udlap.mx
hacker	Soy un hacker	pts/1	1	Sun 2:23	server.yahoo.com

----Fin finger a primer nivel--

A partir de estos resultados, se analizan nuevamente las máquinas desde las cuales los usuarios estan conectadas y se obtiene una lista de los usuarios presentes. Para este caso se obtiene informacion sobre server.yahoo.com

---
----- Finger a primer nivel----
Feb26_12:17:43
[201.33.11.26]

Login	Name	TTY	Idle	When	Where
hacker	Soy un hacker	pts/1	1	Sun 2:23	server.yahoo.com

----Fin finger a primer nivel--

Estos datos se envían al administrador, pues muestra evidencia de un posible intruso. Este envío se hace a través de un sistema de correo propio de la máquina que se está monitoreando, es decir, no necesita enviar estos datos confidenciales a alguna otra máquina para ser enviados.

Detección de intentos de conexiones foráneas

Resultaría muy sospechoso encontrar una lista de máquinas externas a la UDLA que inciden en en intentar conectarse a un nodo de la red de la universidad. Para ello se desarrollo un sistema automático de detección de máquinas externas (es decir no pertenecen a la familia 140.148.) con intentos fallidos de conexión a nodos UDLA. Este reporte se envía a través del sistema de correo propio mencionado a lo largo de este capítulo.

Deteccion de nuevos *root*

Otra tarea importante es detectar a aquellos usuarios normales (sin permisos de root) de alguna manera consiguen convertirse en root. Para ello se desarrollo un sistema que verifica todos los últimos éxitos de usuarios no privilegiados que consiguieron convertirse en superusuario y se envía el reporte al administrador. Este chequeo incluye la fecha y hora en que el usuario obtuvo acceso. Esto se hace automáticamente con un sistema de correo autónomo y se propone se obtenga el reporte cada hora. Teniendo estos datos, es posible comenzar el rastreo de un posible hacker.

El tipo del reporte es el siguiente:

Date: Thu, 19 Apr 2001 02:24:39 -0600 (CST)
From: inspector_intrusos@mail.udlap.mx
Subject: nuevos_root

Estos son los ultimos usuarios que se han
convertido en root en esta maquina
Tomar precauciones...

+ tesiss-root	02/21_16:46
+ tesiss-root	03/08_21:08
+ ar105674-root	04/08_21:26
+ tesiss-root	04/11_19:49
+ tesiss-root	04/14_18:29
+ telefono-root	10/25_16:01
+ phone-root	12/21_10:48
+ phone-root	01/07_08:59
+ nt204732-root	01/11_17:12
+ pepe-root	01/21_11:58

Deteccion de usuario convertidos en algun miembro privilegiado

Es importante monitorear aquellos usuarios no privilegiados que por algún motivo lograron entrar a la cuenta de alguno de los miembros del centro de cómputo (miembros privilegiados) así como la fecha y la hora de dicho evento. Para ello se cuenta con un sistema que reporta (se propone que cada hora) de manera automática esas entradas de usuarios sospechosos bajo el siguiente formato :

Date: Thu, 19 Apr 2001 17:09:59 -0600 (CST)
From: inspector_intrusos@mail.udlap.mx
Subject: nuevos_cass

Estos son los ultimos usuarios que se han
convertido en algun miembro de cass en esta maquina
Tomar precauciones...

+ is092333-enrique 02/10_22:53
+ is092333-enrique 02/10_22:53
+ ar082333-rcastro 02/10_22:58
+ is092333-aldrette 2/10_22:58
+ is092333-rcastro 02/10_23:12
+ is092333-angel 02/10_23:25
+ tesiss-enrique 02/23_23:16
+ phone-angel 01/07_15:17
+ pepe-angel 01/07_19:32
+ esanchez-sandra 04/19_15:20

Con esta evidencia se tiene una base sólida para hacer un seguimiento y encontrar un posible intruso.

Detección de procesos con *setuid* ejecutados por cualquier usuario que no sea root

Durante una sesión de trabajo de un usuario, hay algunos procesos que son ejecutados como root directamente por el sistema operativo. Sin embargo, un hacker puede instalar procesos que se ejecuten con permisos de setuid (Capítulo 2.1.2.1) en una cuenta normal, y eso puede ser una pista clara de que alguien con permisos ilegales está utilizando una máquina. Para ello se generó un sistema de alerta sobre los procesos que están con setuid y que no son legales. El resultado es el siguiente:

Date: Thu, 26 Apr 2001 20:18:29 -0600 (CST)
From: inspector_procesos_SP@mail.udlap.mx
Subject: procesos_SP

---
sh S uucp __ 0.02 secs Thu Apr 26 20:11
sh S guess __ 0.02 secs Thu Apr 26 20:00

```
uudemon.        F       uucp     __          0.00 secs Thu Apr 26 20:00
ping            S       sandra   pts/7       0.01 secs Thu Apr 26 19:56
sendmail        SF      sandra   __          0.00 secs Thu Apr 26 19:42
mail            S       hacker   __          0.01 secs Thu Apr 26 19:42
sendmail        S       sandra   pts/6       0.11 secs Thu Apr 26 19:42
sendmail        F       sluis    pts/6       0.01 secs Thu Apr 26 19:42
sh              F       hacker   pts/6       0.00 secs Thu Apr 26 19:42
```

La mayoría de los programas que se ejecutan automáticamente están definidos en el crontab de la máquina para su ejecución periódica. A continuación se muestra un ejemplo:

```
lara# crontab -l
#ident  "@(#)root       1.14    97/03/31 SMI"   /* SVr4.0 1.1.3.1 */
#
# The root crontab should be used to perform accounting data collection.
#
# The rtc command is run to adjust the real time clock if and when
# daylight savings time changes.
20 21 * * * /etc/ADM/sandra/bin/new_root/new_root > /dev/null 2>&1
20 21 * * * /etc/ADM/sandra/bin/new_root/new_solaris /dev/null 2>&1
1 12,18 * * * /etc/ADM/sandra/bin/procesos_SF/new_solaris /dev/null 2>&1
```

3.2.4.1 Sistema de correo autónomo.

Para este trabajo fue necesario crear un sistema de correo autónomo para no depender bajo ninguna circunstancia del sistema propio de las máquinas.

Se creó basado en los siguientes puntos:

- Cada máquina cliente no debe contar con el sistema de correo que el sistema operativos tiene por default. Este se quita al momento de su primera instalación.
- Si un hacker lograra levantar el sistema y alterarlo a su conveniencia, jamás interceptará los reportes generados por este trabajo de tesis.
- Se tiene absoluto control sobre el encabezado del mensaje : From, To, Subject y cuerpo del mensaje.

3.2.5 ASIS_ASET

Automated Security Enhancement Tool (ASET) es un conjunto de utilerías que se incluyen en la distribución del sistema operativo Solaris 2.x y que permiten al administrador:

3 Detectar alteraciones en el sistema de archivos del sistema operativo o en algún otro archivo definido.
4 Verificar atibutos y contenido de los archivos.
5 Analizar el sistema de archivos bajo tres niveles de monitoreo: bajo, medio y alto.

Basado en esta herramienta se creó ASIS_ASET, un sistema de monitoreo para detectar cambios en los archivos del sistema operativo, los cuales son las principales víctimas de un hacker.

Ofrece las siguientes ventajas:

4 **Radiografía inicial:** Cuando una máquina se instala por primera vez, se crea una "radiografía" de los archivos que se consideran importantes y que nunca deben cambiar.
5 **Envío automático de reporte con sistema propio:** Al detectar archivos corruptos, se envia de manera automática a través del sistema de correo autónomo el reporte de la actividad sospechosa en dicha máquina.
6 **Cambios controlados:** Ofrece un ignore list para aceptar cambios hechos por el administrador en esos archivos críticos y que no se reporten como una alteración.
7 **No se degrada el servicio:** ASIS_ASET incluye sólo un nivel de evaluación : high. Este servicio se levanta cada hora (obligados por el consumo de energía y ganar meno exposición por tener la máquina apagada). Su consumo de CPU es mínimo: (Se presenta un ejemplo de uso de cpu mediante la instrucción *iostat* durante la ejecución de aset)

```
lara# date
Wed May  2 17:24:41 CST 2001

lara# ps -ef | grep aset
   root 27592     1  0 17:25:22 pts/6    0:00 /bin/sh aset
   root 29031 29030  0 17:25:40 pts/6    0:00 /bin/sh /usr/aset/util/addcksum
   root 28668 28667  0 17:25:36 pts/6    0:00 /bin/sh /usr/aset/tasks/cklist
   root 28669 28668  0 17:25:36 pts/6    0:00 /bin/sh /usr/aset/tasks/create_cklist
/usr/aset/masters/cklist.low
   root 28667 27592  0 17:25:36 pts/6    0:00 /bin/sh aset

lara# iostat -c 5 6
        cpu
 us sy wt id
  1  1  0     98
 24 39 20 17
 38 62  0  0
```

```
 37 63  0  0
  1  2  0 97
  0  0  0 99

lara# date
Wed May  2 17:26:41 CST 2001
```

- **Adios Tripwire:** Tripwire se tenía instalado y que se ejecutaba una vez al día por el alto consumo de recursos y de tiempo (Aprox. 2 horas y ocupaba aprox. 90% de cpu). Además de que las salidas que se reportaban nadie las analizaba, pues la persona que lo hacía ya no trabaja aquí y no generó ningun sistema automático que lo hiciera. Por otro lado, Tripwire se tenía que obtener de la red, instalar y configurar cliente por cliente.

- **Incluido en el server instalador de máquinas:** Se preparó un script de autoinstalación de ASIS_ASET para incluirlo en el servidor de instalaciones de máquinas. Así todo el proceso de instalación y preparación para su uso es transparente para todos.

3.3 Servicios Nuevos en la UDLAP con Firewall

Aunque la Universidad de las Américas ya tiene un historial de 4 años consecutivos sin problemas graves provocados por intrusos maliciosos de Internet ni locales, contar ya con los servicios de un Firewall es una medida prudente que arrojará varios beneficios que se describirán enseguida.

3.3.1 Servicios al Exterior

Todos los servicios al exterior estaban cerrados hasta antes de este trabajo de tesis, que fue el primer eslabón de este trabajo de tesis, excepto el correo electrónico y un uso restringido de *ftp* y *telnet* a través del nodo **miudla.**

Con el firewall, es posible ofrecer servicios más generales de ftp, telnet. También se hace más seguro el uso del correo electrónico y se pueden planear mecanismos más seguros para otros servicios que se van demandando en Internet.

Telnet en Internet

Las opciones de telnet en Internet se pueden clasificar en dos tipos: **encriptados** y **no encriptados**. Debido a la complejidad para instalar y configurar un telnet encriptado (por ejemplo, con soporte a Kerberos) obliga a que en la realidad los usuarios finales no

lo usen o no lo puedan usar, de tal manera que para el gran volumen de usuarios de la UDLA esta solución debe irse permeando poco a poco. Por otro lado, para un telnet no encriptado se hace necesario que se cumplan cuatro requisitos:

- Que el password del usuario sea seguro.
- Que la autentificación del usuario se guarde en bitácora.
- Que el nodo destino del usuario sea el apropiado o dirigido.
- Que las direcciones origen y destino se guarden en bitácora.

Los cuatro requisitos anteriores le permiten al administrador hacerle difícil la entrada a un intruso y, en el caso de que se presente un ataque, rastrearlo y delimitar responsabilidades.

Ftp en Internet

Las opciones de Ftp en Internet se clasifican de igual manera que el telnet. Un punto débil que se presenta tanto en Ftp como en Telnet no encriptados es que un intruso interno a la UDLA puede estar espiando las conversaciones provenientes del firewall siendo esto de gran provecho para él ya que todos los usuarios de Internet entran por ahí. Para evitar esto, se añadirá un quinto requisito a los servicios no encriptados en general:

5. Que el servicio sea un proxy ubicado en un nodo único a un segmento y dicho nodo sea un firewall.

Este quinto requisito evita que un espía observe las conversaciones porque los programas que realizan tal tarea se valen de la característica de que las redes de tipo *broadcast* ponen los datos disponibles a todos los nodos miembros de un segmento; al hacer que el firewall sea el único miembro de un segmento excluye a los espías del orígen de las conversaciones (Firewall) y los delimita a espiar cada destino por separado.

WWW en Internet

Este servicio ya se ofrecía antes del presente trabajo, sin embargo, ahora se añadirán nuevas características de seguridad, contenido y desempeño.

Seguridad en WWW

Los nodos de la UDLA que ofrecen servicios de WWW pueden ser atacados de varias formas:

7.1.1.1 Explotando hoyos de seguridad que permiten ejecutar programas del sistema.
7.1.1.2 Accesando el contenido de páginas confidenciales.
7.1.1.3 Realizando peticiones programadas masivas que degradan el desempeño y debilitan la seguridad de los sistemas.
7.1.1.4 Explotando hoyos de seguridad que permiten corromper el servicio.

No se hace mención exhaustiva de todos los posibles ataques a un servidor de WWW aquí, sino que se ejemplifican los necesarios para demostrar las nuevas funcionalidades derivadas del presente trabajo.

Para evitar los problemas del punto (1) los programas servidores de WWW se corren con permisos no privilegiados, anulando así el acceso a programas del sistema.

Para evitar problemas del tipo (2) ahora el acceso a los servidores de WWW se harán por medio del Firewall. De esta manera, el Firewall sólo permite que exista tráfico entre Internet y los nodos de servicios WWW que no sean confidenciales.

Para evitar problemas del tipo (3) a los nodos con servicios WWW, como es el Firewall quien inicialmente atiende estas peticiones, se especifica una cantidad máxima de peticiones de un sólo orígen y al ser rebasadas se rechazan el resto. De esta manera, los nodos finales que sirven WWW se ven librados de una gran cantidad de peticiones nocivas mejorando su desempeño general y, más notablemente, el local.

Para minimizar el efecto de problemas del tipo (4), como es el Firewall quien recibirá las peticiones o ataques, en el peor de los casos se puede observar un deterioro en el desempeño del Firewall mismo como resultado de dichos ataques, librando a los nodos reales y manteniendo así la integridad de su información.

Contenido de WWW

Las peticiones de acceso a direcciones WWW (URL's) tanto de entrada como de salida serán a través del Firewall. Éste cuenta con mecanismos para bloquearlas o redirigirlas, lo cual es muy benéfico para la UDLA cuyo quehacer es el académico. Al bloquear las direcciones indeseadas para los propósitos académicos (pornografía, por ejemplo), se libera de esa carga a los enlaces que la UDLA tiene al exterior, pudiéndose aprovechar en objetivos mejores. De igual manera, se pueden dejar ver o publicar únicamente aquellos nodos internos pertinentes al exterior (Internet).

Desempeño de servicios WWW

El Firewall permite que ciertas direcciones puedan ser redirigirlas a servidores alternos logrando de esta manera un balanceo de cargas y, por lo tanto, un mejor desempeño. Al evitar que los nodos servidores internos no reciban peticiones nocivas y que los enlaces al exterior se liberen de cargas nocivas tambien mejora el desempeño.

Correo Electrónico en Internet

La historia del correo electrónico en Internet se ha visto poblada de experiencias dolorosas por los hoyos de seguridad que los servidores exponen en todos los sistemas operativos.

Las debilidades del servicio de correo se pueden numerar así:

5.1.1.1 Ejecución de programas del sistema a traves del correo.
5.1.1.2 Ejecución de opciones que permiten averiguar la existencia de cuentas de usuarios.
5.1.1.3 Peticiones de conexiones masivas para degradar el desempeño y debilitar la seguridad del sistema en general. (Denial of Service Attack).
5.1.1.4 Envío de mensajes voluminosos para saturar la capacidad de almacenamiento del servidor de correo.
5.1.1.5 Petición de envio masivo de mensajes a terceros por un nodo externo ajeno a la identidad del servidor de correo (SPAM attack).

Para evitar problemas del tipo (1) el tráfico de correo se hará a través del Firewall. El programa que recibe el correo inicialmente se ejecuta bajo un usuario no privilegiado eviatndo así el acceso a programas del sistema.

Para evitar problemas del tipo (2) el programa que recibe el correo interpreta las opciones peligrosas y las ignora, produciendo respuestas inocuas al intruso.

Para evitar problemas del tipo (3) el Firewall cuenta con un mecanismo que permite ir listando aquellos sitios que son característicos por este tipo de ataques y los bloquea.

Para evitar problemas del tipo (4) el programa en el Firewall solo acepta mensajes hasta un cierto tamaño, desechando aquellos mayores.

Para evitar problemas del tipo (5) el programa en el Firewall solo permite ser usado como agente de enlace a aquellos nodos cuya dirección IP y nombre estén

registrados debidamente en Intenert a través del Domain Name System (DNS) y que sean locales a la UDLA.

3.3.2 Servicios en el Interior.

Una de las premisas de este trabajo de tesis y de la política de la seguridad informática en la UDLA es que los usuarios internos son tan capaces de realizar ataques a la seguridad como los usuarios externos. La experiencia de varios años ha enseñado a los administradores de centros de cómputo con acceso a Internet que los usuarios pueden obtener los procedimientos para violar la seguridad de cualquier sistema operativo de una manera relativamente fácil. Todo esto lleva como implicación que los servicios internos tengan que ser asegurados en un mismo nivel que los accesos externos.

El Firewall

Este trabajo consiguió un Firewall de Gateway de Host Apantallado como se describe en el capítulo 2.7. De ahí se puede considerar que el Firewall es una coraza que le permite al administrador de seguridad diferenciar a los usuarios externos de los internos en caso de que un ataque se presente. Si se presenta un ataque a los nodos internos es por dos posibles razones:

1. Un intruso logró colarse por el Firewall y ahora posee una cuenta interna.
2. Un usuario interno decidió atacar un nodo.

En cualquiera de los dos casos, el Firewall permitirá extraer información de sus bitácoras para rastrear al intruso dentro o fuera de la red interna. La ausencia de interacción con el Firewall le dirá que el usuario es interno. La presencia de información le dirá que existe la posibilidad de que el intruso sea externo y esté usando una cuenta interna, tal vez, sin el conocimiento del dueño real de dicha cuenta.

La identidad final del usuario atacante siempre es hallada por la inspección de las bitácoras de los nodos internos y, en ocasiones, con la cooperación de administradores externos (en el caso de un ataque externo).

Ftp y Telnet Internos

Desde hace un par de años se ha impulsado el uso de telnet y Ftp encriptados dentro de la UDLA con un fracaso bastante grande. Por el momento tales servicios se usan de manera encriptada por los administradores del centro de cómputo y las herramientas de administración automáticas que en él se producen.

Se espera que el uso de tales herramientas se vayan haciendo más comunes a la par que se introducen los puntos de acceso a la red a través de clientes delgados (por el momento, las *sunrays*).

Aparte de usarse el Ftp y Telnet encriptados, los nodos de la UDLA tienen asociado un Wrapper (descrito en el capítulo 3.2.3), con lo cual se logra que tales servicios solo estén disponibles para los nodos apropiados y con el método apropiado (encriptado o no encriptado).

World Wide Web Interno

El servicio de Wen interno es ofrecido sin interacción del Firewall, lo cual es muy conveniente para el usuario y para el Firewall al evitar dicha carga. Por otro lado, internamente se cuentan con servidores que manejan conexiones encriptadas (a través de Secure Sockets Layer) las cuales se caracterizan por realizarse en el puerto 443 normalmente y, en lugar del prefijo *http* se usa el prefijo *https.*

Por ejemplo, el nodo *miudla.pue.udlap.mx* accesible tanto del interior como externamente a la UDLA es un portal para hacer pagos con cualquier tarjeta de crédito y sus transacciones son seguras (encriptadas) desde el inicio, lo cual se puede observar al identificar que el ícono de candado, que se presenta en la mayoría de los navegadores, aparece cerrado.

La dirección completa de miudla.pue.udlap.mx es entonces:

https://miudla.pue.udlap.mx:443

La interaccíon de este nodo con Bancomer no puede ser descrita en este documento para no violar las cláusulas firmadas ante esta institución.

Tampoco se puede mencionar la infraestructura de otros servidores que contienen información crítica para la operación de la UDLA.

Correo Electrónico Interno

El servicio de correo electrónico era y es ofrecido de la misma forma tanto para usuarios internos como externos y con las mismas auntentificaciones. La razón de esto es que las herramientas para lectura de correo electrónico no cuentan con un mecanismo de autentificación encriptada, esto es, que al pedir el *username* y el *password* del usuario

estos viajen encriptados. No hay que confundir esto con que el contenido de los mensajes viaje encriptado.

Este puede ser un punto débil para cualquier red de área local, ya que si los usuarios internos tienen que autentificarse con su servidor de correo electrónico en modo texto, entonces un usuario malintencionado puede espiar el tráfico hacia tales servidores y capturar una gran cantidad de *passwords.*

El efecto de esta debilidad se minimiza al poner la coraza del Firewall, ya que al sospechar que un passowrd está comprometido se rastrean únicamente a los usuarios internos.

Otros Servicios Internos

Una vez que el uso de servicios encriptados se haya permeado en la comunidad en general, se pueden liberar otros servicios de acceso entre nodos y de ejecución de comandos remotos seguros, tales como el Secure Shell (descrito en el capítulo 3.2.2), el Telnet y Ftp *Kerberizados*, correo electrónico con contenido encriptado y sistemas de archivos encriptados.