

# **Proyecto UNAM/Cray de Seguridad en el Sistema Operativo Unix**

Tesis que para obtener el título de  
**Ingeniero en Computación**  
presenta  
**Diego Martín Zamboni**

Facultad de Ingeniería  
Universidad Nacional Autónoma de México  
Junio de 1995.

**A Lali y Beto (mis papás),  
por estar ahí, siempre.**



# Índice General

<b>Introducción</b>	<b>xiii</b>
Alcance de este documento . . . . .	xiii
Estructura de este documento . . . . .	xiii
Convenciones utilizadas . . . . .	xv
Tipos de letras . . . . .	xv
Cómo interpretar URLs y especificaciones de FTP . . . . .	xv
Agradecimientos . . . . .	xvi
<b>I Antecedentes</b>	<b>1</b>
<b>1 Conceptos básicos</b>	<b>3</b>
1.1 ¿Qué es seguridad en cómputo y por qué es importante? . . . . .	3
1.1.1 Definición de seguridad . . . . .	3
1.1.2 Tipos de Seguridad . . . . .	4
1.2 Elementos de la seguridad . . . . .	5
1.2.1 Vulnerabilidades . . . . .	5
1.2.2 Amenazas . . . . .	6
1.2.3 Contramedidas . . . . .	8
1.3 Necesidad de concientización y conocimiento sobre seguridad en cómputo . . . . .	9
1.4 Diferencia entre <i>Seguridad en Red</i> y <i>Seguridad en el Sistema Operativo Unix</i> . . . . .	10
1.5 Seguridad en Supercómputo . . . . .	10
1.6 ¿Cuánta seguridad? . . . . .	12
1.6.1 Análisis de riesgos . . . . .	12
1.6.2 Reglamentaciones sobre seguridad . . . . .	14
<b>2 Aspectos históricos de la seguridad en cómputo</b>	<b>19</b>
2.1 Controles de la información . . . . .	19
2.2 Evolución de la computación . . . . .	19
2.3 Antecedentes de seguridad en cómputo a nivel mundial . . . . .	21
2.4 Antecedentes de seguridad en cómputo en la UNAM . . . . .	24

<b>3</b>	<b>Situación actual en la UNAM</b>	<b>27</b>
3.1	Estructura general de la red de cómputo en la UNAM . . . . .	27
3.1.1	Topologías y medios de comunicación usados en REDUNAM . . . . .	27
3.1.2	Protocolo TCP/IP . . . . .	28
3.2	Sistemas de cómputo utilizados en la UNAM . . . . .	28
3.2.1	Sistemas Unix . . . . .	29
3.2.2	Mainframes . . . . .	30
3.2.3	Computadoras personales (PCs) . . . . .	32
3.3	Situación de la red de supercómputo dentro de REDUNAM . . . . .	33
3.4	Sistemas de cómputo involucrados en el proyecto . . . . .	34
3.4.1	Sala de la Supercomputadora Cray . . . . .	34
3.4.2	Departamento de Supercómputo . . . . .	35
3.4.3	Laboratorio de Visualización . . . . .	35
<b>4</b>	<b>Problemática en la UNAM</b>	<b>37</b>
4.1	La confianza . . . . .	37
4.2	El cómputo administrativo . . . . .	37
4.3	Computadoras personales (PCs) . . . . .	39
4.4	Incidentes previos de seguridad en la UNAM . . . . .	39
4.4.1	Respuesta al incidente . . . . .	40
4.4.2	Las causas y los efectos . . . . .	41
<b>5</b>	<b>Recursos disponibles</b>	<b>43</b>
5.1	Recursos humanos . . . . .	43
5.1.1	Administradores . . . . .	43
5.1.2	Usuarios . . . . .	44
5.1.3	Personal directivo . . . . .	45
5.2	Recursos técnicos . . . . .	46
5.2.1	Herramientas comerciales . . . . .	46
5.2.2	Herramientas de dominio público . . . . .	46
5.2.3	Herramientas desarrolladas en la UNAM . . . . .	47
5.2.4	Herramientas de difusión de la información . . . . .	47
5.3	Recursos organizacionales y legales . . . . .	49
<b>II</b>	<b>Acciones tomadas</b>	<b>51</b>
<b>6</b>	<b>Acciones iniciales en DGSCA</b>	<b>53</b>
6.1	Reinstalación de sistemas operativos . . . . .	53
6.1.1	¿Por qué reinstalar sistemas operativos? . . . . .	53
6.1.2	Qué se hizo en DGSCA . . . . .	54
6.2	Instalación de herramientas de seguridad . . . . .	55
6.2.1	¿Qué se necesita? . . . . .	55
6.2.2	¿Qué herramientas? . . . . .	56
6.2.3	Qué se hizo en DGSCA . . . . .	58
6.3	Concentración de los reportes generados . . . . .	60

6.3.1	Medios de transferencia de la información . . . . .	61
6.3.2	Cómo se transfiere la información . . . . .	61
6.3.3	Qué se hace con los reportes . . . . .	61
6.4	Formación de un grupo de administradores . . . . .	62
<b>7</b>	<b>Grupo de Administración y Seguridad en Unix</b>	<b>63</b>
7.1	Fundación y forma inicial de operación . . . . .	63
7.1.1	¿Quiénes? . . . . .	63
7.1.2	Primera reunión . . . . .	64
7.1.3	Resultados de la primera reunión . . . . .	65
7.2	Crecimiento y evolución . . . . .	65
7.2.1	Segunda reunión . . . . .	65
7.2.2	Resultados de la segunda reunión . . . . .	66
7.2.3	Lista de correo electrónico y consolidación de GASU . . . . .	66
7.2.4	Nuevas áreas de acción . . . . .	68
7.3	Seminarios, cursos y talleres . . . . .	68
7.4	Difusión de las herramientas de seguridad . . . . .	69
7.5	Presente y futuro de GASU . . . . .	70
7.5.1	¿Que opina la gente? . . . . .	71
7.5.2	El futuro . . . . .	74
<b>8</b>	<b>Día Internacional de la Seguridad en Cómputo</b>	<b>75</b>
8.1	Antecedentes . . . . .	75
8.2	El DISC en la UNAM . . . . .	76
8.2.1	Convocatoria . . . . .	76
8.2.2	Respuesta . . . . .	76
8.2.3	Realización . . . . .	76
8.3	Resultados obtenidos y planes a futuro . . . . .	77
<b>9</b>	<b>Capacitación</b>	<b>79</b>
9.1	Conferencia de Seguridad en Cómputo y Comunicaciones . . . . .	79
9.2	Formación de un seminario interno sobre seguridad y criptografía . . . . .	80
9.3	Adquisición de bibliografía sobre seguridad . . . . .	81
9.4	Creación de una carpeta de seguridad . . . . .	82
9.5	Contacto con expertos en seguridad . . . . .	82
9.6	Plan de becarios de supercómputo . . . . .	83
<b>10</b>	<b>Desarrollo de Herramientas de Seguridad</b>	<b>85</b>
10.1	La problemática . . . . .	85
10.2	La solución . . . . .	86
10.3	New CARP . . . . .	86
10.3.1	Antecedente: COPS y CARP . . . . .	86
10.3.2	Una versión mejorada de CARP . . . . .	88
10.3.3	Estructura de CARP . . . . .	90
10.3.4	Diseño e implementación de NCARP . . . . .	93
10.3.5	El futuro de NCARP . . . . .	98

10.4	SAINT: Una herramienta de integración de análisis de seguridad . . . .	99
10.4.1	Antecedente: utilización de herramientas de seguridad . . . . .	99
10.4.2	Una nueva forma de ver los datos: SAINT . . . . .	99
10.4.3	Estado actual de SAINT . . . . .	99
10.4.4	Funciones y características de SAINT . . . . .	99
10.4.5	¿Qué hace SAINT? . . . . .	100
10.4.6	Recolección y homogeneización de datos . . . . .	101
10.4.7	Ordenamiento de los eventos . . . . .	104
10.4.8	Análisis de los eventos . . . . .	104
10.4.9	Presentación de resultados . . . . .	109
10.4.10	Módulo de control . . . . .	110
10.4.11	Archivo de configuración . . . . .	110
10.4.12	Consideraciones sobre SAINT . . . . .	111
10.4.13	El futuro de SAINT . . . . .	112
10.5	Dónde obtener las herramientas . . . . .	112
<b>11</b>	<b>Servicios de seguridad</b>	<b>113</b>
11.1	Difusión de Información . . . . .	113
11.1.1	Lista de correo electrónico <i>gasu</i> . . . . .	113
11.1.2	Lista de correo electrónico <i>cert-advisory</i> . . . . .	114
11.1.3	FTP anónimo de seguridad . . . . .	114
11.1.4	<i>World Wide Web</i> (WWW) . . . . .	116
11.1.5	Boletín de Supercómputo . . . . .	117
11.1.6	Asistencia a foros externos . . . . .	117
11.2	Revisión y consultoría de seguridad . . . . .	120
11.2.1	Revisión remota utilizando SATAN . . . . .	120
11.2.2	Análisis remoto utilizando NCARP . . . . .	121
11.2.3	Otros servicios, el futuro . . . . .	122
<b>12</b>	<b>Actividades Institucionales y Legales</b>	<b>123</b>
12.1	Asesorías a otras dependencias universitarias . . . . .	123
12.1.1	Instituto de Astronomía . . . . .	123
12.1.2	Unidad de Servicios de Cómputo Académico, Facultad de Ingeniería(USCAFI) . . . . .	124
12.2	Consideración de la seguridad en cómputo por parte de las autoridades universitarias . . . . .	124
12.3	Legislación universitaria sobre seguridad en cómputo . . . . .	125
12.4	Trabajo con otros organismos de seguridad . . . . .	125
12.4.1	CERT . . . . .	126
12.4.2	Laboratorio Nacional de Los Alamos . . . . .	126
<b>13</b>	<b>Resultados obtenidos y conclusiones</b>	<b>127</b>

<b>A</b>	<b>Servidor de listas de correo electrónico del Departamento de Supercómputo</b>	<b>129</b>
A.1	¿Qué es un servidor de listas de correo electrónico? . . . . .	129
A.2	¿Qué servidor se utiliza en el Departamento de Supercómputo? . . . . .	130
A.3	Configuración del servidor de listas . . . . .	131
A.4	Cómo se usa el servidor de listas de correo . . . . .	131
<b>B</b>	<b>Servidor de FTP anónimo del Departamento de Supercómputo</b>	<b>133</b>
B.1	¿Qué es un servidor de FTP anónimo? . . . . .	133
B.2	¿Qué servidor de FTP se utiliza en el Departamento de Supercómputo? . . . . .	133
B.3	Configuración del servidor de FTP anónimo . . . . .	134
<b>C</b>	<b>Servidor de HTTP del Departamento de Supercómputo</b>	<b>137</b>
C.1	¿Qué es un servidor de HTTP? . . . . .	137
C.2	¿Qué servidor se utiliza en el Departamento de Supercómputo? . . . . .	138
C.3	Configuración del servidor de HTTP . . . . .	138
C.4	Cómo se usa el servidor de HTTP . . . . .	138
<b>D</b>	<b>Herramientas de seguridad de dominio público</b>	<b>139</b>
D.1	Autenticación . . . . .	139
D.2	Firmas criptográficas . . . . .	141
D.3	Seguridad en red . . . . .	141
D.4	Monitoreo de red . . . . .	142
D.5	Monitoreo del sistema . . . . .	143
D.6	Herramientas generales de seguridad . . . . .	144
<b>E</b>	<b>Acciones iniciales a tomar sobre seguridad</b>	<b>147</b>
E.1	Acciones Correctivas . . . . .	147
E.2	Claves . . . . .	147
E.3	Programas de seguridad . . . . .	148
E.4	Comunicaciones . . . . .	149
E.5	Procedimientos . . . . .	149
E.6	Bibliografía . . . . .	150
<b>F</b>	<b>Manual de instalación de COPS, TCP-Wrapper y passwd+</b>	<b>151</b>
F.1	Introducción . . . . .	151
F.2	Obtencion de los Programas . . . . .	152
F.3	Descripcion de las Herramientas . . . . .	153
F.3.1	COPS . . . . .	153
F.3.2	TCP-Wrapper . . . . .	155
F.3.3	passwd+ . . . . .	158
F.4	Apéndice: Instalación del <b>nsyslog</b> en sistemas Ultrix . . . . .	161

<b>G</b>	<b>Detalles técnicos de las listas de correo electrónico <i>gasu</i> y <i>cert-advisory</i></b>	<b>163</b>
G.1	<i>gasu</i> . . . . .	163
G.1.1	Características . . . . .	163
G.1.2	Estadísticas . . . . .	163
G.2	<i>cert-advisory</i> . . . . .	164
G.2.1	Características . . . . .	164
G.2.2	Estadísticas . . . . .	164
<b>H</b>	<b>Programa de conferencias del DISC 1994</b>	<b>165</b>
<b>I</b>	<b>Cómo usar un servidor de FTP anónimo</b>	<b>169</b>
<b>J</b>	<b>Manual del usuario de New CARP</b>	<b>173</b>
J.1	Introducción . . . . .	173
J.2	Requerimientos para usar NCARP . . . . .	174
J.3	Cómo obtener NCARP . . . . .	174
J.4	Cómo instalar NCARP . . . . .	174
J.5	Utilización de NCARP . . . . .	175
J.5.1	Utilización básica . . . . .	175
J.5.2	Utilización avanzada . . . . .	176
<b>K</b>	<b>Manual del usuario de SAINT</b>	<b>177</b>
K.1	Introducción . . . . .	177
K.2	Requerimientos para usar SAINT . . . . .	177
K.3	Cómo obtener SAINT . . . . .	178
K.4	Cómo instalar SAINT . . . . .	178
K.5	Utilización de SAINT . . . . .	179
K.5.1	Utilización básica . . . . .	179
K.5.2	Utilización avanzada . . . . .	179
<b>L</b>	<b>Estado de la seguridad en la supercomputadora de la UNAM</b>	<b>183</b>
L.1	Actual status of security-related items . . . . .	183
L.2	What has been done in security-related issues . . . . .	184
L.3	References and resources . . . . .	185
L.4	Questions . . . . .	185
<b>M</b>	<b>Anuncio del servicio de revisión remota utilizando SATAN</b>	<b>187</b>

# Índice de Figuras

3.1	Red de supercómputo. . . . .	34
4.1	Cómo la confianza puede ocasionar que los problemas de seguridad se extiendan. . . . .	38
7.1	Versiones de Unix representadas en la encuesta de GASU . . . . .	72
10.1	Fragmento de un reporte generado por COPS . . . . .	87
10.2	Ejemplo de tabla generada por CARP . . . . .	88
10.3	Ejemplo de reporte generado por NCARP . . . . .	89
10.4	Formato de la base de datos de mensajes de NCARP. . . . .	94
10.5	Formato del archivo de configuración de NCARP. . . . .	97
11.1	Forma de registro de asistencia al DISC en WWW . . . . .	118
11.2	Encuesta sobre GASU en WWW . . . . .	119
B.1	Ejemplo de compresión y compactación automática de archivos utilizando wu-ftp . . . . .	135
I.1	Ejemplo de sesión en FTP . . . . .	171
J.1	Formato de reporte producido por NCARP . . . . .	175
J.2	Formato del archivo de configuración de NCARP . . . . .	176
K.1	Ejemplo de un reporte de SAINT . . . . .	180



# Índice de Tablas

1.1	Diferencia entre <i>Seguridad en Red</i> y <i>Seguridad en el sistema operativo Unix</i> . . . . .	11
1.2	Clases de seguridad definidas en el Libro Naranja, y ejemplos de cada una de ellas. . . . .	15
1.3	Resumen del Criterio de Evaluación de Sistemas de Cómputo Confiables (TCSEC) . . . . .	16
3.1	Principales plataformas Unix utilizadas en la UNAM. . . . .	30
3.2	Sistemas operativos de <i>mainframes</i> utilizados en la UNAM. . . . .	30
3.3	Estaciones de trabajo del Departamento de Supercómputo. . . . .	35
3.4	Estaciones de trabajo del Laboratorio de Visualización. . . . .	36
6.1	Herramientas de seguridad instaladas en los sistemas de Supercómputo y Visualización en la DGSCA. . . . .	60



# Introducción

Esta tesis, que se presenta para obtener el título de Ingeniero en Computación en la Facultad de Ingeniería de la Universidad Nacional Autónoma de México, expone el trabajo realizado por el autor en el campo de seguridad en Unix, principalmente de julio de 1993 a la fecha.

El objetivo de este proyecto es incrementar la seguridad en todos los sistemas de cómputo que hacen uso del sistema operativo Unix, desde la supercomputadora Cray hasta el gran número de estaciones de trabajo y computadoras personales que se utilizan con Unix en la UNAM y en otras instituciones externas.

## Alcance de este documento

La intención y el alcance de esta tesis se centran en dar a conocer las motivaciones del proyecto de seguridad, los elementos que han intervenido en su diseño y las acciones que se han tomado para implementarlo. Aparte de su intención como tesis de Licenciatura, pretende servir como referencia para administradores y usuarios que se interesen por la seguridad en Unix. En este documento se describen los aspectos logísticos, humanos y técnicos esenciales para implementar políticas y mecanismos sólidos de seguridad en una red de sistemas Unix.

## Estructura de este documento

Esta tesis está dividida en dos partes; incluye 13 capítulos y 13 apéndices.

**Parte I, Antecedentes.** Sienta los conceptos teóricos sobre los que se basa el trabajo de seguridad en Unix, así como las motivaciones, justificaciones y diseño del proyecto.

El Capítulo 1 describe los aspectos teóricos de la seguridad en cómputo. Se proporciona la definición de seguridad en cómputo y sus elementos, así como las razones por las cuales es un tema que debe tomarse en cuenta actualmente como un elemento indispensable de la cultura computacional. Particularmente, se habla de la seguridad en supercómputo, tema que interesa centralmente a este proyecto dado el ambiente en el que se desarrolla.

El Capítulo 2 presenta una cronología de los eventos que, en el transcurso de la segunda mitad de este siglo, han dado lugar a la seguridad en cómputo como una disciplina seria y establecida. También se mencionan los antecedentes que existen de seguridad en cómputo en la UNAM.

El Capítulo 3 describe el estado del cómputo en la Universidad en dos aspectos: infraestructura física y sistemas operativos. También se describen brevemente los sistemas de cómputo involucrados directamente en esta tesis, así como el papel del supercómputo dentro de la situación descrita.

El Capítulo 4 describe la situación de la seguridad en cómputo en la UNAM, tanto en los aspectos cubiertos por esta tesis (cómputo de investigación y, en particular, Unix) como en los que no se tratan directamente pero que están muy relacionados (cómputo administrativo y otros sistemas operativos). También se describe el incidente de seguridad que ocasionó el inicio oficial de este proyecto.

El Capítulo 5 presenta un análisis de los recursos humanos, técnicos y legales con los que se dispuso en este proyecto para atacar el problema de la seguridad en Unix.

**Parte II, Acciones tomadas,** describe el trabajo concreto realizado durante el transcurso del proyecto.

El Capítulo 6 describe las actividades realizadas en la DGSCA, “casa” del proyecto de seguridad, desde su inicio y continuando hasta la fecha.

El Capítulo 7 documenta uno de los aspectos principales del proyecto: la formación de un grupo humano, a nivel Universidad, interesado por la seguridad y la buena administración en el sistema operativo Unix.

El Capítulo 8 describe la organización, como parte del proyecto de seguridad, del Día Internacional de la Seguridad en Cómputo.

El Capítulo 9 describe las acciones que se han realizado para lograr la capacitación del personal que participa en el proyecto de seguridad.

El Capítulo 10 describe las herramientas que se han desarrollado como parte de esta tesis, para cubrir necesidades específicas de seguridad en la UNAM, y que no eran atendidas por ninguna de las herramientas ya existentes.

El Capítulo 11 describe los servicios que se ofrecen, como parte del proyecto de seguridad, a la comunidad de cómputo en México, tanto universitaria como externa. Éstos comprenden la difusión de información por distintos medios, así como la revisión de seguridad en sus diversas formas.

El Capítulo 12 cubre las acciones que se han realizado en forma conjunta con otras instituciones universitarias y con otros organismos de seguridad a nivel mundial. También se describen brevemente las actividades que se realizaron en el campo legal, aunque estos aspectos no son cubiertos por esta tesis.

El Capítulo 13 presenta las conclusiones de la tesis.

**Los apéndices** contienen generalmente, detalles técnicos de los temas tratados en los capítulos, así como documentos y otra información a la que se hace referencia en el texto de la tesis.

## Convenciones utilizadas

### Tipos de letras

Para facilitar la lectura del texto, se utilizan ciertas convenciones tipográficas para denotar diferentes elementos del mismo. Éstas son:

*Cursivas* Se utilizan para:

- Títulos de documentos, congresos y conferencias.
- Términos en idiomas diferentes al español (como *password*).
- En ejemplos de comandos, para indicar algún elemento variable (como *archivo*).
- Resaltar algunos elementos importantes del texto cuando son mencionados por primera vez.

**Negritas** Se utilizan para:

- Nombres de claves en Unix (como **root**).

**Courier** Se utiliza en ejemplos, para representar la salida de comandos. También se utiliza para denotar nombres de archivos (como `/etc/passwd`). Cuando está subrayado, representa algo que debe ser tecleado por el usuario.

*Inclinada* Se utiliza para nombres de listas de correo electrónico y de máquinas (por ejemplo, *gasu* y *ds5000*).

**comando(n)** Se refieren a comandos, llamadas al sistema o funciones de biblioteca, o *funcion(n)* indicando la sección de páginas de manual del sistema en el que se puede encontrar su descripción (por ejemplo, **login(1)**, *read(2)* o *crypt(3)*).

**SansSerif** Se utiliza para denotar nombres de programas, como COPS y passwd+.

### Cómo interpretar URLs y especificaciones de FTP

En esta tesis se hace mención frecuentemente a recursos disponibles en la red Internet, tales como archivos, documentos y manuales.

Para hacer mención a estos recursos se utilizan elementos conocidos como URLs (*Uniform Resource Locator*). Se puede considerar que un URL es una extensión al concepto tradicional de la ruta de un archivo dentro de un disco: un URL no solamente puede apuntar a un archivo en un directorio, sino que ese archivo y ese directorio pueden estar en cualquier máquina en la red, y puede ser accesado a través de diferentes métodos.

A continuación se describen los tipos de URL utilizados en esta tesis.

### URLs de archivos

Supóngase que un documento llamado README se encuentra almacenado en un servidor de FTP anónimo llamado `ftp.super.unam.mx`, dentro del directorio `/pub/security`. El URL para ese archivo es:

```
ftp://ftp.super.unam.mx/pub/security/README
```

El directorio `/pub` de ese servidor es, entonces:

```
ftp://ftp.super.unam.mx/pub/
```

De esta forma se hace referencia, en esta tesis, a todos los archivos y directorios disponibles a través de FTP anónimo en la red, y que pueden ser accesados utilizando cualquier cliente de FTP existente.

### URLs de HTTP

Este tipo de URL se utiliza para especificar la ubicación de documentos disponibles en WWW (*World Wide Web*) a través del protocolo conocido como HTTP (*HyperText Transfer Protocol*).

Si existe un archivo `seguridad.html` que se encuentra en el servidor de HTTP llamado `www.super.unam.mx`, en el directorio `/pub`, su URL es:

```
http://www.super.unam.mx/pub/seguridad.html
```

Para acceder a documentos de HTTP es necesario contar con un visualizador apropiado, tal como Mosaic, NetScape o Lynx.

## Agradecimientos

Hay demasiadas personas a quienes quiero agradecer por haber contribuido de manera invaluable a la realización de esta tesis. Voy a mencionarlos sin ningún orden en particular, a medida que vienen a mi mente los nombres. Cualquier omisión es culpa de mi mala memoria, y no de un deseo intencional.

A Susana, cuyo amor, apoyo, comprensión (y a veces regaños) me acompañaron durante este proyecto y me impulsaron a seguir siempre adelante.

A Martha, mi directora de tesis, por su incondicional apoyo en mis proyectos, por las muchísimas ideas valiosas que aportó, por su guía firme, y por la presión de trabajo, a veces difícil, pero sin la cual esta tesis no se hubiera terminado en el tiempo apropiado.

A Alejandro, por todos los comentarios, ideas y críticas aportados, y por las bromas, la plática y el apoyo que hizo el trabajo siempre más fácil de sobrellevar.

A Rafael y a Gerardo Cisneros, por haber sido mis principales críticos (después de Martha), y por haber aportado muchas ideas valiosas.

A mis papás, Lali y Beto, por su apoyo incondicional, su confianza sin límites, su amor, y sencillamente, por haber estado ahí durante toda mi vida, y haber hecho posible, en primera instancia, la realización de este proyecto.

A toda la gente del Supercómputo y Visualización que me apoyó, me criticó, me dio ideas y prestó sus máquinas para mis experimentos de seguridad.

Al personal de Cray Research de México, que participó constantemente y de manera entusiasta en las actividades realizadas, y contribuyeron su apoyo, sus ideas y sus comentarios.

A la gente de otros departamentos de DGSCA y de otras dependencias de la UNAM, que también se prestaron a mis nuevas ideas, las criticaron y apoyaron, y contribuyeron a la consolidación de mi proyecto a nivel UNAM. Entre ellos: Eduardo Sacristán Ruiz-Funes, David Vázquez, Víctor Jiménez, Gaby Medina, Mike De Leo, Rafael Durán y José Luis Orozco.

A las autoridades de DGSCA por su apoyo al proyecto: Dr. Enrique Daltabuit Godás, Dr. Alberto Alonso y Coria y Dr. Víctor Guerra Ortiz.

Finalmente, a los becarios del plan de becarios de supercómputo 94–95, que les tocaron por primera vez curso y proyectos de seguridad, y que están, en este momento, contribuyendo a que el proyecto de seguridad siga creciendo y sea cada vez más y mejor.

Gracias a todos.

**Parte I**

**Antecedentes**



# Capítulo 1

## Conceptos básicos

### 1.1 ¿Qué es seguridad en cómputo y por qué es importante?

Durante la segunda mitad de la década de los '80 se le comenzó a dar mucha publicidad a los incidentes de seguridad en cómputo. Los medios masivos de comunicación descubrieron una nueva “mina de oro” al dar a conocer al público los detalles —a veces inventados— de algunos de los más conocidos problemas de este tipo, como el Ataque a HBO en 1986, el Gusano de Internet [Spa91] y la “Conexión KGB” [Sto89] en 1988.

Aunque estos incidentes han servido como entretenimiento al público, también han servido para ir despertando la conciencia, tanto en el mismo público como en los administradores y usuarios de sistemas de cómputo, de que la seguridad en cómputo debe ser, actualmente, una preocupación real. Las situaciones que antes pertenecían a las historias de ciencia ficción ahora se encuentran en la vida real con una frecuencia cada vez mayor.

Sin embargo, la seguridad en cómputo va mucho más allá de impedir que los “chicos malos” se roben la información y se la vendan al enemigo. También implica proteger a los usuarios contra sus propios errores (como borrar accidentalmente sus archivos), realizar a tiempo los respaldos de los discos, etc. Los problemas de seguridad cotidianos son mucho menos “atractivos” que la persecución remota de un espía alemán a través de redes telefónicas y de telecomunicaciones, pero son igual de importantes y, sobre todo, son mucho más frecuentes, de manera que es vital estar preparado para atenderlos y resolverlos.

#### 1.1.1 Definición de seguridad

Una de las definiciones más generales de seguridad es la siguiente [GS92, pág. 4]:

“Un sistema de cómputo es seguro si se puede confiar en que él y su *software* se comportarán como se espera que lo hagan, y que la información almacenada en él se mantendrá inalterada y accesible durante tanto tiempo como su dueño lo desee.”

Esta definición reconoce la amplitud y subjetividad del concepto. De acuerdo a la idea popular de seguridad en cómputo, su único objetivo es impedir que la información caiga en manos no autorizadas, es decir, la confidencialidad de los datos. Aunque este aspecto de la seguridad es muy importante, no lo es todo. Se considera la existencia de tres tipos principales de seguridad en cómputo [RG92, pág. 9]: confidencialidad, integridad y disponibilidad.

### 1.1.2 Tipos de Seguridad

#### Confidencialidad

Un sistema de cómputo no debe permitir que la información contenida en él sea accesible a nadie que no tenga la autorización adecuada. Esta es la parte más conocida de la seguridad, por ser la más fácilmente entendible. Algunos de los incidentes más sonados de seguridad han consistido en la violación de la confidencialidad de un sistema.

#### Integridad y Autenticidad

Un sistema de cómputo seguro no debe permitir modificaciones no autorizadas a los datos contenidos en él. Esto comprende cualquier tipo de modificaciones:

- Por errores de *hardware* y/o *software*.
- Causados por alguna persona de forma intencional.
- Causados por alguna persona de forma accidental.

En algunos medios, principalmente en cuanto a telecomunicaciones, se habla de un tipo de seguridad relacionado con la integridad llamado autenticidad. Éste se refiere a contar con un medio de verificar quién está enviando la información, así como poder comprobar que los datos no fueron modificados durante su transferencia.

#### Disponibilidad

La información puede estar sana y salva en el sistema, pero de poco sirve si los usuarios no tienen acceso a ella. La disponibilidad significa que los recursos del sistema, tanto de hardware como de software, se mantendrán funcionando de forma eficiente, y que los usuarios los podrán utilizar en el momento en que los necesiten. También significa que el sistema sea capaz de recuperarse rápidamente en caso de ocurrir un problema de cualquier especie.

Este es el tipo de seguridad menos tomado en cuenta comúnmente, por ser el más “transparente”. Sin embargo, es fundamental para el funcionamiento del equipo, pues de nada sirven todos los demás tipos de seguridad si no se puede utilizar la máquina, aunque sea para revisar si dichos esquemas de seguridad se están cumpliendo. Mucha gente no toma en cuenta que mantener el sistema funcionando es en sí un tipo de seguridad.

## 1.2 Elementos de la seguridad

En todo análisis de seguridad en cómputo se deben tomar en cuenta tres elementos: vulnerabilidades, amenazas y contramedidas.

### 1.2.1 Vulnerabilidades

Las vulnerabilidades son los puntos débiles del sistema, a través de los cuales la seguridad se puede ver afectada. Las principales vulnerabilidades que se consideran comúnmente son las que se mencionan a continuación.

#### **Vulnerabilidades físicas.**

Por más seguro que sea el sistema operativo y por más cuidadosos que sean los usuarios y administradores, alguien que tenga acceso físico a la máquina puede hacer prácticamente cualquier cosa, empezando por apagarla o destruirla. Para evitar esto es necesaria la utilización de mecanismos tradicionales de seguridad física, como puertas, chapas, candados, rejas, alarmas, vigilantes, etc.

#### **Vulnerabilidades naturales.**

Todos los edificios están expuestos a tragedias naturales, como incendios, terremotos, inundaciones, etc. Las computadoras son particularmente sensibles a cualquier alteración en su medio ambiente, así como a la humedad, los cambios de temperatura, el polvo, etc.

#### **Vulnerabilidades de *hardware* y *software*.**

El equipo físico y los programas fallan, eso es bien sabido. Lo que a veces no se tiene en cuenta es que dichas fallas pueden producir huecos en la seguridad de un sistema, o por sí mismas ocasionar algún daño. Por ejemplo, un programa que falle podría poner mal los permisos de algunos archivos críticos, abriendo la posibilidad de alguna acción ilegal. Si el equipo físico falla el problema es aún peor, pues pueden perderse directamente los datos.

#### **Vulnerabilidades de almacenamiento.**

Los discos, diskettes, cintas y todos los dispositivos utilizados para almacenamiento secundario son muy delicados, y pueden ser dañados incluso por una partícula de polvo, así como ser robados, pues normalmente se transportan muy fácilmente.

Por otro lado, los mecanismos de borrado de datos en casi todos los sistemas de cómputo no son irrevocables: los datos siguen estando ahí, aunque ya no sean accesible tan fácilmente. Esto abre la posibilidad de que esos datos sean recuperados por alguien que esté interesado en ellos y que tenga acceso al sistema.

**Vulnerabilidades por emanaciones.**

En ambientes donde se requiere un muy alto nivel de seguridad es importante tener en cuenta este aspecto. Todo lo que se maneja en los sistema de cómputo son, a fin de cuentas, señales electrónicas, y todas las señales de este tipo producen emanaciones electromagnéticas. Por muy débiles que sean, dichas emanaciones pueden ser interceptadas y descifradas, con lo que se puede saber qué es lo que se está haciendo en la computadora.

**Vulnerabilidades de comunicaciones.**

Tener una computadora bien comunicada con el mundo exterior es muy conveniente para los usuarios del sistema, pero mientras más comunicada esté, más aumenta el peligro de accesos no autorizados. Si una computadora tiene comunicación con el exterior, los mensajes pueden ser interceptados, desviados o falsificados. De hecho, estas son las formas más comunes de lograr accesos no autorizados a un sistema. También las líneas que conectan al sistema con el mundo exterior pueden ser dañadas físicamente.

**Vulnerabilidades humanas.**

El elemento humano de un sistema de cómputo representa la mayor de todas las vulnerabilidades, pues los humanos tenemos fallas y debilidades. Casi siempre la seguridad de todo el sistema está en manos de un administrador humano. Si éste comete un error, decide realizar alguna acción indebida o desconoce el sistema, éste está en grave peligro. También los usuarios, operadores y cualquier otra persona que tenga contacto con la máquina puede ser engañado u obligado a divulgar una clave, abrir una puerta, o hacer cualquier otra cosa que pueda poner en peligro la seguridad del sistema.

**1.2.2 Amenazas**

Se conoce como amenazas a aquellos elementos que pueden utilizar alguna de las vulnerabilidades mencionadas en la sección anterior para causar algún daño en el sistema. Se clasifican principalmente en tres tipos: naturales, no intencionales e intencionales.

**Amenazas naturales.**

Son las amenazas a las que todo elemento físico está expuesto: incendios, temblores, terremotos, rayos, fallas de energía eléctrica, etc. Casi nunca es posible prever este tipo de desastres, pero es posible detectarlos rápidamente (mediante alarmas y detectores, por ejemplo), minimizar los daños causados (utilizando mecanismos automáticos) y tratar de evitar sus ocurrencias en lo posible (implantando políticas de uso que reduzcan los riesgos).

**Amenazas no intencionales.**

Son los peligros causados por la ignorancia. Por ejemplo, un usuario que por error borra sus propios archivos, o un administrador sin la capacitación necesaria que hace cambios erróneos en un archivo de configuración del sistema.

Estas amenazas también se presentan por descuido: dejar caer una cinta magnética, poner un disco cerca de un imán, etc.

A pesar de que no tienen nada de atractivo para la opinión pública, son mayores y mucho más frecuentes los daños causados por acciones no intencionales que por malicia.

### **Amenazas intencionales.**

Estas son las amenazas que llaman la atención, las que acaparan las primeras planas de los periódicos cuando se conocen y las que cautivan la imaginación del público. Sin embargo (y afortunadamente) son las menos frecuentes. Los ataques intencionales pueden ser realizados por dos tipos de personas:

**Externos.** Pueden ser intrusos ocasionales, que por suerte o por un poco de habilidad pudieron obtener acceso al sistema. También pueden ser intrusos decididos que atacan el sistema con un objetivo específico, que puede ser la obtención de dinero, el chantaje a la empresa, la obtención de información clasificada, etc. También el objetivo puede ser únicamente la diversión, el reto de vencer los mecanismos de seguridad y la fama que ello puede crear en el bajo mundo de la computación.

Vale la pena en este punto definir a los atacantes mediante un término. A continuación se presenta esta definición.

Normalmente se utiliza la palabra *hacker* para referirse a las personas que hacen uso no autorizado de un sistema de cómputo. Sin embargo, este término tiene una larga y honorable historia, pues surgió inicialmente para referirse a una persona con un gran interés en las computadoras, en explorarlas al máximo y llevarlas hasta sus límites. Esto significa que existen muchas personas que se consideran a sí mismas como *hackers*, y a quienes no les agrada su utilización para denotar a un criminal. Es por ello que se ha acuñado el término *cracker* para denotar a quienes utilizan los sistemas de cómputo de forma ilegal. Con este término o con la palabra *intruso* se hará referencia en este texto a los “criminales del cómputo”.

**Internos.** A pesar de que normalmente se le tiene más miedo a los atacantes externos, las estadísticas muestran que alrededor del 80% de los ataques exitosos son llevados a cabo por personal interno, usuarios legales del sistema, pero que hacen uso del acceso con el que cuentan para llevar a cabo actividades ilícitas.

En algunas ocasiones, se trata de empleados inconformes o demasiado ambiciosos, que hacen uso de sus privilegios para causar algún daño. Los ejemplos clásicos son el del encargado de la nómina que hace cambios para que todos los datos sean borrados en caso de que él sea despedido, o el encargado de los datos financieros que desvía unos pocos centavos mensuales a su cuenta personal.

En los más de los casos, sin embargo, se trata tan solo de empleados distraídos, que olvidan cambiar su clave de acceso, o la anotan en un papel y lo dejan sobre su escritorio, o dejan los impresos confidenciales en una pila en el suelo. Cualquiera de estos detalles puede ser aprovechado por gente mal intencionada —casi siempre también internos a la empresa— para hacer mal uso del sistema o de la información.

### **1.2.3 Contramedidas**

Aquí tenemos, finalmente, las acciones que se pueden tomar para prevenir o minimizar los daños. Se clasifican en cuatro grupos: seguridad interna, seguridad en comunicaciones, seguridad física y seguridad humana.

#### **Seguridad interna**

Se refiere a la protección de la información almacenada en un sistema de cómputo, mientras dicha información permanece dentro del mismo. Se concentra en las características del sistema operativo que controlan el acceso al sistema y a la información.

Muchas de las medidas que se discuten en este documento pertenecen a este punto.

#### **Seguridad en comunicaciones**

Se estudia aquí la protección de la información que está siendo transmitida de un sistema a otro, ya sea por teléfono, cable, microondas, satélite, etc. La preocupación es básicamente que la información llegue a su destino sin alteraciones de ninguna especie, y que efectivamente venga de donde debería venir.

Esta tesis no trata ningún aspecto de seguridad en comunicaciones, por tratarse de un tema que puede proporcionar material para otra tesis completa. Cuando mucho se harán referencias a este aspecto cuando esté relacionado con alguno de los que aquí se discuten.

#### **Seguridad física**

Consiste en la protección del equipo de cómputo físico contra daños de cualquier especie, ya sea por desastres naturales o por intrusos. Los métodos utilizados pueden ser muy variados, desde los tradicionales candados y llaves hasta mecanismos mucho más sofisticados, como tarjetas inteligentes y dispositivos biométricos.

En este trabajo tampoco se discute ningún aspecto de seguridad física de los sistemas de cómputo.

#### **Seguridad humana**

Este es el tema principal de esta tesis. Consiste en la labor que se tiene que realizar con todas las personas que están en contacto con un sistema de cómputo: usuarios, operadores, administradores y hasta personal de mantenimiento. De poco sirve toda la seguridad

que el sistema pueda tener internamente si las personas que lo utilizan no están conscientes de ello y no toman las medidas necesarias para que los mecanismos de seguridad sean efectivos.

Este es un problema muy grave en la UNAM, y es el que se atacó directamente durante el desarrollo de este trabajo.

### **1.3 Necesidad de concientización y conocimiento sobre seguridad en cómputo**

En sus inicios, las computadoras eran grandes y escasas. En ese entonces, las únicas amenazas para el sistema eran los propios administradores y usuarios, de manera que la protección de la información era relativamente fácil. Además, todos los que trabajaban con las computadoras en esas épocas eran científicos muy especializados, que tenían un propósito bien específico en mente, y sobre los cuales normalmente pesaban estrictas normas de seguridad que hacían muy difícil realizar cualquier acción ilícita.

Con la utilización creciente de redes de computadoras, sin embargo, la situación ha cambiado. Ahora los usuarios de un sistema compartido están en rangos que van desde niños de 10 años hasta personal altamente capacitado. Las tecnologías de comunicación han hecho posible la “omnipresencia” de las computadoras, y su acceso a todo tipo de personas.

Debido a esto, existen ahora muchos más puntos vulnerables en los sistemas. El administrador ya no tiene tanto control sobre lo que sucede, y tampoco los usuarios tienen siempre tanta conciencia de lo que están haciendo. Incluso es muy probable encontrarse con usuarios malintencionados que aprovecharán cualquier resquicio en la seguridad del sistema para obtener algún beneficio.

Como en la gran mayoría de los problemas, la mejor solución es prevenir los incidentes de seguridad y no permitir que sucedan. Y como siempre, la mejor manera de prevenirlos es mediante la educación. Lo ideal es inculcar en los usuarios de las computadoras, desde sus inicios, los conceptos básicos de seguridad. De esta manera se logran varios objetivos:

1. Los usuarios adquieren conciencia de la importancia de la seguridad, y dejan de considerarla como algo de ciencia ficción.
2. Al considerar la seguridad como algo “natural” al cómputo, los usuarios tienen más facilidad para aceptar y aplicar las medidas de seguridad sugeridas, y de hecho las buscan por ellos mismos sin esperar a que el administrador las imponga o las proponga.
3. Los usuarios tendrán mayor conocimiento sobre las técnicas que pueden aplicar ellos, como usuarios, para incrementar la seguridad del sistema y su productividad en el mismo.
4. El conocimiento de los usuarios propicia un incremento substancial en la seguridad del sistema, pues su labor, conjuntamente con la del administrador, produce una disminución considerable en los puntos vulnerables del mismo.

La razón principal para crear conciencia y difundir el conocimiento entre los usuarios de un sistema de cómputo sobre temas de seguridad es que todas las medidas que tome el administrador pueden llegar a ser inútiles si no se cuenta con la participación de los usuarios. En un sistema Unix convencional los usuarios pueden ignorar todas las medidas de seguridad y proporcionar, intencional o accidentalmente, puntos de acceso fácil para los intrusos y para cualquier otro tipo de problemas.

Con la proliferación de las redes de computadoras, la conciencia que cada usuario tenga sobre su papel en la seguridad se vuelve más importante, pues un solo usuario que proporcione acceso fácil a su cuenta puede afectar la seguridad del sistema completo, e incluso de otras máquinas conectadas a la misma red.

## 1.4 Diferencia entre *Seguridad en Red* y *Seguridad en el Sistema Operativo Unix*

Con la creciente utilización comercial del sistema operativo Unix, la seguridad se ha convertido en el “tema de moda” en muchas publicaciones, trabajos, tesis y discusiones de todo tipo. Sin embargo, la seguridad es un tema sumamente amplio, y es frecuente confundir seguridad en el sistema operativo Unix con la seguridad en una red de cómputo. La tabla 1.1 resume las diferencias existentes entre estos dos conceptos.

## 1.5 Seguridad en Supercómputo

Para poder entender las implicaciones que tiene la seguridad en supercómputo, primero es necesario saber qué es el supercómputo. Aunque no existe una definición formal, se considera generalmente que supercómputo se refiere a los sistemas de cómputo con más potencia de cálculo existentes en un momento dado, distantes de los sistemas convencionales por al menos un orden de magnitud, y que tienen capacidad de resolver problemas de frontera muy grandes, que los sistemas convencionales no pueden atacar.

Debido a estas características, las supercomputadoras son sistemas sumamente caros, y las instituciones que cuentan con ellas imponen severas restricciones para tener acceso a ellas, requiriéndose normalmente la presentación de un proyecto que justifique la utilización de la máquina.

Estas mismas características, sin embargo, convierten a las supercomputadoras en objetivos muy atractivos para un *cracker*. Teniendo acceso a una supercomputadora se tiene acceso a una capacidad de cómputo enorme, y muy posiblemente también a muchas aplicaciones que normalmente no se encuentran en los sistemas de cómputo convencionales.

Es por esto que en las supercomputadoras se presta particular atención a todos los aspectos de seguridad. Normalmente no se conecta una supercomputadora directamente a una red, sino que se utilizan máquinas conocidas como *firewalls*, que permiten establecer un control estricto sobre qué información es la que puede entrar y salir de la supercomputadora.

Seguridad en red	Seguridad en el sistema operativo Unix
<p>La seguridad en red se refiere a la seguridad de los datos mientras son transferidos de un sistema a otro.</p> <p>Seguridad física de las líneas de transmisión.</p> <p>Cifrado de los datos.</p> <p>Autenticación de los datos recibidos, tanto en su contenido como en su origen.</p>	<p>Todos los aspectos de operación interna del sistema.</p> <p>Autenticación de los usuarios del sistema, es decir, que todas las personas que hagan uso del sistema sean en realidad usuarios autorizados.</p> <p>Autenticación de los procesos que se ejecutan en el sistema, es decir, verificar que todo lo que se ejecuta en la máquina sea un proceso autorizado y con una función útil y bien definida.</p> <p>Control de acceso a los datos contenidos en el sistema, es decir, que solamente tengan acceso a los datos contenidos en la máquina las personas que estén autorizadas a tenerlo.</p> <p>Vigilancia y control de la integridad de los datos contenidos en el sistema, de manera que no puedan sufrir modificaciones no autorizadas.</p> <p>Registro (<i>log</i>) de las actividades relevantes en el sistema, para tener una base de información sobre la cual se puedan llevar a cabo análisis en caso de que sea necesario para determinar qué está pasando en un momento dado en el sistema.</p>

Tabla 1.1: Diferencia entre *Seguridad en Red* y *Seguridad en el sistema operativo Unix*

La UNAM cuenta con una supercomputadora Cray Y-MP4/432 desde noviembre de 1991, y que en diciembre de 1993 fue actualizada al modelo Y-MP4/464 mediante la adición de memoria principal. Este sistema es utilizado actualmente por más de 400 investigadores de todo el país, en más de 200 proyectos de investigación. La utilización de la supercomputadora ha resultado en la publicación de múltiples artículos a nivel internacional, y ha servido para dar un fuerte impulso a la investigación científica en México.

Desgraciadamente, la presencia de la supercomputadora en la UNAM también ha traído los problemas mencionados anteriormente. Por esta razón, es muy grande el interés de las autoridades universitarias en incrementar el nivel de seguridad en la supercomputadora. Como las máquinas de Cray utilizan el sistema operativo Unix, esta máquina se ve comprendida dentro de los sistemas cubiertos por el proyecto tratado en esta tesis.

## 1.6 ¿Cuánta seguridad?

La seguridad en cómputo, como de cualquier otro tipo, cuesta tiempo, dinero y, sobre todo, esfuerzo. Durante mucho tiempo se consideró a la seguridad como “una tarea más” del administrador del sistema, que podía ser llevada a cabo en el tiempo libre, como algo secundario. Sin embargo, cada vez más se entiende a la seguridad como una actividad fundamental y con “vida propia”. La seguridad es una disciplina completa en sí misma, y a la cual una persona —o incluso más— se puede dedicar de tiempo completo.

### 1.6.1 Análisis de riesgos

Sin embargo, como en todo, es necesario establecer niveles y prioridades. No tiene caso caer en la “fiebre tecnológica” de tener siempre “lo último y lo más grande” en medidas de seguridad, gastando en el proceso grandes cantidades de dinero, si la información que se va a proteger vale menos de lo que se va a gastar (esta regla no siempre se aplica, pero es bastante válida). Por eso, antes de comenzar a planear la seguridad del sistema, es necesario hacerse las siguientes preguntas:

**¿Qué se quiere proteger?** Es muy importante determinar qué información se tiene en el sistema, y que tan importante es para la organización en que se está trabajando. Esta valoración se tiene que hacer de forma individual en cada caso, pues la información que es muy valiosa para una organización puede no tener ningún valor para otra.

**¿Contra qué se quiere proteger?** Para no incurrir en gastos innecesarios, es importante determinar cuáles son los riesgos reales que corre la información. Sería poco realista, por ejemplo, pensar que los datos de las finanzas personales de una persona van a ser de interés para un gobierno extranjero, salvo que dicha persona sea el Presidente de la República. Por otro lado, puede haber información que sea de muy alto interés, pero que es tan inaccesible —estando, por ejemplo, en una computadora fuertemente vigilada, con un solo usuario autorizado y sin conexiones de red— que no tiene caso ninguna medida adicional para protegerla.

Existen peligros de los cuales todos los sistemas se tienen que proteger, y son los . Cualquier sistema puede sufrir un incendio o un terremoto, por lo que la medida básica de seguridad debe ser realizar respaldos periódicos de la información. Fuera de estos cuidados fundamentales, se tiene que hacer un análisis para ver cuales son los riesgos reales que corre el sistema.

**¿Cuánto tiempo, dinero y esfuerzo se puede invertir?** Este es el punto más importante de los tres, pues es el que puede determinar en última instancia qué es lo que se va a hacer en la realidad. Los tres costos mencionados son:

**Tiempo:** Para tener un nivel de seguridad alto, es necesario que alguien dedique tiempo a configurar correctamente los parámetros de seguridad del sistema, configurar el ambiente de trabajo de los usuarios, revisar y fijar los permisos de acceso de los archivos, ejecutar programas de monitoreo de seguridad, revisar las bitácoras (*logs*) del sistema, etc. No se puede pretender tener seguridad completa si la persona encargada de estas actividades es un empleado temporal de medio tiempo a quien solo le interesa cumplir con lo indispensable para que le paguen.

**Dinero:** Como consecuencia lógica del punto anterior, el tener alguien que se encargue de las actividades de seguridad, y que lo haga de forma responsable, cuesta dinero. También cuesta dinero adquirir los productos de seguridad que se vayan a utilizar, ya sean programas o equipo de seguridad especializado. En muchas ocasiones el encargado de seguridad tiene que convencer no solo a los usuarios para que tomen conciencia de la importancia de la seguridad, sino también a los directivos para que entiendan que los gastos en seguridad valen realmente la pena.

**Esfuerzo:** Finalmente, establecer y mantener un nivel adecuado de seguridad puede significar un esfuerzo considerable por parte del encargado. Esto es particularmente cierto cuando ocurren problemas de seguridad. En [Sto89], Clifford Stoll narra su persecución de un intruso alemán que estaba accediendo sin autorización al sistema a su cargo. Y narra como a veces tenía que ir corriendo a su laboratorio a media noche cuando el sistema le notificaba (a través de un *beeper*) que el intruso había establecido conexión.

Del nivel de compromiso que el encargado de seguridad tenga con el sistema, dependerá en gran medida el nivel real de seguridad existente.

Es importante en este punto analizar también los costos que tendría la pérdida o acceso no autorizado a la información. Dependiendo de la información de la que se trate y, en su caso, de quién haga el acceso no autorizado, el efecto puede ser pérdidas monetarias, poner en peligro la seguridad nacionalseguridad de un país, pérdida de ventaja competitiva, pérdida de confianza pública, etc.

El contestar estas preguntas de forma objetiva y realista es lo que en algunas ocasiones se conoce como la realización de un análisis de riesgos, en el que se evalúan los riesgos que se corren, conjuntamente con los costos que tiene evitarlos, para lograr llegar a un punto de equilibrio entre lo que se puede gastar y lo que se quiere en cuanto a seguridad.

Es importante tener en cuenta que es prácticamente imposible hacer que un sistema sea totalmente seguro. Se atribuye comunmente a Gene Spafford el siguiente comentario [M<sup>+</sup>93]:

“ El único sistema que es realmente seguro es uno que está apagado y desconectado, guardado en una caja fuerte de titanio, encerrado en un *bunker* de concreto, rodeado por gas neurotóxico y guardias armados muy altamente pagados. Aún así, no apostaría mi vida por él.”

La seguridad se puede incrementar hasta niveles muy altos, pero siempre habrá una manera de obtener acceso no autorizado al sistema, aunque esto signifique un gasto millonario y la inversión de varios años de trabajo. Es por esto que la seguridad es un tema tan subjetivo: la forma de medir varía para cada organización e individuo.

### 1.6.2 Reglamentaciones sobre seguridad

A medida que la seguridad ha ido adquiriendo importancia no solo como una disciplina científica, sino también como una herramienta en aplicaciones reales, era de esperarse que se quisieran establecer reglamentaciones al respecto. El país en el que se ha dado gran parte de la revolución computacional es los Estados Unidos de Norteamérica, de manera que también es de esperarse que sea en dicho país donde se tengan los mayores avances en cuanto a dichas reglamentaciones. El Departamento de Defensa de los Estados Unidos de Norteamérica (DOD) publicó en agosto de 1983 la primera versión del *Department of Defense Trusted Computer System Evaluation Criteria (Criterio de Evaluación de Sistemas de Cómputo Confiables del Departamento de Defensa)*, el cual fue revisado en diciembre de 1985 para dar lugar a la versión actual [DoD85]. Este documento es conocido comunmente como *Orange Book (Libro Naranja)*, debido al llamativo color de su cubierta.

#### El Libro Naranja

La motivación principal para la creación del Libro Naranja[DoD85] fue ofrecer una norma que permitiera cuantificar el nivel de seguridad en los sistemas de cómputo. Esta necesidad surge debido a que cada organización, y cada tipo de información dentro de cada organización, requiere de diferentes tipos de seguridad. En particular, el Libro Naranja define las “reglas” para la compra de equipo de cómputo confiable por parte del Gobierno de los Estados Unidos.

El Libro Naranja define cuatro amplias divisiones jerárquicas de seguridad. En orden creciente de confiabilidad, dichas divisiones son:

- D** Seguridad mínima.
- C** Seguridad discrecional.
- B** Seguridad obligatoria.
- A** Seguridad verificable.

Clase	Nombre	Ejemplos
D	Seguridad mínima	Se consideran dentro de esta clase los sistemas que fueron sometidos a la evaluación del DOD y fallaron. Si fueran evaluados formalmente, los sistemas operativos “primitivos” de las computadoras personales, como MS-DOS para PCs y System 7 para Macintosh, quedarían en esta clase.
C1	Seguridad discrecional	IBM: MVS/RACF (una versión posterior fue clasificada como C2). Se considera comunmente que el Unix estándar queda en esta clasificación, aunque nunca ha sido evaluado formalmente.
C2	Seguridad de acceso controlado	Computer Associates International: ACF/2/MVS Digital Equipment Corporation: VAX/VMS 4.3 Gould: UTX/32S Hewlett-Packard: MPE V/E Wang Laboratories: SVS/OS CAP 1.0
B1	Seguridad por etiquetas	AT&T: System V/MLS IBM: MVS/ESA SecureWare: CMW+ Unisys: OS 1100
B2	Seguridad de protección estructurada	Honeywell Information Systems: Multics Cray Research, Inc.: Trusted UNICOS Trusted Information Systems: Trusted XENIX
B3	Seguridad por dominios	Honeywell Federal Systems: XTS-200 (en evaluación hasta 1992).
A1	Seguridad verificable	Honeywell Information Systems: SCOMP Boeing Aerospace: SNS

Tabla 1.2: Clases de seguridad definidas en el Libro Naranja, y ejemplos de cada una de ellas.

Cada división consiste a su vez de una o más clases numeradas, donde los números mayores indican un nivel mayor de seguridad. Por ejemplo, la división C contiene dos clases, donde C2 ofrece mayor seguridad que C1. La división B contiene tres clases, y la división A solamente contiene una. La tabla 1.2 (tomada de [RG92]) muestra cada una de las clases existentes, junto con algunos ejemplos de sistemas que han sido evaluados con éxito en cada una de ellas.

Cada clase se define por una serie de criterios que un sistema debe cumplir para obtener su clasificación. La tabla 1.3 muestra una comparación entre las clases existentes, mostrando las características específicas que requiere cada clase y, en términos generales, cómo los requerimientos se incrementan de clase en clase.

El propósito del Libro Naranja, según se define en el mismo, es lograr tres objetivos básicos:

	C1	C2	B1	B2	B3	A1	
Control de acceso discrecional							Políticas de Seguridad
Reutilización de Objetos							
Etiquetas							
Integridad de etiquetas							
Exportación de Información Etiquetada							
Exportación de Dispositivos Multinivel							
Exportación de Dispositivos Uninivel							
Etiquetamiento de salida legible por humanos							
Control de Acceso Obligatorio							
Etiquetas de Sensibilidad de Sujetos							
Etiquetas de dispositivos							
Identificación y Autenticación							Contabilidad
Auditoría							
Ruta confiable							
Arquitectura del sistema							Aseguramiento
Integridad del sistema							
Pruebas de seguridad							
Especificación y Verificación del diseño							
Análisis de canales secretos							
Manejo de elementos confiables							
Manejo de configuración							
Recuperación confiable							
Distribución confiable							
Guía de usuario de características de seguridad							Documentación
Manual de elementos confiables							
Documentación de pruebas							
Documentación de diseño							

 No existe el requerimiento para esta clase.  
 Requerimiento nuevo o mejorado para esta clase.  
 No existe requerimiento adicional para esta clase.

Tabla 1.3: Resumen del Criterio de Evaluación de Sistemas de Cómputo Confiables (TCSEC)

**Medición:** Proveer a los usuarios con una medida de qué tanto pueden “confiar” en un sistema dado, de acuerdo a su clasificación de seguridad.

**Guía:** Dar a los proveedores de equipo una guía para que sepan qué características incorporar en sus equipos si quieren cumplir con las especificaciones oficiales.

**Adquisición de equipo:** Proveer un criterio para la selección de equipo al momento de su adquisición. Una vez que un equipo ha sido certificado como perteneciente a una cierta clase, los compradores pueden estar seguros de que dicho equipo proporciona un cierto nivel mínimo de seguridad, y no tienen que revisar distintos aspectos uno por uno.



## Capítulo 2

# Aspectos históricos de la seguridad en cómputo

### 2.1 Controles de la información

La seguridad en cómputo es un tema “de moda” actualmente, pero no es de ninguna manera nuevo. En realidad, la seguridad en cómputo es solamente el giro más reciente de la seguridad de la información. Desde que el hombre fue capaz de almacenar y transmitir información, ha sentido la necesidad de protegerla contra aquellos que pudieran hacer uso no autorizado de ella. Una vez que los gobiernos se dieron cuenta también de la importancia de la información, dicha necesidad fue avalada por medidas gubernamentales que establecían los mecanismos para proteger y controlar el flujo de la información.

A lo largo de la historia, siempre que se ha desarrollado alguna nueva manera de procesar la información, ha surgido inmediatamente la preocupación de cómo controlar esa tecnología, sobre todo por parte de los gobiernos y las grandes empresas, para quienes la información es el recurso más valioso que existe. De hecho, es un debate todavía no resuelto el decidir si los gobiernos tienen derecho a controlar las tecnologías de protección de la información, tales como el cifrado de datos.

La computación no es más que la forma más nueva, eficiente y flexible de manejar la información. La utilización de dispositivos electrónicos conlleva múltiples ventajas con respecto a otras formas, pero también incluye su propio conjunto de desventajas, y su propio conjunto de debates, en cuanto a quién y cómo se debe controlar la protección de la información.

### 2.2 Evolución de la computación

En sus inicios, la computación era privilegio de muy pocos. Los escasos centros de cómputo existentes giraban alrededor de grandes máquinas centrales, atendidas por personal altamente especializado. A estas máquinas los usuarios comúnmente nunca las veían. Los programas y datos eran proporcionados a través de tarjetas perforadas, que

eran entregadas a los operadores para que ellos las introdujeran al sistema. Al usuario solamente se le entregaban los resultados impresos de los cálculos.

Posteriormente, con la aparición de las terminales de video, los usuarios podían tener acceso a la computadora a través de una pantalla y un teclado. Estas terminales estaban conectadas directamente al sistema central, y normalmente estaban ubicadas en salas dentro del mismo edificio, protegidas por diversos medios “tradicionales” (como guardias y cerraduras) contra el acceso no autorizado.

Bajo estas condiciones, la seguridad en cómputo se reducía prácticamente a la protección física del equipo y de los respaldos de la información. La seguridad en cómputo era un aspecto más de la seguridad general de la planta.

Ni siquiera los usuarios autorizados del sistema eran un problema grave. Poca gente sabía como utilizar una computadora, y aquellos que conocían sus secretos eran considerados como personal con muy alta capacitación, de manera que a nadie se le hubiera ocurrido desconfiar de ellos.

Sin embargo, las cosas han cambiado actualmente. La tecnología avanza rápidamente, y los últimos 30 años han sido testigos de progresos vertiginosos en el área de la computación. La reducción de costos ha significado que mucha más gente tenga acceso a sistemas de cómputo cada vez más poderosos. Y las redes de computadoras han representado la globalización de las computadoras, al hacer posible la comunicación entre sistemas ubicados, a veces, en puntos opuestos del planeta.

Estos factores contribuyeron a que cada vez más gente tuviera conocimientos sobre cómputo. También contribuyeron a que las universidades y las instituciones educativas en general comenzaran a incluir la computación en sus planes de estudio, y a adquirir equipo de cómputo para uso de los estudiantes y profesores.

El impacto en el procesamiento de la información y la comunicación ha sido dramático. Actualmente se puede acceder a información almacenada en cualquier parte del planeta con la misma facilidad (o más) con la que se extrae un expediente de un archivo.

Sin embargo, esta flexibilidad y facilidad de uso también propició más abusos. Ahora no es tan fácil controlar quiénes están accediendo a la información. Además de preocuparse por los usuarios autorizados y por los daños físicos, las personas encargadas de sistemas de cómputo deben preocuparse por quién podría estar intentando acceder de forma no autorizada a la computadora. Actualmente, a través de una línea telefónica, alguien puede entrar a un sistema de cómputo, hacer cualquier tipo de daño, y desaparecer sin dejar rastro.

Actualmente, ha sido finalmente aceptado que la computación es un recurso universalmente útil. Casi ninguna actividad humana en nuestros tiempos se lleva a cabo sin la ayuda de computadoras. Casi todas las computadoras tienen algún tipo de capacidades de comunicación remota, y casi todas estas computadoras son vulnerables a ser atacadas. Es por esto que los administradores y los usuarios de los equipos de cómputo deben estar más atentos, más capacitados, y más conscientes que nunca, para juntos presentar un frente más sólido a los problemas de seguridad que surgen aparejados a las tecnologías de cómputo modernas.

## 2.3 Antecedentes de seguridad en cómputo a nivel mundial

Las actividades relacionadas con la seguridad en cómputo iniciaron al mismo tiempo que comenzó a difundirse la utilización de las computadoras en los centros industriales, educativos y gubernamentales. La siguiente lista presenta una relación cronológica de los hechos más notorios en este campo:

**1950s:** Desarrollo del primer estándar de seguridad TEMPEST.

Establecimiento de la primera organización gubernamental de seguridad en los Estados Unidos: el *U. S. Communications Security Board* (COMSEC, Consejo de Seguridad en Comunicaciones).

**1967:** La *Conferencia Conjunta de Cómputo de Primavera* (Spring Joint Computer Conference) de este año se considera como el sitio donde se llevó a cabo la primera presentación detallada de seguridad en cómputo. En esta sesión de conferencias, coordinada por Willis H. Ware de la Corporación RAND, se presentaron ponencias que tocaban diversos problemas de seguridad en cómputo, enfocadas a un auditorio técnico.

En Octubre de ese año, el Departamento de Defensa de los Estados Unidos (DOD) formó un grupo de trabajo auspiciado por el Consejo de Ciencias de la Defensa dentro de la Agencia de Proyectos de Investigación Avanzada (ARPA, actualmente conocido como DARPA), cuya función era examinar sistemas y redes de cómputo, identificar vulnerabilidades y amenazas, e introducir métodos para proteger y controlar el acceso a las computadoras, sistemas y redes del DOD. Los resultados de este trabajo fueron publicados en 1970.

**1968:** La Oficina Nacional de Normas (*National Bureau of Standards* o NBS) realiza un estudio inicial para evaluar las necesidades de seguridad en cómputo del gobierno.

**1970:** Se publica el documento *Security Controls for Computer Systems* [War79], resultado del grupo de trabajo mencionado arriba. Este documento, publicado inicialmente como clasificado, fue una publicación histórica para la seguridad en cómputo, pues fijó la pauta que seguirían muchos grupos de trabajo e investigaciones en años subsecuentes.

**1972:** El DOD emite una directiva [DoD78] y un manual adjunto [DoD79] que establecían una política consistente para todos los controles y técnicas de seguridad en cómputo dentro del DOD.

NBS patrocina una conferencia sobre seguridad en cómputo en conjunción con la ACM (*Association for Computing Machinery*).

En este año, y sentando un precedente a nivel mundial, Japón adopta una *Política Nacional de Cómputo*, en la que ya se consideraban aspectos de seguridad.

**1973:** NBS inicia un programa para estudiar estándares de desarrollo para mecanismos de seguridad en cómputo.

NBS lanza una invitación para presentar técnicas de cifrado de datos que pudieran ser utilizadas como base para algoritmos de cifrado. Uno de los algoritmos presentados a concurso es el DES (*Data Encryption Standard*), que posteriormente sería adoptado como el estándar de cifrado de datos por el Gobierno de los Estados Unidos.

**1970s:** Durante el transcurso de esta década comienzan a surgir, normalmente bajo el patrocinio del DOD y de empresas privadas, los llamados *tiger teams* (“equipos de tigres”), que eran equipos de *hackers* que intentaban violar los mecanismos de seguridad de los equipos de cómputo, reportando sus resultados con el fin de corregir los problemas encontrados. Estos equipos ayudaron a localizar muchos problemas existentes en sistemas de cómputo de la época [AMP74, BPC78, KS74], pero tenían el problema de que atacaban fallas específicas en sistemas específicos, y su mecanismo de trabajo era prácticamente “prueba y error”. Por consiguiente, era muy difícil que encontraran todos los problemas existentes, y finalmente se vio que eran necesarios mecanismos más generales y formales de análisis de seguridad [Fau84].

También durante los 70's, pero con efectos más duraderos, se iniciaron muchos trabajos de investigación formales sobre seguridad, sentando las bases del estudio teórico de la seguridad en cómputo. Durante este tiempo se desarrollaron los conceptos de política de seguridad y modelo de seguridad. También se realizó el primer modelo matemático de un sistema de seguridad multinivel [BL73], y que ha sido la base de muchos estudios teóricos e implementaciones hasta nuestros días. De hecho, la gran mayoría de los sistemas de seguridad multinivel existentes actualmente se basan en los conceptos de Bell-LaPadula.

Finalmente, durante esta activa década en cuanto a seguridad se comenzaron los primeros desarrollos de sistemas de cómputo “seguros”. Estos proyectos, también patrocinados generalmente por el Gobierno de los Estados Unidos, se concentraron en la realización de prototipos para *kernels* de seguridad. Un *kernel* de seguridad es la parte de un sistema operativo que controla el acceso a los recursos del sistema. El resultado más importante de estos proyectos fue el *kernel* de seguridad para el sistema Multics. Otros sistemas desarrollados fueron el PDP-11/45 (desarrollado por Mitre Corporation para Digital Equipment Corporation) y el PDP-11/70 (desarrollado por UCLA también para DEC).

**1977:** El Departamento de la Defensa de los Estados Unidos anuncia la Iniciativa de Seguridad en Cómputo del DOD, bajo el auspicio de la Secretaría de la Defensa para la Investigación y la Ingeniería (*Secretary of Defense for Research and Engineering*). Esta iniciativa tenía como objetivo atraer la atención y los recursos nacionales hacia la seguridad en cómputo, y lo logró organizando una serie de seminarios y talleres de trabajo con la industria y el gobierno para tratar problemas de seguridad. Estos talleres resultaron en la publicación de numerosos reportes [RM80, Rut80].

Un aspecto digno de mencionarse es que, desde entonces, se reconoció el hecho de que ningún sistema de cómputo puede considerarse completamente seguro. El reporte del taller realizado en 1977 dice [RG92]:

De acuerdo a cualquier definición razonable de “seguro”, ningún sistema operativo actual puede considerarse “seguro”. . . Esperamos que el lector no interprete esto como que no puede manejarse de forma segura información altamente clasificada en una computadora, pues por supuesto esto es hecho todo el tiempo. El punto es que los mecanismos de control internos de los sistemas operativos actuales tienen una integridad muy baja para . . . aislar de forma efectiva a un usuario en un sistema de la información que está en niveles de seguridad “más altos” de los que él tiene permitido.

También como resultado de los talleres, a la Corporación Mitre le fue asignada la tarea de desarrollar un conjunto inicial de criterios para la evaluación de la seguridad en cómputo que pudiera ser utilizada para garantizar el nivel de confianza que pudiera ser depositado en un sistema de cómputo.

En este año también se acepta al algoritmo DES (*Data Encryption Standard*) como estándar federal de procesamiento de información (*Federal Information Processing Standard*, FIPS), con lo cual se convirtió en el método oficial de protección de información en las computadoras de las agencias del Gobierno de los Estados Unidos.

- 1979:** La Oficina del Secretario de la Defensa realiza una serie de seminarios públicos sobre la Iniciativa de Seguridad en Cómputo del DOD.
- 1981:** El 2 de Enero de este año, se forma el Centro de Seguridad en Cómputo del DoD (*DoD Computer Security Center*, CSC) dentro de la Agencia Nacional de Seguridad (*National Security Agency*, NSA), con el fin de continuar el trabajo comenzado por la Iniciativa de Seguridad en Cómputo del DOD.
- 1983:** Se publica la primera versión del Trusted Computer System Evaluation Criteria (TCSEC) [DoD85], comunmente conocido como *Libro Naranja* debido al llamativo color de su cubierta, y que se convertiría (hasta nuestros días) en el estándar por el cual se mide la seguridad de los sistemas de cómputo a nivel mundial. Este documento fue publicado por el NCSA, basándose en los trabajos previos de la Corporación Mitre [Nib79] y en trabajos de investigación como el de Bell-LaPadula [BL73].
- 1984:** El presidente de los Estados Unidos Ronald Reagan firma, el 17 de Septiembre de este año, la *Directiva de Decisión de Seguridad Nacional 145* (National Security Decision Directive 145, NSDD 145), también conocida como la *Política Nacional de Seguridad de Sistemas Automáticos de Telecomunicaciones e Información*. Este documento tuvo consecuencias importantes en el mundo de la seguridad en cómputo, al establecer reglas para el manejo de información clasificada, y autorizar a la NSA a aconsejar a la iniciativa privada.

**1985:** El CSC se convierte en el Centro Nacional de Seguridad en Cómputo (*National Computer Security Center*, NCSC), al expandirse sus responsabilidades a todas las agencias federales de los Estados Unidos.

NSA combina sus funciones de seguridad en cómputo y comunicaciones bajo la Dirección de Sistemas de Seguridad de la Información, conocido comunmente como INFOSEC.

**1986:** Se expide la *Ley sobre Fraude y Abuso en Cómputo* (Ley pública 99-474 en los Estados Unidos), que establece las penalizaciones por acceso no autorizado o fraudulento a computadoras del gobierno.

**1988:** Se expide la *Ley sobre la Seguridad en Cómputo* (Ley pública 100-235 en los Estados Unidos), que establece obligaciones para agencias del gobierno en cuanto al manejo de información clasificada.

**1990:** Se publica en Inglaterra la *Ley sobre Mal Uso del Cómputo*, que define los delitos de seguridad en cómputo y las penas aplicables por ellos.

**1992:** Se publica el *Trusted Information Technology Security Evaluation Criteria (IT-SEC)* en Alemania. Este documento es la contraparte europea del *Libro Naranja*, y pretende tener una visión más amplia que pueda convertirse, en algún momento, en un estándar internacional de seguridad en cómputo.

## 2.4 Antecedentes de seguridad en cómputo en la UNAM

Desgraciadamente, esta es una sección muy breve de este documento. La seguridad en cómputo es un tema que, a nivel mundial, tiene poco tiempo de adquirir importancia. En muchos países, y México entre ellos, ha sido un problema al que no se ha prestado ninguna atención hasta muy recientemente, a pesar de que el problema ya existía<sup>1</sup>.

La UNAM no es la excepción. De acuerdo a la información que fue posible adquirir, los problemas de seguridad han existido desde hace muchos años, pero nunca se les ha atacado de manera organizada.

Se realizó una entrevista con el Sr. Rafael Durán, jefe del Departamento de Operación de la DGSCA, y nos proporcionó algunos datos interesantes, entre los que vale la pena mencionar:

- Desde 1975 se tenían problemas de seguridad en cómputo. Estos eran con los sistemas Burroughs. Ya había en ese entonces en la Universidad gente con la capacidad y el interés de romper las barreras de seguridad impuestas por el sistema,

---

<sup>1</sup>Una excepción notable a esta afirmación es la industria bancaria. Debido a los altos requerimientos de seguridad inherentes a este ramo, los bancos han contado desde hace bastante tiempo con mecanismos estrictos de seguridad en cómputo. Sin embargo, dichos mecanismos casi siempre han sido importados del extranjero, y comunmente los mismos encargados de las computadoras en estas instituciones no los conocen en detalle: adquieren los sistemas de cómputo y los utilizan en calidad de "cajas negras", sin saber qué es lo que ocurre en su interior. Esto ocasiona que dicho personal no pueda darle mantenimiento a los sistemas de seguridad y esté, para efectos prácticos, a la merced de la honestidad del proveedor.

por “el simple gusto de hacerlo”. Contra estas violaciones de seguridad nunca fue posible tomar alguna acción formal debido a la falta de legislación al respecto, así como las fricciones y conveniencias políticas que, desgraciadamente, siempre han invadido los ámbitos académico y científico en nuestra Universidad.

- Los sistemas Unisys, utilizados hasta la fecha en la UNAM, cuentan con un subsistema llamado GUARD que controla todos los aspectos de seguridad. En estos sistemas, por estar diseñado para aplicaciones administrativas, el nivel de seguridad es considerablemente mayor que en sistemas abiertos como Unix.
- Los sistemas Wang, utilizados durante un tiempo en la UNAM, son, según el Sr. Durán, los más estrictos en cuanto a controles de seguridad de todos los que él ha conocido. En estos sistemas, además de los controles tradicionales de acceso (*login y password*), se cuenta con *sub-passwords* que controlan el acceso a diferentes subsistemas. Incluso dentro de cada archivo de datos, el acceso se puede restringir por campo e incluso, en algunos casos, por *byte* de información.
- En los equipos de administración de los sistemas Burroughs se tenían personas encargadas de diferentes aspectos del sistema operativo, pero en ningún momento hubo alguien encargado de la seguridad del mismo.

Como se puede observar, la seguridad en cómputo ha sido un campo prácticamente inexplorado en la UNAM. En los últimos años las computadoras han proliferado más que nunca en la Universidad, pero la situación de seguridad sigue siendo la misma. Este problema es, justamente, el que da origen a esta tesis. El trabajo realizado durante este proyecto pretende incrementar el nivel de la seguridad en cómputo en nuestra casa de estudios, mediante el ataque sistemático a los problemas más importantes y la utilización al máximo de los recursos con los que se cuenta.



## Capítulo 3

# Situación actual en la UNAM

El primer paso en todo proyecto debe ser la evaluación de la situación inicial, y una definición clara del problema. Para establecer el contexto en el que se desarrolla este proyecto, el presente capítulo describe los principales elementos involucrados, así como los antecedentes que le dieron importancia a su realización.

### 3.1 Estructura general de la red de cómputo en la UNAM

REDUNAM es el medio de comunicación entre las computadoras de la UNAM, sobre las cuales trata este proyecto. Por lo tanto, es importante realizar una descripción general de la estructura de esta red [dRds].

#### 3.1.1 Topologías y medios de comunicación usados en REDUNAM

La estructura principal de REDUNAM es un anillo de FDDI (una fibra óptica activa y una de respaldo que pueden transportar información hasta 100 Mbps) que enlaza a 5 enrutadores principales. Conectadas a ellos se encuentran las redes locales de cada dependencia: las que se encuentran en el campus de C.U. son enlazadas por fibra óptica. Aquellas que se hallan fuera de él, se comunican con REDUNAM a través de alguno de los siguientes medios:

1. Dentro del área metropolitana
  - Radio módem.
  - Líneas conmutadas o privadas.
  - Microondas.
  - RDI (Red Digital Integrada).
2. Resto de la República Mexicana
  - RDI (Red Digital Integrada).

- Enlaces satelitales.

En las redes propias de la UNAM las topologías más empleadas son variantes de Ethernet: en primer lugar se tienen las redes tipo estrella, conocida también como red de par trenzado pues este es el medio físico con el que se construyen. Es posible encontrar este tipo de redes complementado con verticales de coaxial grueso en edificios de varios pisos. El segundo medio más empleado es el coaxial delgado, aunque su uso empieza a decaer debido a sus desventajas frente al par trenzado.

Las redes de Token Ring se encuentran en franca desaparición.

### 3.1.2 Protocolo TCP/IP

Una red de datos de tales características requiere de un protocolo de comunicaciones tal que:

1. Permita la conexión transparente entre diferentes clases de computadoras: PC's, *mainframes*, sistemas Unix, Macintosh, etc, así como poder convivir con sistemas operativos de red que se utilizarán en las redes locales.
2. Sea fácil de configurar y requiera pocos ajustes de acuerdo al crecimiento de la red.
3. Sea altamente confiable bajo cualquier condición operativa, y en caso necesario cuente con herramientas poderosas para la corrección de errores. También deberá brindar al administrador facilidades para el monitoreo y mantenimiento preventivo del funcionamiento de la red.
4. Esté diseñado expresamente para redes de área amplia o metropolitana, ofreciendo también la posibilidad de atender apropiadamente redes de área local.

El conjunto de protocolos TCP/IP se perfila como la solución natural a esta lista de requisitos. Es además, el protocolo estándar *de facto* para la comunicación en Internet. Sobre él pueden instalarse sistemas operativos de red tales como Windows NT y sus variantes, LAN Manager, Lantastic, etc., así como NetWare.

## 3.2 Sistemas de cómputo utilizados en la UNAM

El cómputo en la UNAM se ha ido desarrollando de forma continua y un tanto desordenada desde que en 1958 se instaló el primer sistema de cómputo en la Universidad (un sistema IBM-650). A partir de entonces, las distintas dependencias, facultades e institutos de la UNAM comenzaron una cadena de utilización y adquisición de equipo de cómputo. A medida que transcurrieron los años y avanzó la tecnología, esto resultó en una gran diversidad de equipo instalado en la UNAM. Actualmente, se encuentran conectados a REDUNAM una gran variedad de sistemas, desde computadoras personales hasta la supercomputadora Cray, pasando por toda clase de micro, mini y macrocomputadoras que son utilizadas con muy distintas aplicaciones a lo largo y ancho de nuestra casa de estudios.

A continuación se describen los principales tipos de sistemas de cómputo utilizados en la UNAM, y sus principales características en relación al proyecto de seguridad.

### 3.2.1 Sistemas Unix

Unix es uno de los sistemas operativos más antiguos en uso activo actualmente, pero no más obsoletos. Nació en los años 70 en los Laboratorios Bell de AT&T, pero su uso se extendió y popularizó, y actualmente es el sistema operativo estándar *de facto* en los ambientes de cómputo académico y de investigación.

Con el tiempo, en vez de envejecer y morir, Unix ha ido adquiriendo cada vez más fuerza. Alguna vez restringido a los ambientes académicos, Unix actualmente es utilizado con frecuencia cada vez mayor por empresas comerciales que se han dado cuenta de sus múltiples ventajas.

Esta evolución también ha llevado a la diversidad. Unix ya no es el sistema operativo universalmente estándar que era hace 20 años. Aunque en sus aspectos básicos de utilización sigue siendo el mismo, actualmente existen en el mercado casi tantas versiones de Unix como marcas de computadoras, y cada una de ellas tiene detalles —aunque sean pequeños— que la hacen diferente de las demás.

Esta gran diversidad afecta a la seguridad en un aspecto: muchos fabricantes de Unix, en su afán de mejorarlo, han añadido o modificado cosas sin realizar un análisis serio de las consecuencias que dichas modificaciones puedan tener. Muchos problemas de seguridad en la historia de Unix han surgido de interacciones inesperadas entre los programas que componen al sistema operativo, o entre dichos programas y las aplicaciones utilizadas por los usuarios. Es responsabilidad de los fabricantes de Unix tener en cuenta, antes de hacer una modificación al sistema, que el programa que están modificando puede interactuar con muchos otros, pues así es la filosofía de Unix. Aunque existen organizaciones que dictan estándares en cuanto a las versiones de Unix desarrolladas (por ejemplo, POSIX y OSF), estas organizaciones por lo regular no revisan la implementación de los estándares, y en ese aspecto el desarrollador de Unix queda en completa libertad. El no considerar cuidadosamente los efectos que dichas interacciones pueden tener en el funcionamiento del programa puede tener consecuencias catastróficas en muchos aspectos, incluyendo la seguridad del sistema.

Por otro lado, es un hecho que Unix, en sus inicios, no tenía la seguridad como una de las preocupaciones principales. Esto se percibe claramente en la siguiente declaración por parte de uno de los diseñadores de este sistema operativo:

[Unix] No fue diseñado desde al principio para ser seguro. Fue diseñado con las características necesarias para poder darle mantenimiento a la seguridad.

—Dennis Ritchie

Unix es un sistema operativo que fue diseñado en un ambiente de trabajo compartido, en donde la protección de los datos no era realmente importante. Con la utilización de Unix en otros ambientes, sin embargo, esta protección se convierte en una preocupación fundamental.

La tabla 3.1 resume las principales plataformas de cómputo Unix que se utilizan en la UNAM, junto con el nombre de la versión de Unix utilizada en cada una de ellas.

Fabricante	Sistema operativo
Cray Research, Inc.	UNICOS
DEC	Ultrix, OSF/1
Hewlett-Packard	HP-UX
Linus Torvalds	Linux
Santa Cruz Operation	SCO Unix
Silicon Graphics	Irix
Sun Microsystems	SunOS, Solaris

Tabla 3.1: Principales plataformas Unix utilizadas en la UNAM.

Fabricante	Sistema Operativo
Burroughs	MCP
Control Data	NOS/VE
DEC	VMS
IBM	MVS
Unisys	MCP

Tabla 3.2: Sistemas operativos de *mainframes* utilizados en la UNAM.

### 3.2.2 Mainframes

Durante mucho tiempo, las computadoras grandes fueron el único medio para la realización de cómputo “serio”. Estas máquinas, conocidas comúnmente como *mainframes*, tienen la característica de ser físicamente grandes, por lo que regularmente necesitan una sala especial, acondicionada para su uso. También requieren de un equipo de gente “a su servicio”, que esté constantemente atenta de que todo funcione de forma correcta.

La tabla 3.2 muestra en forma resumida los principales sistemas operativos de *mainframes* que han sido utilizados en la UNAM en el transcurso de los años.

Aunque la supercomputadora Cray califica como *mainframe* por su tamaño físico, utiliza el sistema operativo Unix, y por eso se le considera dentro de la sección anterior.

#### Sistemas Unisys

Actualmente, las únicas *mainframes* que continúan en uso activo dentro de la UNAM son las computadoras Unisys. En estos sistemas, hasta la fecha, se sigue llevando casi todo el cómputo administrativo de la Universidad, como nóminas, historias académicas, etc. En la UNAM se utilizan sistemas Unisys de los modelos A12 y A9.

Para obtener mayor información sobre el estado de la seguridad en el sistema operativo de las máquinas Unisys, se realizó una entrevista con el Ing. Fernando Baz, Coordinador Técnico del Departamento de Apoyo a Sistemas Unisys de la Dirección de Cómputo para la Administración Académica de la UNAM. La información obtenida en esta entrevista se puede resumir en los siguientes puntos:

- Unisys proporciona como parte opcional de su sistema operativo un subsistema llamado InfoGuard, diseñado para establecer en un sistema Unisys un nivel de seguridad C2.
- Entre las funciones principales de InfoGuard están:
  1. Asegurar que los usuarios cambien periódicamente sus *passwords*.
  2. Cifrar los archivos importantes del sistema.
  3. Cifrar información que se transfiere por la red.
  4. Proporcionar capacidad de cifrado de datos para los usuarios.
  5. Establecer controles de acceso mediante *passwords* adicionales a ciertos subsistemas y tareas administrativas, fomentando la división de trabajos en la administración de la máquina.
  6. Realización de respaldos automáticamente y comparación periódica de los respaldos con la información almacenada en el sistema.
- En el sistema Unisys A12 ubicado en el edificio del IIMAS, utilizado por la Dirección de Cómputo para la Administración Académica, y en el que se manejan las historias académicas de los alumnos de la Universidad, no se utiliza el sistema InfoGuard.
- InfoGuard sí se utiliza en los sistemas Unisys A9 ubicados en el edificio de DGSCAD en la calle de Pitágoras, donde se procesa la nómina de la UNAM.
- Aún sin utilizar InfoGuard, el sistema operativo de Unisys proporciona hasta tres claves de acceso:
  1. El *user code* y *password* de acceso al sistema operativo.
  2. El *access code*, que proporciona acceso a subsistemas específicos de la máquina.
  3. El *user code* y *password* de CANDE, que es la interfaz normal de los usuarios con la máquina.
- El sistema estándar de Unisys también permite controlar los horarios y lugares (terminales) de acceso de los usuarios al sistema.
- La principal dependencia universitaria que utiliza la A12 es DGAE (Dirección General de Administración Estudiantil).
- La integridad de los datos se asegura mediante respaldos: DGAE realiza entre 3 y 4 respaldos de su información diariamente. De esta manera se asegura, dentro de un rango razonable, de tener siempre su versión más reciente. Sin embargo, no se realiza ninguna verificación de que los datos no sean modificados de forma no autorizada.
- Los sistemas Unisys están conectados también a REDUNAM.

- El control de acceso a través de la red se basa en el hecho de que el protocolo TCP/IP implementado en los sistemas Unisys no permite la detección dinámica de direcciones electrónicas. Por esto, solamente se puede acceder al sistema a través de la red desde máquinas que estén explícitamente declaradas en la computadora Unisys.
- Las máquinas declaradas para permitir acceso a la A12 son principalmente computadoras de DGAE (en su gran mayoría PC's) y algunos otros sistemas grandes de la UNAM, como la Cray, y el servidor Unix de la DCAA (llamado *tzetza*).
- La emulación de terminales en Unisys también es muy limitada, haciendo aún más difícil el acceso a través de la red.
- En la base de datos de usuarios de Unisys se puede definir quiénes tienen permiso de realizar transferencias de información al exterior del sistema a través de la red.
- Los registros de bitácora del sistema solamente se revisan cuando ocurre algún problema, y no de forma rutinaria.
- En el tiempo que el Ing. Baz lleva trabajando con los sistemas Unisys se han presentado problemas de pérdidas accidentales de la información, pero nunca se ha sabido de intentos de acceso no autorizado.
- Actualmente se encuentran en desarrollo proyectos para cambiar todos los sistemas de cómputo administrativo de la UNAM a computadoras con sistema operativo Unix. Se estima que los sistemas Unisys serán utilizados todavía por alrededor de cuatro años, pero después serán completamente sustituidos por sistemas abiertos.

### 3.2.3 Computadoras personales (PCs)

Las computadoras personales son, por mucho, los sistemas de cómputo más abundantes en la Universidad, y prácticamente en cualquier lugar del mundo. Esto se debe a varias razones:

1. Son relativamente baratas.
2. Son relativamente fáciles de usar.
3. Los sistemas operativos que utilizan (principalmente MS-DOS y Windows han adquirido muchísima popularidad a nivel mundial.

Desgraciadamente, como se verá posteriormente, esta amplia utilización de las computadoras personales tiene un serio impacto en la seguridad de una red de cómputo como REDUNAM.

### 3.3 Situación de la red de supercómputo dentro de REDUNAM

La supercomputadora Cray Y-MP4/464 de la UNAM, ubicada físicamente en la DGSCA, está conectada a REDUNAM, como tantos otros sistemas de cómputo en la Universidad. Esto no solo es importante, sino necesario, pues la única forma que tienen los usuarios de utilizar la supercomputadora es conectándose a ella a través de la red para establecer sesiones remotas y llevar a cabo su trabajo.

La conexión de la supercomputadora a REDUNAM es peculiar en cuanto a que está conectada directamente a la red, sin un *firewall* de por medio. Un *firewall* es un sistema que filtra la información que entra y sale de una subred hacia Internet, y su uso es recomendado en aplicaciones de alta seguridad ([GS92, cap. 14], [RG92, pág. 203], [FS93, pág. 164], [CMQ92], [Che91], [Bel92]).

Prácticamente todas las supercomputadoras utilizadas en instituciones de investigación y empresas en todo el mundo se conectan a Internet a través de un *firewall*, para incrementar la seguridad del sistema.

En la UNAM, sin embargo, se decidió no utilizar dicha barrera por las siguientes razones:

1. Se desea que el acceso a la supercomputadora sea lo más fácil posible para los usuarios.
2. Al ser un sistema exclusivamente de investigación académica, se considera que el nivel de seguridad que se consiga sin un *firewall* puede cubrir los requerimientos previstos.
3. El costo de instalar y mantener un *firewall* es elevado.
4. A nivel mundial, pocas supercomputadoras pertenecientes a universidades utilizan sistemas de *firewall*.

Aunque la ausencia de un *firewall* hace más fácil la vida de los usuarios de la Cray, la hace más difícil para el equipo de administración y seguridad del sistema, pues así como es más fácil el acceso para los usuarios autorizados, es más fácil el acceso para quienes desean hacer uso no autorizado de la supercomputadora.

Dentro de DGSCA, la Cray está conectada a dos ruteadores marca NSC (*castor* y *pollux*), que le proporcionan su conexión “al mundo exterior” (figura 3.1). Directamente a los ruteadores se conecta un anillo de FDDI a través del cual la supercomputadora se comunica a los sistemas del Laboratorio de Visualización de la DGSCA, que son utilizados por los usuarios de la Cray que desean realizar visualización científica de los resultados obtenidos. La utilización de FDDI se debe a que la transferencia de datos es mucho más rápida que en los medios tradicionales (como Ethernet), y la velocidad es un factor determinante para poder realizar un despliegue gráfico adecuado.

Sin embargo, las máquinas del Laboratorio de Visualización también cuentan con una red Ethernet estándar (de cable coaxial) cuya velocidad de transferencia de datos es mucho menor que la de FDDI, pero que proporciona redundancia en caso de que exista algún problema con el anillo de FDDI.

Por otro lado, también interactúan con la supercomputadora las estaciones de trabajo utilizadas por el personal de los Departamentos de Supercómputo y Visualización. Estas

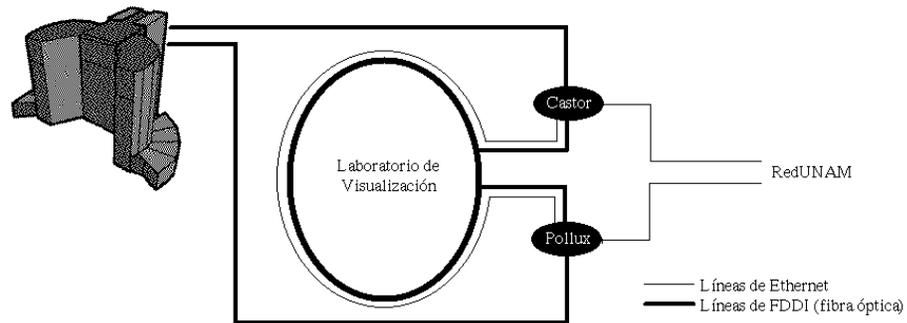


Figura 3.1: Red de supercómputo.

máquinas se conectan a los ruteadores NSC, y de ahí a la supercomputadora Cray, a través de REDUNAM, utilizando las redes Ethernet estándar de DGSCA.

### 3.4 Sistemas de cómputo involucrados en el proyecto

En este proyecto se trabajó de forma directa con los sistemas del Departamento de Supercómputo, el Laboratorio de Visualización y la supercomputadora Cray. Algunos de estos sistemas cumplen funciones específicas dentro del proyecto, mientras que otros son de propósito general. A continuación se describe cada una de estas máquinas, especificando sus características principales y el papel que juegan dentro del proyecto de seguridad. Las secciones siguientes agrupan a los sistemas de acuerdo a su ubicación física.

Dentro de Internet, a cada máquina se le asigna un nombre. Este nombre es arbitrario, y es con el que se conoce a la máquina en la red (para establecer comunicación con ella) y en el trato cotidiano, para hacer referencia a algún sistema en particular. Dentro de las descripciones siguientes, a cada máquina se hace referencia por dicho nombre.

#### 3.4.1 Sala de la Supercomputadora Cray

La comúnmente llamada "Sala de Cray" es un área de la DGSCA acondicionada especialmente para dar cabida a la supercomputadora Cray Y-MP4/464 de la UNAM. Este sistema fue el primer objetivo del proyecto de seguridad, pues al tratarse de un equipo tan grande y poderoso, es un objetivo demasiado atractivo para los *crackers*.

**Marca:** Cray Research, Inc..

**Modelo:** Y-MP4/464.

**Sistema operativo:** UNICOS 8.0.

**Uso:** Investigación científica.

### 3.4.2 Departamento de Supercómputo

La tabla 3.3 lista las estaciones de trabajo pertenecientes a este Departamento.

Nombre	Marca	Modelo	Sistema Operativo	Uso
<i>ds5000</i>	DEC	DECstation 5000/200	Ultrix 4.4	Coordinación del proyecto de seguridad, prueba e instalación inicial de herramientas y técnicas de seguridad, concentración de información de seguridad, suministro de ciertos servicios de red.
<i>tequila</i>	Sun	SparcClassic	Solaris 2.3	Utilización del personal del Departamento de Supercómputo.
<i>tepache</i>	Sun	SparcClassic	Solaris 2.3	
<i>xtabentun</i>	Sun	SparcClassic	Solaris 2.3	
<i>pulque</i>	SGI	Indy	Irix 5.2	
<i>mezcal</i>	Sun	SparcClassic	SunOS 4.1.3	Suministro de servicios de red (FTP, WWW, Listas de correo electrónico, etc.)

Tabla 3.3: Estaciones de trabajo del Departamento de Supercómputo.

### 3.4.3 Laboratorio de Visualización

La tabla 3.4 lista los sistemas pertenecientes a esta área.

Nombre	Marca	Modelo	Sistema Operativo	Uso
<i>polaris</i>	SGI	4D/420VGX	Irix 5.2	Acceso para los usuarios de la supercomputadora Cray y de personal del Laboratorio de Visualización.
<i>mira</i>	SGI	4D/35	Irix 5.3	
<i>capella</i>	SGI	4D/35	Irix 4.0.5	
<i>andromeda</i>	Sun	Sparc 1+	SunOS 4.1.2	
<i>nocdos</i>	HP	9000/730	HP-UX A.B8.05	
<i>cygnus</i>	Sun	SparcClassic	Solaris 2.3	Uso del personal del Laboratorio de Visualización.
<i>altair</i>	Sun	SparcClassic	Solaris 2.3	
<i>pegasus</i>	Sun	SparcClassic	Solaris 2.3	
<i>casiopea</i>	Sun	SparcClassic	Solaris 2.3	
<i>aldebaran</i>	SGI	Indy	Irix 5.2	
<i>deneb</i>	SGI	Indigo 2	Irix 5.3	Suministro de servicios de NIS y NFS a otras máquinas de los departamentos de Supercómputo y Visualización.
<i>diphda</i>	Sun	SparcServer 1000	Solaris 2.3	

Tabla 3.4: Estaciones de trabajo del Laboratorio de Visualización.

## Capítulo 4

# Problemática en la UNAM

Este capítulo define algunos de los problemas más importantes para la seguridad en cómputo en la UNAM.

### 4.1 La confianza

Es un hecho que la gran mayoría de los problemas de seguridad se producen a través de la red. Este problema se debe en gran medida a que muchos sistemas de cómputo automáticamente (o por haber sido configurados así por sus usuarios) “confían” en otras máquinas que estén en su misma red local. En este contexto, “confiar” significa permitirle a dicha máquina acceso a los recursos locales sin realizar las verificaciones rutinarias completas de seguridad y control de acceso. Dependiendo de la configuración de cada sistema, su percepción de “red local” puede ser la de la subred en la que está conectado, la de todo REDUNAM, o la de todo Internet.

#### **Ejemplo:**

Supóngase que la máquina *B* confía en la máquina *A*, de manera que le permite a los usuarios de *A* establecer sesiones interactivas en *B* sin necesidad de proporcionar sus *passwords*. Si un intruso logra acceder a una cuenta en *A* de forma no autorizada, automáticamente tendrá acceso también a una clave de *B*. Si a su vez existen otras máquinas que confían en *B*, el problema se sigue extendiendo, y puede llegar a causar violaciones graves a la seguridad de los sistemas de toda una organización. La figura 4.1 ilustra este concepto.

En resumen, las redes de computadoras pueden hacer que problemas pequeños de seguridad se extiendan hasta alcanzar proporciones estratosféricas.

### 4.2 El cómputo administrativo

El cómputo administrativo en la UNAM, actualmente, se puede resumir en una sola palabra: Unisys. Sin embargo, se tienen en este ámbito los siguientes problemas:

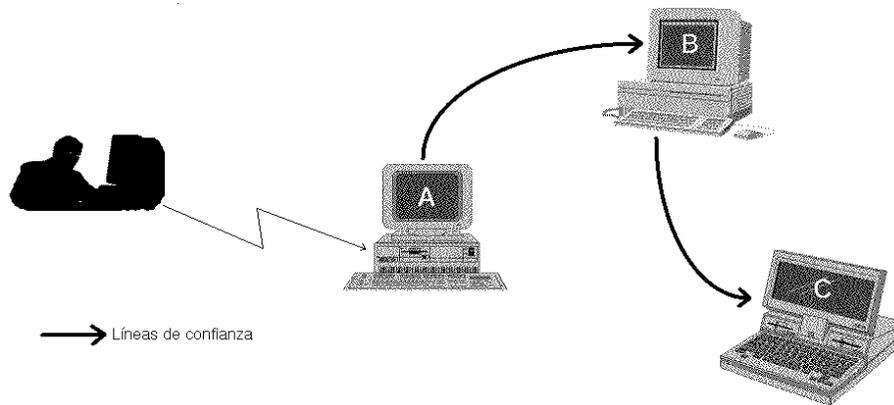


Figura 4.1: Cómo la confianza puede ocasionar que los problemas de seguridad se extiendan.

- La seguridad de los sistemas de cómputo administrativo Unisys usados en la UNAM se basa principalmente en los siguientes elementos:
  1. El acceso al sistema a través de la red está restringido a ciertas máquinas. En este aspecto, si la seguridad de dichos sistemas se ve afectada, la seguridad del sistema Unisys también estará en peligro.
  2. El sistema operativo de Unisys es muy poco conocido. En este aspecto, la seguridad se basa en que es poca la gente que sabrá “qué hacer” en caso de obtener acceso no autorizado al sistema.
- El cambio de los sistemas administrativos a Unix se está llevando ya a cabo sin poner demasiada atención a la seguridad de dichos sistemas.
- El esquema de seguridad utilizado en los sistemas Unisys se puede considerar en gran medida como seguridad por oscuridad, es decir, basado en la falta de información sobre el sistema y en la imposibilidad de conectarse fácilmente a él. Sin embargo, no se toma ninguna medida explícita para controlar y monitorear su seguridad.

Aunque la instrumentación de esquemas de seguridad en el cómputo administrativo de la UNAM esté fuera del alcance de esta tesis, el hecho de que dichos sistemas de cómputo vayan a convertirse a Unix próximamente indica que este proyecto puede sentar antecedentes útiles para quien tenga en sus manos la conversión del cómputo administrativo de Unisys a Unix.

### 4.3 Computadoras personales (PCs)

Las computadoras personales son la peor pesadilla de la seguridad en cómputo, por las siguientes razones:

1. Los sistemas operativos utilizados normalmente (MS-DOS y Windows) no proporcionan ningún control de acceso o división de privilegios. Quienquiera que esté al teclado tiene control absoluto del sistema.
2. Los sistemas de red utilizados (NetWare, LAN Manager, etc.) proporcionan mecanismos de seguridad muy rudimentarios y fácilmente eludibles.
3. Abundan.

Para todos los efectos prácticos, una PC se puede considerar como un sistema Unix en el que el usuario siempre es **root**. Por ejemplo, es muy fácil en una PC monitorear los paquetes de red que pasan por ella, debido a que cualquier usuario puede leer los puertos apropiados en las interfaces de red.

Desgraciadamente, hay poco que se pueda hacer, fuera de proteger a los demás sistemas contra cualquier tipo de acceso desde una PC. Algunas herramientas de seguridad permiten proporcionar este tipo de control en un sistema Unix. Mientras no se popularice la utilización en PCs de sistemas operativos que proporcionen mecanismos de seguridad (por ejemplo, versiones de Unix para PC y hasta cierto punto Windows NT), las PCs existentes en la red seguirán siendo puntos abiertos de acceso y problemas potenciales de seguridad.

### 4.4 Incidentes previos de seguridad en la UNAM

Los incidentes en los que se viola la seguridad de un sistema de cómputo suceden más frecuentemente de lo que normalmente se piensa. En muchas ocasiones, los administradores de las máquinas afectadas ni siquiera se dan cuenta de lo que ha sucedido, debido a que no se tiene la vigilancia adecuada.

Los sistemas de supercómputo son objetivos particularmente atractivos de este tipo de ataques. Esto se debe principalmente a dos razones:

1. Para quien pretende hacer uso ilegal de la máquina, ofrece una cantidad enorme de recursos de cómputo.
2. Para quien lo hace solo como pasatiempo, el “orgullo” de decir: “yo entré a la supercomputadora...”.

Durante el desarrollo de este proyecto se ha tenido noticia de incidentes de seguridad en diversos centros de supercómputo, algunos de los cuales involucraron también directamente a la UNAM. Estos ataques van desde los intentos fallidos hasta los que tienen éxito en entrar al sistema y, en algunas ocasiones, obtener privilegios de administración (clave del superusuario o de **root**). Éste es el objetivo último de todo atacante, pues una vez consiguiéndolo, puede hacer prácticamente lo que quiera con el sistema.

Desafortunadamente, la supercomputadora Cray de la UNAM ya sufrió uno de estos incidentes, en su nivel mayor de gravedad, es decir, obtención de privilegios de superusuario por parte del intruso. Es necesario reconocer que fue este incidente el que efectivamente “disparó” el proyecto de seguridad. A continuación se describe este incidente.

El 10 de Julio de 1993, la Lic. Martha A. Sánchez Cerezo, jefe del Departamento de Supercómputo, descubrió en la supercomputadora una cuenta llamada **god**, asignada a un usuario no existente. La falsedad de esta cuenta era evidente desde su nombre (*god* es “Dios” en inglés).

Al hacer una revisión detallada de la cuenta, se vio que pertenecía a todos los grupos y tenía activados todos los permisos, situación en la que ningún usuario de la Cray (ni siquiera **root**) se encuentra. La creación de la cuenta no aparecía en ninguno de los archivos de bitácora del sistema, y dentro de su directorio *home* existía un programa perteneciente a **root**, y que era una versión modificada del comando **su** de Unix. Este programa le permitía al dueño de la cuenta **god** entrar a **root** sin necesidad de proporcionar ningún *password*, y sin registrar dicha acción en ningún archivo de bitácora del sistema. Este programa es un ejemplo típico de una “puerta trasera” (*backdoors*).

Mientras todavía se estaban revisando estos hechos, se observó la entrada al sistema del usuario **jlm**, perteneciente al personal del Laboratorio de Visualización, pero desde la computadora *roxanne*, situada en el Instituto de Ciencias Nucleares de la UNAM. Se intentó comunicación con este usuario por medio del comando **writeln**, pero no hubo respuesta y el usuario se desconectó en ese momento. Inmediatamente se llamó por teléfono al dueño de la cuenta **jlm**, comprobándose que no había sido él quien había entrado momentos antes.

#### 4.4.1 Respuesta al incidente

Como medidas inmediatas ante este descubrimiento se realizó lo siguiente:

1. Se canceló la cuenta **god**, cambiándole el *password*.
2. Se notificó de los hechos al Director de Cómputo para la Investigación (en ese entonces, el Dr. Enrique Daltauit Godás).
3. Se decidió suspender el servicio de la supercomputadora para poder realizar una revisión exhaustiva sin que hubiera usuarios en el sistema.
4. Se suspendió también el servicio del Laboratorio de Visualización, pues se detectó que también había sido comprometida la seguridad de sistemas de este departamento, así como del Departamento de Supercómputo.
5. Se solicitó la presencia de un experto en seguridad de la empresa Cray Research, Inc. (Bryan Koch) para realizar un análisis detallado de la situación presentada. Este analista estuvo presente durante todas las acciones que se tomaron.
6. Se notificó al CERT [CER90] de lo sucedido, informando solamente de lo ocurrido en las estaciones de trabajo (no en la Cray), a petición de Cray Research,

Inc. El CERT solicita que se le notifique de cualquier incidente de seguridad con los siguientes objetivos:

- (a) Establecer, cuando sea posible, patrones de acción que puedan indicar un incidente ocasionado por el mismo atacante en otros sitios.
  - (b) Proporcionar asesoría en el manejo del incidente.
  - (c) Alertar a otros sitios sobre ataques del mismo tipo que puedan realizarse.
7. Se revisó el sistema completo, y se encontraron tres copias del programa encontrado en el directorio *home* de **god**. Todas ellas estaban en sitios “ocultos”, y con nombres que no indicaban de ninguna manera su función real.
  8. Se realizaron respaldos completos del sistema, para su revisión.
  9. Se reinstaló el sistema operativo UNICOS 6.1, con el fin de contar con un sistema limpio (sin puertas traseras adicionales).
  10. Se deshabilitaron todas las claves del sistema, requiriéndose la solicitud personal del dueño de la clave para su reactivación.
  11. Se instalaron inmediatamente los programas COPS y Crack.
  12. Se procedió a identificar a los responsables del ataque, que resultaron ser estudiantes de la Facultad de Ciencias de la UNAM.

#### 4.4.2 Las causas y los efectos

Una vez “calmadas las aguas” se procedió a realizar un análisis detallado del problema. El principal responsable del ataque accedió a cooperar con las autoridades de Cray Research, Inc., explicando los mecanismos mediante los cuales se obtuvo el acceso a la supercomputadora.

De esto resultó que, como en la gran mayoría de los casos, el éxito del ataque se debió principalmente a un error humano. El atacante había previamente instalado un *sniffer* (un programa que permite capturar los paquetes de red que salen y llegan a un sistema) en una máquina de la División de Estudios de Posgrado de la Facultad de Ingeniería (DEPFI). Cuando en una ocasión una persona autorizada del Departamento de Supercómputo utilizó la clave de **root** de la Cray desde esa máquina, el *sniffer* capturó el *password*, y le permitió al atacante entrar posteriormente a dicha clave. Una vez obtenido este privilegio, procedió, el 29 de Junio de 1993 a la creación de la cuenta **god** y del programa encontrado dentro de la misma. Así, el atacante solo tuvo que utilizar el *password* de **root** una vez, y las veces siguientes le bastaba con entrar a la cuenta que había creado y ejecutar el programa que le concedía todos los privilegios.

Después de este incidente, Cray Research, Inc. añadió un nuevo punto a la cláusula de seguridad del sistema del contrato con la UNAM, que sigue vigente hasta la fecha: la clave de **root** solamente puede ser utilizada desde la consola del sistema, para evitar

que el *password* de esta cuenta viaje a través de la red y se vea expuesto a interceptación por personas no autorizadas.

Por considerarse como un problema grave de seguridad en cómputo, Cray Research, Inc. informó de los hechos al Agregado Científico de la Embajada de los Estados Unidos en México.

Dada la falta de legislación universitaria sobre seguridad en cómputo, no fue posible ejercer ninguna acción formal contra los responsables, salvo una llamada de atención.

## Capítulo 5

# Recursos disponibles

Uno de los principales problemas al implementar un esquema de seguridad a nivel amplio es la indiferencia o ignorancia de la gran mayoría de la gente. Russell L. Brand dice en *Coping with the Threat of Computer Security Incidents—A Primer from Prevention through Recovery* [Bra90, página 5]:

“Históricamente se ha hecho muy poco para prevenir incidentes de seguridad en cómputo, y algunos de los más renombrados científicos en computación del país me han dicho que ‘La Seguridad en Cómputo es una pérdida de tiempo’.”

Este párrafo es representativo del problema al que se enfrenta cualquier equipo que pretenda realizar una labor profunda sobre seguridad en cómputo. Por esto es necesario e importante hacer uso de todos los recursos disponibles. A continuación se describen los recursos con los que se cuenta en la UNAM, y algunos con los que aún no se cuenta, pero que se espera contar en un futuro no muy lejano.

### 5.1 Recursos humanos

Como ya ha sido dicho en varias ocasiones, el principal elemento que hace seguro o inseguro un sistema de cómputo es la gente que lo utiliza. Todos los mecanismos técnicos y legales que se implementen son de poca utilidad si los usuarios y administradores involucrados en el sistema no toman conciencia de su papel y de la importante labor que ellos tienen que desempeñar para incrementar la seguridad de su información.

#### 5.1.1 Administradores

Se utiliza genéricamente el término “administrador”, cuando se habla de sistemas de cómputo, para referirse a la persona encargada de mantener en todos sus aspectos el sistema. El administrador es quien supervisa el correcto funcionamiento de la computadora en cuanto a *software* y *hardware*. Si existe cualquier problema en cuanto al

primero, muy posiblemente sea el administrador quien tenga que resolverlo personalmente. Si el problema es en *hardware*, el administrador es quien debe tomar nota del mismo para coordinar a las personas encargadas del soporte técnico del equipo y lograr la eliminación del problema lo más pronto posible.

En máquinas grandes, la administración del sistema normalmente está a cargo de varias personas, cada una encargada de un aspecto específico del mismo. Sin embargo, a medida que avanza la tecnología y los equipos se hacen más pequeños y poderosos (por ejemplo, estaciones de trabajo y computadoras personales), las tareas de administración se concentran cada vez más en una sola persona. Cada vez es más común, incluso, que una sola persona tenga que encargarse de la administración de varios sistemas.

Para incrementar aún más la carga de trabajo del administrador, él es también el principal responsable del monitoreo y control de seguridad del sistema. Como administrador, es quien tiene la posibilidad y la responsabilidad de establecer técnicas y políticas adecuadas para asegurar un razonable grado de seguridad, de acuerdo a las necesidades de su sitio de trabajo y de los usuarios.

Un administrador tiene que estar capacitado en todos los aspectos de su labor, no solamente en seguridad. Todos los puntos de la administración de Unix, aunque no parezcan estar directamente relacionados con la seguridad, pueden llegar a afectarla. Las interacciones entre los diferentes subsistemas que componen a Unix pueden llegar a ser muy complejas, y como bien dice el refrán: *“Donde menos se espera salta la liebre”*.

Los administradores cuentan con una serie de recursos que permiten aligerar un poco esta responsabilidad mediante la automatización de aspectos técnicos y la regulación de los aspectos organizacionales, humanos y legales involucrados en la seguridad de los sistemas de cómputo.

Es importante dejar muy claro que el administrador del sistema es el elemento individualmente más importante en la seguridad del mismo. Ninguna herramienta puede reemplazar a un administrador humano que esté alerta y conciente de su responsabilidad hacia los usuarios y hacia el mismo sistema. Es posible automatizar la generación y, hasta cierto punto, el análisis de la información. Sin embargo, existen muchos detalles que requieren de una interpretación humana para adquirir su verdadero significado. Hay muchas cosas que para un programa pueden ser normales, pero que a un administrador que sepa qué está pasando en su sistema le pueden dar una pista a seguir para averiguar si algo no está funcionando bien.

Por otro lado, el administrador es la pieza clave para difundir la cultura de seguridad, tanto a los usuarios como a las personas encargadas de tomar decisiones de amplio alcance en una organización.

### 5.1.2 Usuarios

Los usuarios son, en última instancia, quienes “hacen y deshacen” el funcionamiento de un sistema de cómputo. Son los usuarios quienes, en la práctica, dan su aprobación o su rechazo a las políticas, mecanismos y servicios establecidos por los administradores. Esto, obviamente, incluye los aspectos de seguridad. Si el administrador impone restricciones de seguridad que los usuarios consideren inapropiadas, tendrá necesariamente que ofrecer justificaciones o alternativas adecuadas.

Por otro lado, dentro de los límites determinados por el administrador, los usuarios pueden hacer lo que quieran, y es aquí donde importa que los usuarios estén conscientes de su papel dentro de la seguridad del sistema. De poco sirven los mecanismos de control de acceso que establecidos sí, por ejemplo, los usuarios dejan sus cuentas sin *password* que las proteja, o lo escriben en una pieza de papel que dejan sobre su escritorio.

Así pues, es muy importante que los usuarios tengan conciencia de la importancia de la seguridad en cómputo. Casi siempre es el administrador quien debe inculcar esta actitud en los usuarios. Dependiendo del tipo de organización en la que se trabaje, esta labor puede resultar sencilla o casi imposible. En un ambiente militar, por ejemplo, donde la disciplina es la norma, los usuarios usualmente tienen conciencia (o al menos la orden) de cuidar la seguridad del sistema y seguir todas las reglas y políticas establecidas. En un ambiente universitario, como el que cubre el presente trabajo, esta labor se complica bastante debido a la diversidad de personas, intereses, habilidades y necesidades que entran en juego. No para todos los investigadores y estudiantes es clara la necesidad de poner un buen *password* en su cuenta, por ejemplo.

### 5.1.3 Personal directivo

En toda organización existe un cuerpo directivo, es decir, un grupo de personas que se encargan de tomar las decisiones de alto nivel basándose en las políticas y necesidades de la organización. Ya sea un Departamento de Defensa o una Universidad, siempre existirá una o varias personas que tienen que autorizar la ejecución de las medidas de amplio alcance que se tomen.

Si consideramos que la seguridad en cómputo, para que sea realmente efectiva, tiene que ser una decisión institucional que llegue a todos los rincones de la organización, resulta obvio que es muy importante convencer también a los niveles directivos de la importancia de contar con políticas y mecanismos sólidos de seguridad. Pocas cosas son más saludables para un programa de seguridad en cómputo que contar con el apoyo de la directiva. Desde organizar pláticas y seminarios hasta adquirir nuevo equipo de cómputo, pasando por contar con personal especializado y recibir la capacitación adecuada, todas las actividades son mucho más sencillas cuando el cuerpo directivo está consciente de la necesidad de la seguridad en cómputo para resolver problemas reales, y no solamente como un “capricho tecnológico” de los administradores de los sistemas.

Casi siempre, una vez más, es labor del administrador el convencer al personal directivo de esta necesidad. Los administradores y los usuarios son el aspecto “técnico” de la organización, y pueden experimentar los problemas y las necesidades en carne propia. El nivel directivo, por el contrario, no siempre está en contacto directo con las máquinas, y por lo tanto con los problemas. Por esta razón, es necesario convencerlos con argumentos sólidos de la importancia de otorgar su apoyo a la seguridad.

Para la puesta en marcha y realización de este proyecto de tesis se contó con el valioso apoyo del cuerpo directivo de la DGSCA (Dirección General de Servicios de Cómputo Académico) de la UNAM, y principalmente con el del personal del Departamento de Supercómputo y el Laboratorio de Visualización de esta dependencia.

## 5.2 Recursos técnicos

Una vez mencionados los aspectos humanos involucrados en un programa de seguridad en cómputo, es vital también hacer mención de los recursos técnicos con los que se cuenta. Al fin y al cabo, la seguridad en cómputo se encarga de computadoras, y la principal función de las computadoras es relevar al hombre de trabajos repetitivos, monótonos y constantes.

Por esta razón, a pesar de que el humano es aún indispensable para tomar las decisiones, las computadoras mismas son la herramienta más valiosa para el monitoreo y control de la seguridad en cómputo. Una computadora puede ser un vigilante incansable e infalible, aunque no demasiado listo. Aún así, puede proporcionar información valiosa que le sirva a un ser humano para realizar un análisis y tomar las decisiones que se crean convenientes.

### 5.2.1 Herramientas comerciales

Este es un aspecto muy poco discutido en este trabajo, debido a una razón principal: las herramientas comerciales cuestan dinero, y el dinero no siempre es un recurso ampliamente disponible en la UNAM. Además, las herramientas comerciales de seguridad en cómputo se encuentran en un estado de desarrollo bastante incipiente, y existen herramientas de dominio público para cubrir prácticamente todas las necesidades de seguridad de una institución académica como la UNAM.

### 5.2.2 Herramientas de dominio público

El crecimiento de Internet ha promovido el desarrollo y distribución de programas de cómputo de todo tipo, que a través de esta red internacional pueden ser fácilmente obtenidos desde cualquier lugar del mundo.

Los programas relacionados con seguridad en cómputo no se han quedado atrás. Existen actualmente en Internet una gran cantidad de herramientas disponibles de forma gratuita y que permiten monitorear y controlar todos los aspectos principales de la seguridad de una computadora, algunos de los cuales son:

1. Monitoreo y control de acceso de los usuarios.
2. Monitoreo y control de acceso a servicios.
3. Control de calidad de *passwords* de los usuarios.
4. Verificación de integridad del sistema.
5. Verificación de aspectos clave en la seguridad de un sistema.

Estas herramientas fueron las elegidas como soporte técnico debido a su amplia disponibilidad y a que todas incluyen el código fuente, lo que las hace altamente portables a diferentes sistemas de cómputo. Además, su amplia utilización las hace altamente confiables.

### 5.2.3 Herramientas desarrolladas en la UNAM

En ciertas ocasiones, las herramientas de dominio público no son suficientes para satisfacer las necesidades de una organización como la UNAM.

En muchos institutos y dependencias de la UNAM existen casi exclusivamente usuarios de las computadoras, y no hay personal especializado en computación que pueda encargarse de los aspectos técnicos de la seguridad. El problema que esto presenta es que la gran mayoría de las herramientas de seguridad de dominio público requieren de un cierto nivel de conocimiento sobre administración de Unix. Por esta razón, en muchas ocasiones es necesaria la existencia de herramientas que permitan “aislar” al administrador, que puede incluso no saber ni estar interesado demasiado en la seguridad, de los detalles técnicos, y presentarle la información necesaria de una forma clara y precisa, que le permita tomar decisiones bien informadas sin necesidad de ser un experto en seguridad.

Dentro del proyecto de seguridad también se atacó este problema mediante el desarrollo de algunas herramientas enfocadas específicamente a la solución del problema recién mencionado.

### 5.2.4 Herramientas de difusión de la información

Aunque gran parte de este proyecto consta de proporcionar soluciones tecnológicas al problema de la seguridad en cómputo, es un hecho que la seguridad es un problema cultural además de técnico. Es un problema humano, y no solamente de máquinas. Por esta razón, la comunicación entre seres humanos es tan importante como el control y generación de información entre las máquinas.

Afortunadamente, la misma tecnología que se intenta proteger ofrece medios magníficos de comunicación y difusión de la información. A continuación se describen los medios más utilizados en este proyecto.

#### Listas de correo electrónico

El correo electrónico se ha convertido en un medio de comunicación casi indispensable en ciertos medios. En los sistemas conectados a Internet, permite comunicarse de forma casi instantánea a personas que pueden estar físicamente localizadas en puntos opuestos del planeta. Es un medio de comunicación que presenta todas las ventajas del correo convencional, pero con las ventajas adicionales de una conversación en persona: mientras que una carta “normal” puede tardar semanas en llegar a su destino, el tiempo de transmisión de un mensaje electrónico es normalmente de segundos o minutos.

Una lista de correo electrónico es un medio de distribución de mensajes a muchas personas de forma automática. Cuenta con una dirección electrónica y una lista de suscriptores. Cualquiera de los suscriptores puede enviar un mensaje a la lista, y dicho mensaje es automáticamente distribuido a todas las personas suscritas a ella.

En un grupo de personas en el que la gran mayoría de los integrantes tienen acceso a correo electrónico, una lista de este tipo puede servir como un excelente medio de comunicación, con las siguientes ventajas:

- No es necesario convocar a las personas a reuniones “físicas” para dar a conocer alguna información, proponer ideas, realizar consultas, etc.
- El grupo puede mantenerse continuamente en contacto, y se abre la posibilidad de discutir un mismo tema a lo largo de días o incluso semanas, con resultados frecuentemente más útiles que los obtenibles en una junta “normal” de dos o tres horas.
- La información que se distribuye en la lista llega a todos los integrantes del grupo, y no solamente a quienes acudan a las reuniones.
- La distribución de la información es inmediata.
- El mismo sistema de manejo de la lista lleva automáticamente un registro de lo que se dice en ella.

Por estas razones, las listas de correo electrónico juegan un papel determinante en la realización de este proyecto. Véase el apéndice A para mayor información técnica sobre los programas utilizados para mantener estas listas.

### **FTP anónimo**

FTP (*File Transfer Protocol*) es un servicio estándar de Internet que permite realizar transferencia de archivos entre diferentes computadoras a través de la red. Normalmente, esto permite a los usuarios transferir archivos entre diferentes máquinas en las que tengan cuentas.

Sin embargo, también se puede utilizar para proporcionar un servicio conocido como FTP anónimo, en el que una máquina permite acceso a través de FTP a cualquier persona en Internet. Esto es una de las formas más comunes de difusión de información en esta red. Al permitir poner a disposición del mundo ciertos archivos, el servicio de FTP anónimo se utiliza para distribuir programas, documentos, imágenes, y cualquier clase de información.

En este proyecto, se hace uso extensivo de FTP anónimo para difundir principalmente:

- Documentos sobre diversos aspectos de seguridad en cómputo.
- Herramientas de seguridad de dominio público.
- Información sobre los eventos realizados o por realizarse.

En el apéndice B se describen los aspectos técnicos del servidor de FTP utilizado.

### **World Wide Web**

*World Wide Web* (WWW o W3) es un sistema de información distribuida en Internet utilizando técnicas de hipertexto y multimedia, lo que permite tener acceso muy fácil a gran cantidad de información, incluyendo imágenes, sonidos, animaciones, archivos, documentos en diferentes formatos, etc. WWW está rápidamente convirtiéndose en el

medio más utilizado para distribución de todo tipo de información, absorbiendo incluso a servicios muy conocidos como GOPHER y FTP anónimo.

Para la difusión de información en WWW se utiliza un protocolo conocido como HTTP (*HyperText Transfer Protocol*), que fue desarrollado exclusivamente para su utilización en WWW. Los documentos de WWW se elaboran en un lenguaje llamado HTML (*HyperText Markup Language*).

Para acceder a la información en WWW se hace uso de programas especiales conocidos como visualizadores de WWW. Los más conocidos de estos programas son Mosaic [fSAb], NetScape [Cor] y Lynx [oK].

Aparte de sus capacidades de difusión de información, WWW tiene una posibilidad que lo hace muy superior a otros protocolos: a través de WWW no solamente se puede difundir información, sino que también se tiene la capacidad de recopilarla. A través de estructuras configurables conocidas como “formas” de HTML, un documento puede recibir información de la persona que lo está consultando. De esta forma se hace posible la realización de encuestas interactivas, por ejemplo, sin necesidad de que alguien visite a las personas, y sin la inconveniencia de tener que hacerlas a través de correo electrónico.

La UNAM no se está quedando atrás en la utilización de este importante recurso. Cada vez más dependencias de la Universidad están implementando servidores de WWW para difundir su información.

Dentro de este proyecto, se utilizó también WWW para hacer disponible a la comunidad universitaria una fuente amplia y confiable de información sobre seguridad, así como para recopilar información utilizando las capacidades interactivas de HTML.

Para información técnica sobre el servidor de WWW utilizado, consultar el apéndice C.

### 5.3 Recursos organizacionales y legales

Se han discutido hasta el momento los recursos disponibles en el “nivel técnico” de la seguridad en cómputo. Este es el nivel en el que, en última instancia, se llevan a cabo las acciones necesarias para garantizar la seguridad de un sistema, y en el que se desarrolla principalmente este proyecto.

Sin embargo, las acciones para incrementar la seguridad, y sobre todo si se trata de incrementar la seguridad en Unix en toda la Universidad, no pueden quedarse en este nivel. Es necesario pasar a los niveles más altos, en los que se pueden tomar decisiones que afecten de forma profunda y duradera la percepción de la seguridad en cómputo. Este es el nivel en el que este proyecto puede convertirse en parte de la cultura computacional de la UNAM. Se trata del nivel organizacional y legal.

Aunque esta tesis no profundiza en este aspecto, sí es importante mencionar las acciones que se han realizado para promover la toma de decisiones que lleven a, principalmente:

1. El establecimiento de políticas que rijan la seguridad en cómputo a nivel de dependencias, o incluso a nivel universitario.

2. La formulación de legislación apropiadas, a nivel institucional, que contemplen los aspectos de seguridad en cómputo.

Otro aspecto que es importante mencionar dentro de los recursos organizacionales disponibles es la existencia de grupos internacionales dedicados al monitoreo, supervisión, control y acción sobre seguridad en cómputo. Durante el transcurso de este proyecto se trabajó siguiendo de cerca las acciones y recomendaciones de dichos organismos.

## **Parte II**

# **Acciones tomadas**



## Capítulo 6

# Acciones iniciales en DGSCA

Un refrán muy conocido dice “*El buen juez por su casa empieza*”. Otro más reza “*En casa del herrero, azadón de palo*”. Estas dos frases sirven muy bien para describir la situación que prevalecía en DGSCA en cuanto a seguridad en cómputo: siendo la dependencia de la UNAM encargada de coordinar el cómputo a nivel institucional, debía ser la primera en tomar acciones en cuanto a este renglón tan importante de la computación. Sin embargo, no se habían tomado nunca dichas acciones. La seguridad seguía siendo en DGSCA lo que es en tantas partes del mundo: un campo completamente teórico, del que a veces se habla y nunca se hace nada.

Al surgir el proyecto de seguridad que conforma esta tesis, se consideró que la primera medida a tomar era incrementar la seguridad de los sistemas Unix en la DGSCA. Aún cuando el proyecto pretende alcanzar a todas las áreas de la Universidad, se vio la conveniencia de “comenzar por la casa”, tomando medidas internas, que después podrían aplicarse en un contexto más amplio.

En este capítulo se describen dichas acciones.

### 6.1 Reinstalación de sistemas operativos

La primera acción a tomar fue reinstalar los sistemas operativos en todas las estaciones de trabajo de Supercómputo y Visualización, así como en la supercomputadora Cray Y-MP4/464.

#### 6.1.1 ¿Por qué reinstalar sistemas operativos?

Antes del inicio del proyecto de seguridad, y como es muy común en todos los sistemas Unix en cualquier parte del mundo, la atención prestada a la seguridad en cómputo era muy poca. Por lo tanto, existía una posibilidad muy alta de que durante algún tiempo se hubieran estado dando accesos no autorizados a los sistemas de DGSCA sin que estos accesos hubieran sido notados por los administradores.

### Puertas traseras (*backdoors*)

Se conoce como puerta trasera (*backdoor* en inglés) a un mecanismo que permite tener acceso a un sistema de cómputo sin tener que pasar por los mecanismos de verificación normales, como son proporcionar una clave y un *password* de acceso.

Una puerta trasera puede instrumentarse mediante alguna de las siguientes técnicas, entre otras:

1. Modificación de algún programa del sistema operativo (por ejemplo, **login**).
2. Adición de algún programa al sistema operativo (por ejemplo, un *sniffer* que permita monitorear las acciones de los usuarios).
3. Modificación de los archivos de configuración del sistema (por ejemplo, `/etc/passwd`).

En la historia de los incidentes de seguridad se ha observado que una de las primeras acciones que un intruso toma cuando logra obtener acceso a un sistema de cómputo es intentar establecer una puerta trasera que le permita posteriormente acceder al sistema sin problemas. De poco sirven las medidas de seguridad que se implementen en un sistema en el que ya estén instaladas puertas traseras, pues casi con toda seguridad estas puertas permitirán el acceso a los intrusos de cualquier manera.

Una puerta trasera es un problema importante de seguridad, y como tal, debe ser eliminada. Sin embargo, esto puede resultar prácticamente imposible, debido a que las modificaciones que se hayan realizado pueden estar “mezcladas” con el sistema operativo. Siendo Unix un sistema tan grande y complejo, que consta de cientos o miles de archivos y comandos, detectar dichas modificaciones es una labor titánica.

Por esta razón, la mejor forma de garantizar la inexistencia de puertas traseras en un sistema Unix es reinstalar el sistema operativo utilizando el medio original de distribución —normalmente cintas magnéticas, CD-ROM o diskettes—. Con esto se garantiza que todos los programas del sistema están en su forma original, sin puertas traseras de ninguna especie, a menos que alguna de ellas venga de fábrica (lo cual, por cierto, ha sido causa de muchos problemas de seguridad en la historia de Unix). No se puede garantizar lo mismo acerca de los archivos de configuración, pues muchas versiones de Unix, al momento de la instalación, conservan los existentes anteriormente. Sin embargo, la revisión “manual” de dichos archivos es mucho más sencilla que la revisión de los programas del sistema.

Así pues, reinstalar el sistema operativo es la forma más segura —aunque demasiado drástica en algunos casos— de tener un sistema “limpio” sobre el cual se puedan implementar otras medidas de seguridad.

### 6.1.2 Qué se hizo en DGSCA

Durante la segunda mitad de 1993, en la cual se inició “oficialmente” este proyecto, se reinstalaron los sistemas operativos de las siguientes máquinas:

**sirio:** En la supercomputadora se realizó la reinstalación del sistema operativo durante la semana del 19 al 23 de Julio de 1993. Esta reinstalación fue provocada por el

incidente de seguridad mencionado en la sección 4.4, y fue realizada por personal de la empresa Cray Research, Inc., en forma conjunta con personal del Departamento de Supercómputo de la UNAM, incluyendo al autor de esta tesis.

**ds5000:** En esta máquina se reinstaló el sistema operativo el día 27 de Julio de 1993.

**Laboratorio de Visualización:** En *polaris*, *capella*, *mira*, *andromeda* y *nocdos* se llevó a cabo la reinstalación de sistemas operativos durante la semana del 20 al 24 de Septiembre de 1993.

En estas fechas todavía no se contaba con los demás sistemas mencionados en la sección 3.4.

## 6.2 Instalación de herramientas de seguridad

Las herramientas de dominio público son uno de los recursos más importantes para mejorar la seguridad de los sistemas Unix. Uno de los objetivos de este proyecto es promover la utilización de dichas herramientas, con el fin de crear una infraestructura sólida y relativamente homogénea de generación de información y control de actividades, que permita monitorear de manera confiable la seguridad en los sistemas Unix de la UNAM.

Posteriormente se hablará de los esfuerzos por lograr la utilización de las herramientas principales en los sistemas Unix de la UNAM. Inicialmente, sin embargo, se llevaron a cabo actividades tendientes a la instalación y uso correcto de dichas herramientas en los sistemas Unix de Supercómputo y Visualización.

La utilización de herramientas de seguridad permite la generación constante de información, así como monitoreo y control de las actividades en los sistemas, tales como acceso a los servicios de red que ofrecen los mismos.

### 6.2.1 ¿Qué se necesita?

El primer paso en la utilización de herramientas de seguridad es determinar los requisitos de seguridad que se tienen. Existen herramientas que permiten implementar múltiples servicios y políticas de seguridad en casi todas las modalidades que se puedan necesitar. Es muy importante realizar un análisis cuidadoso de los requerimientos reales de seguridad antes de iniciar cualquier acción ([GS92, pp. 12–13,19], [RG92, pp. 89–93]). Después de realizar este análisis, se determinaron los siguientes puntos básicos para el esquema de seguridad en DGSCA:

1. Control de acceso de los usuarios a los servicios de red ofrecidos por el sistema, tales como **telnet**, **FTP**, **talk**, **finger**, **rlogin**, **rsh**, etc. La mayor parte de los problemas de seguridad se presentan a través de los servicios de red ([Bel92], [GS92, cáp. 11]), y por lo tanto es importante controlar quién utiliza dichos servicios y qué uso se les da.
2. Control de calidad de *passwords*(contraseñas de acceso) utilizadas por los usuarios. Los *passwords* son la primera línea de defensa en la seguridad de un sistema Unix ([GS92, pp. 24–36], [RG92, pp. 57–62], [Cur90, pág. 5], [Bra90, pág. 5],

[FS93, pág. 45]), y paradójicamente quizá, su mala selección y utilización es la fuente más común de problemas de seguridad, pues los usuarios tienden inevitablemente a utilizar *passwords* que son fácilmente adivinables [MT79, Kle92].

Por esta razón, es muy importante tomar medidas para asegurar la utilización de buenos *passwords* en los sistemas Unix. Existen múltiples medidas administrativas para lograrlo en diferentes grados ([GS92, pp. 36–43], [RG92, pp. 63–65], [FS93, pp. 58–70], [Cur90, pp. 7–8], [Bra90, pp. 5–6,8–12]). Uno de los esquemas que ha resultado más eficientes [GS92, pág. 43] es:

- (a) Realizar una revisión exhaustiva de los *passwords* ya existentes en el sistema.
  - (b) Implementar un mecanismo que impida a los usuarios elegir *passwords* débiles.
  - (c) Cuando sea apropiado (por ejemplo, para proteger claves muy importantes), implementar mecanismos para proteger los *passwords*, evitando su conocimiento público en la medida de lo posible. Esto implica, principalmente, evitar que los *passwords* puedan ser interceptados al viajar por la red.
3. Revisión cuidadosa de aspectos diversos de la seguridad del sistema, tales como permisos de archivos, privilegios de los usuarios, etc. El sistema operativo Unix es amplio y complejo, y en muchas ocasiones es difícil para un ser humano, ya sea el administrador o los usuarios, estar al tanto de todos los factores que pueden afectar la seguridad del sistema. Por lo tanto, es necesario contar con herramientas que realicen dicho tipo de revisiones de forma automática.
  4. Revisión de la integridad de los archivos del sistema. Una de las formas más comunes de introducir una puerta trasera que permita el acceso fácil a un sistema Unix es realizar modificaciones en los programas del mismo. Por lo tanto, es muy importante mantener un control que evite la realización de modificaciones no autorizadas en los archivos importantes del sistema operativo

## 6.2.2 ¿Qué herramientas?

La red Internet ofrece la ventaja de que es sumamente fácil distribuir programas. Cualquier persona que desarrolla un programa para su uso puede distribuirlo en Internet, y posiblemente lograr “la fama y el reconocimiento” de ver su programa utilizado por otras personas en el mundo.

Esto tiene ventajas y desventajas. Por un lado, en Internet es posible encontrar prácticamente cualquier tipo de programa. Sea cual sea la aplicación que se tenga en mente, es muy probable que alguien ya haya hecho algo semejante a lo que se necesita y lo haya hecho accesible en Internet.

Por otro lado, se promueve la proliferación de programas que en muchas ocasiones realizan funciones muy similares, y algunos de los cuales no siempre cumplen con las expectativas de calidad de los usuarios. Es necesario, entonces, seleccionar qué programas es conveniente usar, de acuerdo a las experiencias de otras personas que ya los

hayan utilizado, así como a nuestro propio análisis de las características, ventajas y desventajas de cada uno de ellos.

Una vez determinados los requerimientos de seguridad en Supercómputo y Visualización, fue necesario analizar las distintas herramientas disponibles en el dominio público que permiten cubrir cada uno de los puntos propuestos. A continuación se mencionan las herramientas consideradas en cada caso. Ver el apéndice D para una descripción completa de estas y otras herramientas de seguridad.

**Control de acceso a servicios** TCP-Wrapper, Xinetd.

**Control y protección de *passwords*** Crack, passwd+, npasswd, anlpasswd, S/Key, Kerberos, SECURE RPC, COPS.

**Revisión general de seguridad** COPS, Tiger, SATAN<sup>1</sup>

**Revisión de integridad de los archivos** TripWire, COPS.

En cada uno de estos casos se determinó la herramienta a utilizar en base a los siguientes factores:

1. Solidez y madurez del producto, tomando en cuenta recomendaciones de otras personas, tiempo de la herramienta “en el mercado”, reputación de los autores y/o instituciones encargadas, nivel de soporte proporcionado por los autores.
2. Facilidad de instalación y uso (nivel de transparencia para los usuarios).
3. Características específicas de la herramienta: configurabilidad, portabilidad, etc.

De acuerdo a este análisis, las herramientas seleccionadas para su instalación y uso en los sistemas de Supercómputo y Visualización en DGSCA fueron las siguientes:

**TCP-Wrapper** Es una herramienta en continuo desarrollo, y ampliamente recomendada por muchas personas e instituciones [QCM93]. Actualmente se encuentra ya en su versión 7.2, y es considerado como una herramienta de seguridad fundamental en cualquier sistema Unix.

**COPS** Un sistema con varios años de haber sido liberado, y utilizada en prácticamente todos los sitios que toman medidas por la seguridad. Se le considera una de las herramientas “obligatorias” de seguridad, y es muy ampliamente recomendado ([GS92, pp. 464–465], [FS93, pp. 249–256], [Cur90, pág. 38]). Cubre muchos aspectos (puede observarse que aparece en la lista de programas considerados para otras áreas de seguridad), pero no en todos es lo mejor que existe, por lo que casi siempre es recomendable combinarlo con otros programas especializados.

**Crack** Es el sistema de “rompimiento” de *passwords* más efectivo que existe, por la gran variedad de técnicas que utiliza para tratar de adivinar los *passwords* de los usuarios ([CHN<sup>+</sup>92], [FS93, pág. 257]).

---

<sup>1</sup> Esta herramienta fue liberada el 5 de abril de 1995, casi al final de esta tesis, por lo que no formó parte del análisis inicial. Sin embargo, se adoptó su uso inmediatamente debido a sus favorables características de revisión de la seguridad.

**passwd+** Es el más completo y configurable de los reemplazos del comando **passwd** de Unix. Cuenta con un archivo de configuración modificable por el administrador del sistema, de manera que las verificaciones realizadas sobre los *passwords* que los usuarios quieran utilizar pueden adaptarse a las necesidades específicas del sitio de trabajo, y hacerse tan flexibles o estrictas como se quiera. Aunque es un programa que sigue estando en versión alfa (una versión de prueba), es bastante recomendado ([FS93, pág. 257], [CHN<sup>+</sup>92, pág. 7], [Hun94, pág. 312]).

**S/Key** Es un sistema único, por su sencillez de funcionamiento y operación, su poca necesidad de recursos (funciona únicamente en base a *software*) y el alto nivel de protección que brinda. Permite proteger los *passwords* que viajan por la red, de manera que es muy útil para acceder a cuentas importantes del sistema (por ejemplo, la cuenta **root**) sin poner en peligro la confidencialidad de su *password*.

**TripWire** Es uno de los pocos sistemas existentes —quizá el único— que permite realizar un monitoreo estricto y confiable sobre la integridad de los archivos importantes del sistema. Tiene relativamente poco tiempo de haber sido liberado [KS94, pág. 1], pero ya es utilizado ampliamente en muchos sitios del mundo.

**SATAN** *Security Administrator Tool for Auditing Networks*, es una de las herramientas más nuevas y controversiales. Es la primera herramienta públicamente disponible que permite revisar, de forma remota, la seguridad en un sistema Unix, y encontrar posibles huecos en la misma. Es un recurso poderoso tanto para los administradores como para los intrusos potenciales, por lo que es muy importante su utilización oportuna.

### 6.2.3 Qué se hizo en DGSCA

Posteriormente a la decisión de las herramientas a utilizar se comenzó con el proceso de instalación de dichas herramientas en las máquinas de Supercómputo y Visualización. En estas áreas de DGSCA, cada estación de trabajo (y la Cray) es administrada por personas diferentes, por lo que fue necesario coordinar las actividades con los administradores de cada uno de dichos sistemas.

#### Privilegios de ejecución de las herramientas

No todas las herramientas utilizadas requieren de privilegios de **root** para su ejecución. De hecho, no se recomienda ejecutarlas con dichos privilegios para evitar abrir nuevos huecos de seguridad. Por lo tanto, se decidió crear en todas las máquinas comprendidas dentro de este proyecto una cuenta especial llamada **oss** (Oficial de Seguridad del Sistema) que fuera la dueña de todas las herramientas de seguridad instaladas. Se acordó la creación de dichas cuentas con los administradores de las máquinas, y el autor de este trabajo es la persona responsable de las mismas.

Para los programas (o partes de ellos) que requieren privilegios especiales para su ejecución o que requieren de modificaciones a archivos del sistema, se realizó la instalación en forma coordinada con los administradores.

### Qué herramientas en qué máquinas

Idealmente, todas las herramientas de seguridad deberían utilizarse en todas las máquinas. Sin embargo, esto no en todas las ocasiones es posible, debido a las diferentes características en las versiones de Unix utilizadas en los sistemas. Los casos especiales que se consideraron son:

**passwd+ en la Cray** El `passwd+`, por su función, necesita modificar directamente la base de datos de usuarios del sistema, normalmente almacenada en el archivo `/etc/passwd`. Sin embargo, la base de datos de usuarios en UNICOS (la versión de Unix utilizada en la Cray) no se almacena en dicho archivo, sino en la base de datos `uadb(5)`. Por lo tanto, el cambio de un *password* de usuario en la supercomputadora no se puede hacer de la misma forma que en un sistema Unix convencional. Para poder instalar `passwd+` en la Cray es necesario hacerle modificaciones al programa para que tenga en cuenta dichas diferencias.

**passwd+ en Solaris 2.3** En esta versión de sistema operativo, utilizado en casi todas las máquinas de marca Sun que se tienen, se presenta un problema similar al de la Cray. Este sistema operativo utiliza un mecanismo de *shadow passwords* que permite proteger los *password*s cifrados, almacenándolos en un archivo diferente a `/etc/passwd`. Por esta razón, `passwd+` no puede operar correctamente a menos de que se le hagan las modificaciones necesarias para tener en cuenta este factor.

**S/Key y TripWire en la Cray** Estos programas hacen uso de sistemas criptográficos para su funcionamiento. Desgraciadamente los programas que implementan estos sistemas están originalmente diseñados para funcionar en máquinas que trabajan internamente con números de 32 bits, que es el caso de casi todas las estaciones de trabajo. La Cray, sin embargo, utiliza números de 64 bits. Esto ocasiona que los programas puedan compilarse y utilizarse en la supercomputadora, pero generen resultados diferentes de los obtenidos en otros sistemas de cómputo.

En el caso del **S/Key**, esto lo hace prácticamente inutilizable, pues este programa interactúa forzosamente con versiones de sí mismo que están instaladas en otros sistemas. Al no generarse los mismos resultados en ambos, se impide el correcto funcionamiento de los mecanismos de autenticación utilizados.

En el caso del **TripWire**, el programa puede seguirse utilizando a pesar del problema. Los resultados obtenidos en las pruebas de integridad de los archivos son diferentes a los que se obtendrían con los mismos archivos en otras máquinas, pero la utilización de estos resultados es exclusivamente interna, y siguen sirviendo como prueba de la integridad de los archivos del sistema operativo en la Cray.

### Método utilizado

En todos los casos, las herramientas de seguridad se instalaron primero en *ds5000*, en su calidad de coordinadora de todas las actividades de seguridad. Después de un período de prueba en esta máquina (que dependiendo de la herramienta en cuestión osciló

Máquina	COPS	TCP-Wrapper	passwd+	Crack	S/Key	TripWire	SATAN
<i>ds5000</i>	✓	✓	✓	✓	✚	✓	✓
<i>sirio</i>	✓	✓	✗	✓	◇	✓	②
<i>polaris</i>	✓	✓	✓	①	✚	✓	②
<i>mira</i>	✓	✓	✓	①	✚	✓	②
<i>capella</i>	✓	✓	✓	①	✚	✓	②
<i>deneb</i>	✓	✓	✓	①	✚	✓	②
<i>diphda</i>	✓	✓	✗	①	✚	✓	②
<i>andromeda</i>	✓	✓	✓	①	✚	✓	②
<i>nocdos</i>	✓	✓	✓	①	✚	✓	②
<i>mezcal</i>	✓	✓	✓	①	✚	✓	②
<i>tequila</i>	✓	✓	✗	①	✚	✓	②
<i>pulque</i>	✓	✓	✗	①	✚	✓	②
<i>xtabentun</i>	✓	✓	✗	①	✚	✓	②
<i>aldebaran</i>	✓	✓	✗	①	✚	✓	②
<i>casiopea</i>	✓	✓	✗	①	✚	✓	②
<i>pegasus</i>	✓	✓	✗	①	✚	✓	②

①El Crack para todas las demás máquinas se hace en *ds5000* y *sirio*.

②La revisión con SATAN de todas las máquinas se hace desde *ds5000*.

✓ Herramienta instalada y utilizada activamente.

✚ Herramienta instalada, pero no utilizada activamente.

◇ Herramienta en proceso de adaptación al sistema.

✗ Herramienta no instalada.

Tabla 6.1: Herramientas de seguridad instaladas en los sistemas de Supercómputo y Visualización en la DGSCA.

entre un par de días y varias semanas), las herramientas fueron siendo instaladas en las otras estaciones de trabajo de Supercómputo y Visualización. Siempre hasta el final se instalaron las herramientas en la supercomputadora Cray, debido a las políticas de utilización de este sistema, que especifican que cualquier programa instalado tiene que haber sido probado extensamente con anterioridad.

### Estado actual

Teniendo en cuenta todas las consideraciones realizadas, las herramientas de seguridad se encuentran actualmente en uso constante. La tabla 6.1 resume el estado de instalación de las diferentes herramientas en cada una de las máquinas consideradas en este proyecto.

## 6.3 Concentración de los reportes generados

El objetivo principal de la utilización de herramientas de seguridad es contar con información que permita detectar problemas y proponer soluciones. Para poder lograr este

objetivo es necesario analizar la información, y esta labor se facilita considerablemente al tener la información concentrada en un solo sitio.

### 6.3.1 Medios de transferencia de la información

Afortunadamente, el problema de transferencia de información entre máquinas se reduce considerablemente al tener los sistemas conectados a la red. Unix ofrece mecanismos comunes y sencillos para la transferencia de información en diferentes formatos entre las máquinas. Los mecanismos utilizados en este proyecto son:

- Correo electrónico.
- FTP.

El correo electrónico se utiliza en casi todas las ocasiones, pues es un medio sumamente sencillo de utilizar, y que permite transferir archivos en formato de texto, que es el formato en el que todas las herramientas generan su información.

El FTP solamente es utilizado cuando no es factible la transferencia por correo electrónico. Este es el caso, por ejemplo, en la supercomputadora, donde no se encuentra habilitado el servicio de correo electrónico hacia otra máquinas, debido a políticas de uso establecidas por el Comité Académico de Supercómputo.

### 6.3.2 Cómo se transfiere la información

La gran mayoría de las herramientas cuentan con opciones de configuración que permiten especificar la dirección electrónica a la que se deben enviar los reportes generados. En esos casos, basta con especificar en dicha opción la cuenta `oss@ds5000.dgsca.unam.mx`, en la cual se realiza la concentración de los reportes generados por todas las demás máquinas.

En los casos de herramientas que no permiten especificar una dirección electrónica, sino que almacenan sus reportes en disco, se ejecuta automáticamente, después de la herramienta, un pequeño programa que lee el reporte generado y lo envía por correo electrónico a la dirección mencionada arriba.

Cuando no es posible utilizar correo electrónico, se ejecuta automáticamente, después de la herramienta, un programa que establece una sesión de FTP con la cuenta `oss` en `ds5000`, y realiza la transferencia del archivo.

### 6.3.3 Qué se hace con los reportes

Actualmente, los únicos reportes que son procesados de forma automática son los generados por `COPS`. Estos reportes son procesados utilizando la herramienta `New CARP`. Los reportes generados por este programa son enviados a los administradores de cada una de las máquinas de forma automática.

Próximamente serán procesados también los reportes generados por todas las herramientas instaladas, utilizando el programa `SAINT`.

En todos estos casos, el administrador de cada sistema recibe, por correo electrónico, el reporte generado por la herramienta utilizada para el análisis.

## 6.4 Formación de un grupo de administradores

Una de las partes medulares de este proyecto es la formación de un grupo humano a nivel UNAM, preocupado por la buena administración y la seguridad en Unix. Los orígenes de este grupo, sin embargo, se dan en un grupo formado dentro de la DGSCA, integrado por los administradores de todas las máquinas de los departamentos de Departamento de Supercómputo, Laboratorio de Visualización y el Departamento de Redes.

Entre junio y diciembre de 1993 se realizaron de forma periódica (casi siempre semanal) reuniones de todos los administradores de las estaciones de trabajo y la supercomputadora. En estas sesiones se trataron múltiples aspectos de administración y seguridad en Unix, como los siguientes:

- Conceptos básicos de seguridad en cómputo.
- Conceptos básicos de seguridad en Unix.
- Manejo de claves y grupos de usuarios.
- Configuración de servicios de red.
- Instalación de herramientas de seguridad.

Este grupo, compuesto por aproximadamente 10 personas, fue el semillero donde surgieron muchas de las ideas e iniciativas que posteriormente serían extendidas a toda la UNAM. Fue un foro donde se discutió y se dió asesoría sobre muchos aspectos de administración en general, así como puntos más específicos sobre seguridad, pero siempre restringido a los sistemas de Departamento de Supercómputo, Laboratorio de Visualización y el Departamento de Redes.

En el mes de diciembre de 1993 se suspendieron las reuniones de este grupo, al combinarse con el recién creado Grupo de Administración y Seguridad en Unix (GASU), discutido en el siguiente capítulo.

## Capítulo 7

# Grupo de Administración y Seguridad en Unix

Los beneficios de proporcionar a los usuarios y administradores una capacitación adecuada y fuentes de información apropiadas para la realización de su trabajo son tan grandes, que desde el principio de este proyecto se consideró como parte esencial del mismo la formación de un grupo de personas interesadas por la seguridad en Unix. En este capítulo se describen los esfuerzos en esta dirección, así como los resultados logrados, y los planes a futuro para este grupo.

### 7.1 Fundación y forma inicial de operación

En el mes de diciembre de 1993 se decidió convocar a la formación oficial de un grupo de administradores a nivel UNAM, que permitiera extender los esfuerzos por mejorar la seguridad en Unix en toda la Universidad.

#### 7.1.1 ¿Quiénes?

La primera pregunta que tuvo que ser contestada al formar este grupo fue: ¿quiénes deben formar parte de él? Toda la gente involucrada en un sistema de cómputo es importante para la seguridad. Sin embargo, se decidió enfocar inicialmente la atención en el grupo que más influencia puede tener sobre los otros: los administradores. El proporcionar información, herramientas y capacitación a los administradores de los sistemas Unix es el primer paso en lograr la difusión de la cultura de seguridad, pues los administradores pueden ejercer una influencia poderosa en dos direcciones: los usuarios de los sistemas, y los directivos que pueden tomar decisiones de fondo acerca de esos sistemas y su utilización, así como apoyar las acciones realizadas.

### 7.1.2 Primera reunión

Así, se decidió convocar a una junta el día 6 de diciembre de 1993, cuyo objetivo sería conformar oficialmente dicho grupo. La convocatoria se hizo llegar a la gran mayoría de los centros de cómputo de la UNAM, principalmente a través de los siguientes medios:

- Correo electrónico.
- Servicio postal.
- Mensajeros.

A continuación se presenta la orden del día para dicha reunión, así como una descripción de los puntos cubiertos en ella.

**Presentación** Con el objetivo de que los asistentes conocieran a las personas detrás de este proyecto, que principalmente son el autor de este trabajo y Martha A. Sánchez Cerezo, jefe del Departamento de Supercómputo de la DGSCA.

**Por qué y para qué estas juntas** Presentar los objetivos del grupo que se pretendía formar en relación al Proyecto de Seguridad en Cómputo, y que se pueden resumir como sigue: formar un grupo de administradores de Unix en la UNAM que estén preocupados por la seguridad de los sistemas, y que estén dispuestos a participar en acciones tendientes a mejorar dicha seguridad.

**Acciones a tomar** Presentar un plan de acción inicial que se propuso a los administradores, y que se describe en un documento distribuido ese día, y que se puede consultar en el apéndice E. Este documento, básicamente, sugiere la realización de las mismas actividades realizadas en DGSCA. Este documento también fue difundido a diferentes institutos, facultades y dependencias de la UNAM a través del Dr. Enrique Daltabuit Godás, que en ese tiempo era Director de Cómputo para la Investigación en DGSCA.

También en este punto se mencionó la idea de la formación de una lista de correo electrónico como medio de comunicación del grupo. Esta idea fue ampliada y llevada a cabo posteriormente, tomando en cuenta algunas aportaciones importantes realizadas durante esta junta.

**Entrega del manual de instalación de COPS, TCP-Wrapper y passwd+** Por considerarse estas herramientas como las más importantes dentro del paquete utilizado en DGSCA, se entregó a los administradores un manual para su instalación, y que se incluye en el apéndice F de esta tesis.

**Tutorial sobre instalación y configuración del COPS** Este programa se considera como el primero que hay que instalar, por ser el de más amplia cobertura en la seguridad de un sistema Unix. Por lo tanto, se expuso en forma breve el proceso de instalación y utilización de este programa, con la esperanza de que el mayor número posible de los asistentes lo instalara y utilizara en los sistemas a su cargo.

**Foro de Discusión** Este punto tuvo como objetivo fomentar la discusión y la participación de los asistentes con ideas, comentarios, preguntas o sugerencias. Para muchos de los asistentes, la seguridad, e incluso la administración de Unix, eran temas totalmente nuevos, por lo que las participaciones fueron escasas.

**Determinación de la periodicidad para la realización de las juntas** Obviamente, el proyecto del grupo no podía quedarse en una sola reunión. La idea inicial fue realizar juntas periódicas en las que se discutieran temas diversos de administración y seguridad. Después de una breve consulta con los asistentes de la plática se determinó que las juntas fueran el primer lunes de cada segundo mes, por lo que la siguiente reunión fue programada para el 7 de Febrero de 1994.

**Registro de asistentes** Con el fin de tener una lista inicial de los integrantes del grupo, se solicitó a los presentes que proporcionaran los siguientes datos:

- Nombre.
- Dirección electrónica, en caso de tenerla.
- Sitio de trabajo.
- Número aproximado de máquinas a su cargo.

Estos datos permitieron tener una idea inicial de la composición del grupo, así como contactar posteriormente a los integrantes mediante correo electrónico.

### 7.1.3 Resultados de la primera reunión

A la junta inaugural del grupo de administradores asistieron 45 personas en representación de 15 entidades diferentes: 14 dependencias de la UNAM y una empresa particular. El común denominador fue que todos eran administradores de máquinas Unix. Se manifestó bastante interés por la formación del grupo, así como por la realización de actividades que promovieran su crecimiento en el futuro. El grupo todavía no tenía un nombre oficial, pero ya se comenzaba a poner en actividad.

## 7.2 Crecimiento y evolución

### 7.2.1 Segunda reunión

Como se mencionó anteriormente, la segunda reunión del grupo de administradores se realizó el 7 de febrero de 1994. La orden del día para esta junta fue:

**Registro de asistentes** En esta ocasión la asistencia fue de 35 personas, representando a un total de 11 entidades diferentes.

**Aviso de la formación de una lista de correo electrónico.**

**Creación de un servidor de FTP anónimo de seguridad** Este medio es muy importante para la difusión de material relacionado con seguridad. En esta junta se notificó de la apertura de este servicio.

**Propuesta de seminarios, cursos y talleres** Uno de los objetivos principales de este grupo, como ya se ha mencionado en múltiples ocasiones, es la capacitación de los administradores en diversos aspectos de administración y seguridad en Unix. Por tal razón, se propuso la impartición periódica de seminarios, talleres y cursos, en los que se trataran distintos temas.

**Incidentes de seguridad** Durante el período vacacional de fin de año de 1993 ocurrieron dos incidentes de seguridad que involucraron a la UNAM:

1. Intentos de acceso no autorizado a la supercomputadora Cray de la UNAM, realizados desde universidades en las ciudades de Guadalajara y Monterrey. Estos intentos no fueron exitosos, gracias al correcto funcionamiento de los mecanismos de seguridad que en ese momento ya se encontraban instalados en la supercomputadora.
2. Accesos no autorizados a una supercomputadora localizada en el Centro de Supercómputo de Carolina del Norte (NCSC) en los Estados Unidos. Estos accesos fueron realizados desde sistemas localizados en la UNAM, específicamente en el Instituto de Astronomía. Las autoridades del NCSC se comunicaron con DGSCA para solicitar una averiguación al respecto y avisar de la posibilidad de que algunos sistemas de la UNAM estuvieran comprometidos.

Aunque ninguno de estos incidentes tuvo consecuencias graves para la UNAM, son un aviso de que en seguridad en cómputo no se puede descansar. La Lic. Martha A. Sánchez Cerezo tomó la palabra en este punto para recalcar la importancia de que los administradores de sistemas de cómputo se mantengan siempre alertas para detectar cualquier actividad “sospechosa”, y para corregir a la brevedad posible los problemas que se presenten.

### 7.2.2 Resultados de la segunda reunión

Desde la primera junta del grupo de administradores se vio la necesidad de contar con un medio de comunicación que fuera más efectivo y flexible que las reuniones “físicas” realizadas periódicamente. Esta idea se vio reforzada dada la menor asistencia de personas a la segunda junta. A continuación se describe la solución dada a este problema.

### 7.2.3 Lista de correo electrónico y consolidación de GASU

Una verdad en la vida “moderna” se resume en esta frase: Hay demasiado trabajo y muy poco tiempo. Esto, aunado al hecho de que la gran mayoría de los administradores realizan esta tarea aparte de algunas otras en su sitio de trabajo, ocasiona el problema de que para muchas personas resulta imposible movilizarse para asistir a una junta periódica. Resulta prácticamente imposible reunir físicamente, en un mismo lugar y a una misma hora, a todos los integrantes de un grupo, sea del tipo que sea.

Esto ha llevado a la búsqueda de soluciones tecnológicas que le permitan a los seres humanos permanecer en contacto sin tener que desplazarse físicamente de un lado a

otro. Afortunadamente, existen actualmente múltiples soluciones a este problema. Una de ellas, una de las más prácticas en Internet, son las listas de correo electrónico. Dado que la gran mayoría de los integrantes del grupo de administración tienen acceso a REDUNAM, y de ahí a Internet, se consideró apropiado formar una lista de correo electrónico que sirviera como medio de comunicación entre los integrantes del grupo.

### **Antecedentes de listas de correo electrónico en la UNAM**

Durante el período de planeación de esta lista de correo se encontró la existencia previa de una lista dedicada a la administración de Unix. Esta lista se llamaba *cvyd* (“Corre Ve Y Dile”) y fue administrada, durante su existencia, por Eduardo Sacristán Ruiz-Funes (Instituto de Astronomía) y David Vázquez (Instituto de Geofísica).

Esta lista se abandonó debido principalmente a:

1. Falta de tiempo por parte de los organizadores.
2. Falta de otra persona que se hiciera cargo de ella a falta de los organizadores iniciales.
3. Falta de los recursos necesarios para administrar una lista de correo: equipo de cómputo y espacio en disco.

Cabe mencionar que los dos administradores de la antigua *cvyd* estuvieron en el grupo de administración desde la primera junta, y ambos aportaron ideas muy valiosas para la formación de la nueva lista.

### **Primera etapa: lista *unam-admin***

El primer paso después de instalar el servidor de listas de correo fue formar una lista de correo “temporal”, que sirviera como foro de discusión para llevar a la determinación de las características definitivas del grupo de administración y de su lista de correo. Esta lista fue llamada *unam-admin*, y fue creada el 26 de Enero de 1994.

A esta lista se suscribió automáticamente a todos los asistentes a la primera junta. También se integró en ella la lista de suscriptores de la antigua lista *cvyd*, proporcionada por sus administradores.

En *unam-admin* se lanzó la convocatoria a proponer nombres definitivos para la lista y para el grupo de administración.

### **Segunda etapa: lista *gasu***

En respuesta a la convocatoria, muchas personas propusieron nombres para la lista y para el grupo de administración. El nombre aceptado finalmente para ambos fue GASU (Grupo de Administración y Seguridad en Unix). Por lo tanto, la lista *unam-admin* fue cerrada, para ser sustituida por *gasu*, que se encuentra en operación hasta la fecha.

### 7.2.4 Nuevas áreas de acción

La convocatoria incluía también preguntas sobre las características que debería tener el grupo, así como las actividades que deberían realizarse en opinión de los administradores. De esta encuesta resultaron los siguientes puntos:

1. Ampliar el enfoque del grupo. Inicialmente se había planteado como un “grupo de seguridad en Unix”. La opinión casi unánime fue que debería ser un “grupo de administración y seguridad en Unix”, dado que ambos temas están tan íntimamente ligados.
2. Abrir el grupo. Inicialmente se formó únicamente con gente de la UNAM, pero la opinión general fue que se permitiera el acceso también a personas de otras instituciones, para fomentar el crecimiento y el enriquecimiento del grupo.

Estas sugerencias fueron adoptadas inmediatamente, puesto que no restan nada a los objetivos originales del grupo, y a cambio presentan la posibilidad de enriquecerlo considerablemente.

## 7.3 Seminarios, cursos y talleres

De las actividades organizadas por GASU, quizá la más promisoría sea la organización de seminarios, cursos y talleres. Estas actividades abren la posibilidad de proporcionar información útil, práctica y real a los administradores de sistemas Unix en la UNAM.

Dada la naturaleza heterogénea y abierta de GASU, el nivel de experiencia y conocimiento de los integrantes varía desde quienes tienen una semana administrando un sistema Unix hasta aquellos para quienes Unix es ya un estilo de vida. También las áreas de acción varían enormemente: administración general de Unix, redes y telecomunicaciones, programación en Unix, etc.

La idea desde la formación de GASU fue aprovechar esta gran variedad para el beneficio del grupo. En la segunda junta se propuso que aquellas personas que tuvieran experiencia sobre algún tema específico de administración de Unix presentaran un seminario, curso o taller sobre el tema, para beneficio de los demás integrantes del grupo. Esta idea, de florecer y mantenerse, permitiría a GASU ser un grupo en constante crecimiento, con aportaciones nuevas de sus diferentes miembros. Al fomentar la retroalimentación dentro del mismo grupo, estas actividades fomentarían la elevación del nivel de conocimiento del grupo en general.

Hasta la fecha se han impartido dos seminarios. La idea de los seminarios es que duren un día completo, y que proporcionen información detallada sobre algún tema. De preferencia, los seminarios tienen que ser “autocontenidos”, en cuanto a que los asistentes no necesiten conocimientos adicionales (salvo los prerequisites que sean indispensables) para poder hacer uso de la información proporcionada. Los seminarios realizados hasta la fecha son los siguientes:

**Uso y configuración de DNS** Impartido por Eduardo Sacristán Ruiz-Funes (Instituto de Astronomía) el 7 de marzo de 1994. Cubrió los conceptos básicos, así como instrucciones detalladas de utilización, configuración y administración de DNS

(*Domain Name Service*), el protocolo utilizado en Internet para conversión de los nombres de máquinas (por ejemplo `ds5000.dgsca.unam.mx`) a los números que son utilizados en última instancia para establecer la comunicación (por ejemplo, `132.248.204.8`).

**Introducción al uso de utilerías de Unix** Impartido por el autor de este trabajo el 2 de Mayo de 1994. Cubrió la utilización de las utilerías que todo administrador de Unix debe dominar para poder realizar de forma más eficiente su labor, tales como **grep**, **sort**, **sed** y **awk**.

Se tiene proyectada la realización de los siguientes seminarios:

**Introducción a TCP/IP** Impartido por Rafael Lacambra Macedo.

**Introducción a la administración en Unix** Impartido por Martha A. Sánchez Cerezo.

**Conceptos básicos de seguridad** Impartido por Diego Zamboni.

**Instalación de herramientas de seguridad** Impartido por Diego Zamboni.

## 7.4 Difusión de las herramientas de seguridad

Otro de los objetivos fundamentales de GASU es difundir en la UNAM la utilización de las herramientas de seguridad existentes. Como mínimo, se tiene el objetivo de promover las herramientas que, de acuerdo a los análisis realizados, son más útiles y completas. Para ello, hasta el momento se han llevado a cabo las siguientes acciones:

- Difusión del *Manual de Instalación de COPS, TCP-Wrapper y passwd+* [Zam93].
- Impartición de un tutorial sobre la instalación de COPS en la primera junta del grupo de administración.
- Distribución en la lista *gasu* de los siguientes tutoriales:
  - *TCP-Wrapper: Introducción, instalación y uso* [Zam94b].
  - *Comentarios sobre configuración de passwd+* [Zam94a].
- Otorgamiento constante de asesorías sobre instalación de herramientas de seguridad a miembros del grupo que así lo solicitan, normalmente a través de correo electrónico.
- Distribución de las herramientas de seguridad a través del servidor de FTP anónimo.
- Difusión de información sobre herramientas de seguridad a través de la página de WWW de seguridad.

Con el fin de seguir promoviendo la utilización de estas herramientas, se tienen planeadas las siguientes actividades:

- Mantener al día la información sobre las herramientas ya utilizadas (problemas presentados, nuevas versiones, etc.).
- Continuar con la investigación, prueba y, en su caso, utilización y difusión de nuevas herramientas de seguridad.
- Generación y/o ampliación de tutoriales sobre las siguientes herramientas:
  - COPS
  - passwd+
  - Crack
  - S/Key
  - TripWire
- Organización de seminarios sobre instalación de diversas herramientas de seguridad.

## 7.5 Presente y futuro de GASU

Actualmente, GASU cuenta con 142 integrantes, y este número sigue creciendo constantemente. En la lista de correo se discuten todo tipo de temas; entre los más relevantes encontramos los siguientes:

- Preguntas y asesorías sobre configuración de sistemas (discos, memoria, etc.).
- Preguntas y asesorías sobre instalación de *software* de dominio público.
- Preguntas y asesorías sobre diversas versiones de Unix.
- Preguntas y asesorías sobre los mecanismos para proporcionar o utilizar servicios de red.
- Sugerencias sobre programas o servicios diversos que pueden ayudar a hacer mejor uso de los sistemas.
- Anuncios de eventos como pláticas, cursos, seminarios y talleres de diversa índole (no necesariamente realizados bajo el auspicio de GASU).

GASU ha ido creciendo lenta, pero constantemente. Actualmente ya es bastante conocida en los círculos de administradores y usuarios de sistemas Unix en la UNAM y en algunas otras instituciones del país. Cuenta ya con miembros en distintos puntos del país (Guanajuato, Chiapas, Jalisco, Monterrey y Coahuila), así como miembros que pertenecen a otras instituciones educativas (ITAM, CONACYT, UAM) e incluso a dependencias gubernamentales (SCT).

### 7.5.1 ¿Que opina la gente?

Con motivo del primer aniversario de la formación de GASU, en enero de 1995 se realizó una encuesta entre los integrantes del grupo para recopilar opiniones acerca del funcionamiento de GASU en este año, sugerencias para su funcionamiento futuro, e información estadística sobre la seguridad en cómputo en las diferentes instituciones adscritas al grupo. La encuesta fue diseñada para ser llenada por institución, dependencia o área, y no de forma individual. Constaba de las siguientes preguntas:

1. Dependencia, facultad o instituto.
2. Área.
3. Descripción de sistemas operativos que se utilizan en su sitio de trabajo.
4. ¿Existe en su sitio de trabajo una preocupación por la seguridad de los sistemas de cómputo?
5. ¿Qué medidas de seguridad se han tomado durante el último año (1994)?
6. El grupo GASU ha tenido alguna influencia en las políticas, herramientas y mecanismos de seguridad utilizados en su sitio de trabajo, así como en la administración en general de los sistemas Unix?
7. Otros comentarios.

Esta encuesta fue realizada a través de correo electrónico en *gasu*, ofreciendo también la posibilidad de contestarla a través de WWW. Los resultados arrojados por esta encuesta fueron los siguientes:

**Número de encuestas recibidas:** 12

**Áreas y dependencias representadas:**

- Laboratorio de Cómputo Avanzado, División de Estudios de Posgrado de la Facultad de Ingeniería.
- Departamento de Cómputo, Instituto de Ciencias Nucleares.
- Departamento de Administración de Supercómputo, Dirección General de Servicios de Cómputo Académico.
- Laboratorio de Visualización, Dirección General de Servicios de Cómputo Académico.
- Departamento de Cómputo, Instituto de Química.
- Departamento de Redes y Telecomunicaciones, Dirección General de Servicios de Cómputo Académico.
- Coordinación Técnica de Redes e Interoperabilidad, Dirección de Cómputo para la Administración Académica.
- Departamento de Cómputo, Instituto de Astronomía.

- Departamento de Ingeniería Eléctrica, División de Estudios de Posgrado de la Facultad de Ingeniería.
- Departamento de Astrofísica, Instituto Nacional de Astrofísica, Óptica y Electrónica.
- Departamento de Ingeniería en Informática, División de Estudios de Posgrado de la Facultad de Ingeniería.
- Departamento de Biología, Centro de Ecología.
- Departamento de Ecología Terrestre, Centro de Ecología.

**Número de sistemas Unix representados:** 126.

**Número de sistemas no Unix representados:** 211.

**Versiones de Unix utilizadas:** AIX, Irix, Solaris, HP-UX, Ultrix, SunOS, Linux, UNICOS, NEXTSTEP, SCO Unix. La figura 7.1 muestra la distribución de estas versiones de Unix en la población total muestreada.

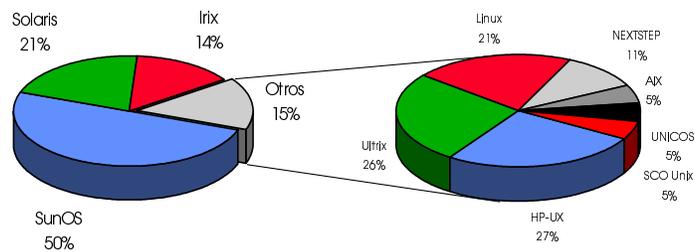


Figura 7.1: Versiones de Unix representadas en la encuesta de GASU

**Version de Unix más utilizada:** SunOS, en sus versiones 4.1.x.

**Versión de Unix menos utilizada:** SCO Unix (sin contar UNICOS, que se utiliza solamente en la supercomputadora Cray).

**Sistemas operativos no Unix utilizados:** MS-DOS, Windows NT, System 7 (Macintosh), Windows for Workgroups.

**Preocupación por la seguridad:** Todas las encuestas indicaron la existencia de una preocupación sobre la seguridad en cómputo, aunque una de ellas indicó que dicha preocupación es solamente en cuanto a virus.

**Influencia de GASU:** De las encuestas recibidas, el 58% (7 de ellas) manifestaron haber recibido influencia de GASU en cuanto a políticas, herramientas y mecanismos de seguridad, así como en la administración de los sistemas Unix.

**Medidas de seguridad:** Las medidas de seguridad más comunes que han sido tomadas son:

- Reinstalación de sistemas operativos.
- Instalación de herramientas, principalmente TCP-Wrapper y COPS.
- Restricción de acceso físico a los sistemas.
- Restricción de acceso lógico a las máquinas, utilizando TCP-Wrapper.
- Mantenimiento de bitácoras de administración de los sistemas.
- Revisión de las bitácoras del sistema operativo.
- Realización de pláticas con los usuarios.
- Obligar a los usuarios a cambiar periódicamente sus *passwords*.

**Otros comentarios:**

“La gracia de GASU es que mantiene en contacto a los administradores de cómputo de la Universidad y de otros sitios.”

“La idea de GASU es realmente buena, pero creo que muchos usuarios no están listos para usar una lista de correo [...]”

“Tal vez encontrar nuevos mecanismos de intercambio de información y conocimientos, por ejemplo organizar seminarios a través de Internet.”

“Para hacer más dinámica la lista y crear un ambiente de ayuda y conocimiento, se podrían realizar una especie de preguntas y respuestas acerca de administración y seguridad, o mensajes cortos dando *tips* al respecto.”

“La lista ha sido útil y creo que es un buen inicio en cuanto a cultura de seguridad, pero estoy convencido de que la seguridad en México y en la UNAM está muy relajada, más que por descuido es por falta de una preparación real de administración de sistemas, ya que los administradores se van improvisando.”

De acuerdo a estos resultados es posible apreciar que los integrantes de GASU lo consideran como algo útil, pero al que todavía le falta mucho madurar y crecer. En general, los encuestados reconocen a GASU como una iniciativa útil y con futuro, aunque no todos se han sentido influenciados por el grupo. También de los comentarios es posible extraer algunas ideas que pueden ser implementadas a futuro, como organización de seminarios a través de la red y creación de documentos con respuestas a preguntas frecuentes (conocidos comúnmente como FAQ, por *Frequently Asked Questions*) sobre seguridad.

En cuanto a las estadísticas, es posible apreciar el claro dominio numérico que sigue existiendo por parte de las computadoras personales que no utilizan Unix: casi duplican en número a los sistemas Unix. Las computadoras personales no entran dentro del campo directo de acción de GASU, pero la cantidad de ellas haría factible, en un momento dado, dedicarles una mayor atención específica dentro de las actividades del grupo.

### **7.5.2 El futuro**

Se puede considerar a GASU como el objetivo medular de esta tesis. El tener un grupo bien organizado, sólido y en continuo desarrollo proporciona todas las facilidades para la realización de los demás objetivos. Actualmente es necesario que el autor de este trabajo “cuide y alimente” a GASU con continuas aportaciones, invitaciones, mensajes, sugerencias e incluso declaraciones polémicas, con el fin de motivar discusión en la lista de correo. Pero cada vez más, GASU es un cuerpo independiente de las personas, que crece y se alimenta por si mismo, y cuya existencia no está determinada por la presencia de alguien en particular.

Esa es justamente la idea. Algún día, quienes integramos actualmente este proyecto dejemos tal vez la UNAM, pero GASU y el proyecto de seguridad deben sobrevivir a las personas, y convertirse en instituciones sólidas que garanticen la difusión y perduración de la cultura de la seguridad en cómputo dentro de la Universidad.

## Capítulo 8

# Día Internacional de la Seguridad en Cómputo

La preocupación por difundir la cultura de seguridad ha ido creciendo a nivel mundial. Es por eso que desde 1988, la *Association for Computing Machinery* (ACM) estableció la realización anual del *Día Internacional de la Seguridad en Cómputo* (DISC). Este evento, organizado y coordinado por el Grupo de Interés Especial en Seguridad, Auditoría y Control (SIGSAC) de la ACM, tiene como objetivo difundir la cultura de la seguridad, así como la conciencia de que todo aquel que utilice una computadora debe tomar las medidas apropiadas para proteger a la computadora, los programas y la información.

### 8.1 Antecedentes

El DISC nació en 1988, cuando el SIGSAC de la ACM decidió llamar la atención del público hacia los temas de seguridad en cómputo. Se eligió como fecha de realización del DISC el primer día laborable de Diciembre, para reforzar la atención a la seguridad durante las fiestas decembrinas, en las que normalmente las máquinas están más desatendidas que durante el resto del año.

Una de las ideas del DISC es que nadie tenga pretexto para no asistir. Es por esto que desde el principio se organizó de forma distribuida: se realiza de forma simultánea (dentro de lo posible) en todas las organizaciones participantes. El comité del DISC en la ACM determina el tema central de cada año y otorga la información y el apoyo necesarios, pero cada institución es responsable de organizar los eventos de forma local. Esto, además de facilitar el alcance del DISC a todos los usuarios de computadoras, permite a cada lugar organizarlo de acuerdo a sus necesidades particulares. También se permite flexibilidad en cuanto a la fecha de realización del evento, siempre y cuando sea alrededor de la fecha oficial fijada por el comité organizador.

## 8.2 El DISC en la UNAM

La idea de organizar el DISC en la UNAM surgió durante la segunda mitad de 1993. Sin embargo, debido a presiones de tiempo, fue imposible organizarlo oficialmente ese año, llevándose a cabo solamente la impartición de un seminario por parte del autor de este trabajo, titulado “Conceptos básicos de seguridad en Cómputo”, y realizado el día primero de diciembre de 1993. En este seminario se contó con poca asistencia.

La situación fue diferente en 1994. En este año se planearon con tiempo las actividades. Se obtuvo el reconocimiento oficial de la ACM para la DGSCA como participante en el DISC, con lo cual la UNAM se convirtió en la primera institución mexicana en participar en dicho evento, cuya realización se fijó para el lunes 5 de diciembre.

En 1994, el tema del DISC, designado por su comité organizador, fue “Responsabilidad Individual”.

### 8.2.1 Convocatoria

La difusión del DISC se realizó principalmente a través de GASU, así como a través del servidor de FTP anónimo y de WWW. La convocatoria para presentar ponencias se difundió a principios del mes de Septiembre, fecha en la cual se publicó también en la Gaceta UNAM.

La convocatoria para asistencia se difundió en el mes de Noviembre, abriéndose el registro de asistentes a través de estos medios.

### 8.2.2 Respuesta

La respuesta a la convocatoria del DISC por parte de los asistentes sobrepasó todas las expectativas. Dos semanas antes de la realización del evento ya estaban registradas más de 60 personas.

En cuanto a ponencias, la respuesta fue menos entusiasta debido a la falta de trabajo sobre seguridad en nuestro país. Sin embargo, fue posible al final reunir 10 pláticas sobre temas muy diversos de seguridad, desde técnicas básicas para usuarios hasta conceptos de seguridad corporativa y en redes. El apéndice H contiene el programa detallado del DISC 1994.

### 8.2.3 Realización

La realización del DISC el 5 de Diciembre de 1994 fue un éxito completo. Se contó con la asistencia de más de 70 personas durante todo el día, llenando casi por completo el auditorio de la DGSCA. Aparte de las ponencias, se contó con el apoyo del Dr. Victor Guerra Ortiz, director general de la DGSCA, y quien inauguró el evento. Durante la mañana estuvo montada afuera del auditorio una exposición de bibliografía de seguridad que atrajo bastante la atención de los asistentes.

Entre los participantes estaban representadas múltiples organizaciones: facultades, institutos y dependencias de la UNAM, otras instituciones educativas (como la UAM), empresas privadas (Cray Research de México, Hewlett-Packard y Red Uno) e incluso

la Secretaría de la Defensa, que asistió con un contingente de aproximadamente 15 personas.

### **8.3 Resultados obtenidos y planes a futuro**

Se considera la realización del DISC en la UNAM en 1994 como un hecho sumamente positivo y que da una gran esperanza para la cultura de la seguridad en cómputo en México. Se pretende continuar con la organización del DISC anualmente, y se espera contar con una participación creciente de diferentes instituciones educativas, comerciales y de gobierno. También se espera que crezca el número de trabajos presentados cada año.

Aunque el DISC es solamente un día al año, su intención es crear conciencia para tomar en cuenta la seguridad durante todos los días. El que se logre este objetivo depende principalmente de su celebración año con año, aunado con la difusión y buena organización que se le de, hasta que llegue a ser un evento anual al que muchas personas en México esperen con entusiasmo.



## Capítulo 9

# Capacitación

Regresando a un refrán ya mencionado: “*El buen juez por su casa empieza*”, para poder impulsar un proyecto amplio y duradero de seguridad, debe comenzarse por la capacitación y formación del personal directamente involucrado en la coordinación y organización de dicho proyecto.

Afortunadamente, el proyecto de seguridad contó desde el principio con apoyo amplio de las autoridades de DGSCA, incluyendo al Dr. Victor Guerra Ortiz (Director de la DGSCA), el Dr. Enrique Daltabuit Godás (Director de Cómputo para la Investigación hasta el mes de marzo de 1995), el Dr. Alberto Alonso y Coria (Director de Cómputo para la Investigación desde marzo de 1995) y la Lic. Martha A. Sánchez Cezeo (Jefe del Departamento de Supercómputo de la DGSCA). Esto ayudó a contar con los recursos necesarios para la realización de todas las actividades de capacitación realizadas dentro del proyecto, que se describen en este capítulo.

### 9.1 Conferencia de Seguridad en Cómputo y Comunicaciones

Desde 1993, el SIGSAC de la ACM organiza la *Conferencia de Seguridad en Cómputo y Comunicaciones*. Esta conferencia tiene como objetivo ofrecer una panorámica amplia de seguridad, combinando el trabajo teórico con la práctica, e incluyendo dentro de su programa todos los aspectos de seguridad, desde seguridad tradicional en Unix hasta temas avanzados de criptografía.

También desde 1993, personal de la DGSCA ha asistido a dicha conferencia. Esto ha servido para cumplir los siguientes objetivos:

- Obtener información sobre los proyectos y tecnologías más vanguardistas en materia de seguridad.
- Mantenerse al día en la evolución de los mecanismos tradicionales de seguridad, y obtener información sobre nuevas técnicas de seguridad.

- Crear relaciones y contactos con otras personas involucradas en seguridad, con quienes se ha logrado compartir problemas, ideas y posibles soluciones.
- Obtener información sobre las últimas publicaciones de seguridad.
- Obtener información sobre otros eventos relacionados con la seguridad en cómputo.

Se tiene planeada la asistencia a futuras ediciones de este evento. No se descarta tampoco la posibilidad de asistencia a otros eventos relacionados con seguridad, si se considera que pueden aportar conocimientos e información que sean útiles para el proyecto de seguridad en la UNAM.

Obviamente, se espera que en el futuro la participación no se limite a asistir a las conferencias, sino que sea posible aportar ponencias con las experiencias, los conocimientos y los desarrollos logrados en la UNAM.

## 9.2 Formación de un seminario interno sobre seguridad y criptografía

Con el fin de difundir e incrementar el conocimiento sobre temas de seguridad en cómputo, desde el 18 de febrero de 1994 se ha realizado periódicamente en la DGSCA un Seminario de Seguridad y Criptografía, con la participación de las siguientes personas:

- Dr. Enrique Daltabuit Godás
- Lic. Martha A. Sánchez Cerezo
- Ma. Susana Soriano Ramírez
- Dr. Horacio Tapia Recillas
- Mat. Gerardo Vega Hernández
- Diego Zamboni

En este seminario, realizado semanalmente, se han discutido múltiples temas relacionados con seguridad, como son:

- Conceptos básicos de seguridad en cómputo.
- Conceptos básicos de criptografía.
- Historia de la criptografía.
- Análisis de diferentes algoritmos de cifrado, como DES, y RSA.
- Análisis de protocolos y herramientas de seguridad, como MLS, Kerberos, SECURE RPC y S/Key.
- Implementación en DGSCA de algoritmos de cifrado.

- Discusión de las ventajas y desventajas de diferentes esquemas, protocolos y herramientas de seguridad, para considerar su implementación en la DGSCA y la UNAM.

El método de funcionamiento de este seminario ha sido mediante exposiciones semanales por parte de sus mismos integrantes. También se ha contado periódicamente con aportaciones de personas externas, como el Dr. Guillermo Mallén (Universidad Iberoamericana) y Jeff McIver (Sun Microsystems de México). Esta forma de operación tiene los siguientes objetivos:

1. Fomentar la investigación y el estudio sobre temas de seguridad y criptografía.
2. Fomentar el buen entendimiento de dichos temas, al tener que preparar exposiciones sobre los mismos.
3. Difundir dicho conocimiento a todos los integrantes del grupo.

Cabe mencionar que en la primera sesión del seminario se invitó a personal de diferentes oficinas administrativas de la UNAM, incluyendo la oficina del Rector, con el fin de sondear la apreciación de la seguridad que se tiene en las “altas esferas” administrativas de la Universidad. Desgraciadamente, se observó una falta casi total de conocimiento y de interés en los temas de seguridad. Este es un problema grave, pero está fuera del alcance de esta tesis, que se preocupa principalmente por los aspectos técnicos y humanos de la seguridad en cómputo.

Sea como sea, el Seminario de Seguridad y Criptografía ha sido un mecanismo muy efectivo para la capacitación del personal directamente involucrado en el proyecto. Las exposiciones y discusiones generadas en su seno han sido la semilla de muchas de las otras acciones que se describen en esta tesis.

### 9.3 Adquisición de bibliografía sobre seguridad

La seguridad no es un tema nuevo. Como tal, mucho ha sido escrito sobre él. Recientemente, al pasar la seguridad a la vista del gran público mundial, las publicaciones sobre el tema se han multiplicado a un ritmo vertiginoso.

Parte de la tarea de los encargados del proyecto de seguridad es hacer todo lo posible por mantenerse al día en cuanto a las publicaciones importantes sobre seguridad en cómputo. Por lo tanto, se ha fomentado la adquisición de libros y publicaciones diversos sobre el tema, y la formación de una amplia bibliografía de seguridad que sirva como material de referencia para quien necesite consultarla.

En la bibliografía de este documento pueden encontrarse los títulos de algunos de los documentos más relevantes sobre seguridad, y que más han ayudado a la ejecución del proyecto. Para una lista completa de publicaciones de seguridad y temas relacionados, consultar [oST89].

## 9.4 Creación de una carpeta de seguridad

Durante el desarrollo de este proyecto se ha recolectado una cantidad considerable de información sobre distintas áreas de la seguridad en cómputo, desde herramientas para seguridad en Unix hasta estudios teóricos sobre criptografía.

Con el fin de proporcionar una referencia a quienes en futuro continúen con la coordinación de este proyecto, todo el material estudiado y desarrollado durante su tiempo de vida ha sido recopilado en una “Carpeta de Seguridad”. Esta carpeta consta actualmente de dos gruesos volúmenes, y nuevo material se le adiciona constantemente.

Esta carpeta fue recopilada por el autor de esta tesis, con las aportaciones de muchas otras personas involucradas en este proyecto. Cabe mencionar que gran parte de la información ha sido obtenida a través de Internet.

La carpeta se encuentra disponible en el Departamento de Supercómputo de la DGSCA, y ahí permanecerá mientras dicho departamento sea el centro de coordinación del proyecto de seguridad, a disposición de cualquier persona que desee consultarla.

## 9.5 Contacto con expertos en seguridad

En toda disciplina es importante saber aprender de las personas que tienen más experiencia y conocimientos sobre el tema, y la seguridad en cómputo no es la excepción. Durante el desarrollo de esta tesis se mantuvo contacto en múltiples ocasiones con reconocidas personalidades de la seguridad. Esta comunicación siempre fue a través de correo electrónico, y los temas que se trataron cubrieron temas tan diversos como:

- Preguntas sobre temas técnicos específicos de seguridad en Unix.
- Preguntas sobre problemas al compilar o utilizar herramientas de seguridad.
- Asesoría para el desarrollo de herramientas de seguridad.
- Aportación de correcciones a errores encontrados en herramientas de seguridad.
- Participación conjunta para el mejoramiento de herramientas de seguridad de dominio público.

Las personas con las que se estableció comunicación en diferentes momentos fueron:

- Simson Garfinkel
- Gene Spafford
- Eugene Kim
- Dan Farmer
- Wietse Venema

## 9.6 Plan de becarios de supercómputo

El plan de becarios de supercómputo se lleva a cabo de manera anual, y tiene como objetivo la formación de recursos humanos con conocimientos de supercómputo, visualización y optimización, así como utilización y administración de Unix, programación en **C** y **FORTRAN**, y utilización de diversas bibliotecas para programación gráfica y científica.

El plan de becarios de supercómputo se encuentra ya en su tercera generación (94–95), y en ella se incluyó ya el curso “Temas Avanzados de Seguridad en UNICOS”, con el que se pretende dar a los becarios los conceptos esenciales de seguridad en Unix y en el sistema operativo UNICOS. De este plan de becarios, 3 personas se encuentran trabajando en proyectos de seguridad que tienen como objetivos lograr una mayor difusión de información sobre seguridad, así como impulsar la utilización de herramientas y de los servicios ofrecidos por la coordinación del proyecto de seguridad.



## Capítulo 10

# Desarrollo de Herramientas de Seguridad

Aunque la gran mayoría del trabajo técnico en este proyecto se realizó con herramientas de seguridad existentes en el dominio público, hay requerimientos específicos a la UNAM que no son satisfechos por ninguna herramienta de la que se tenga noticia. Esto condujo al autor de esta tesis la necesidad de desarrollar herramientas especiales, que llenen los huecos que quedan al utilizar otras herramientas de seguridad.

Este capítulo describe el trabajo realizado en este sentido durante el proyecto.

### 10.1 La problemática

Los principales problemas que se han detectado con la utilización de las herramientas de seguridad ya existentes se puede resumir en los siguientes puntos:

1. Para lograr un buen esquema de seguridad, es necesario utilizar varias herramientas, cada una de las cuales se especializa en una necesidad particular.
2. Cada una de dichas herramientas genera información de forma separada, y en formatos diferentes.
3. Para tener un panorama completo de lo que esta sucediendo, el administrador tiene que revisar múltiples reportes y bitácoras generadas por las herramientas, a veces a lo largo de un período de tiempo.
4. La conjunción entre la información generada por dichas herramientas y que pueda tener relación entre sí tiene que ser hecha “a mano” por el administrador.
5. Toda la información esta en inglés. Aunque el inglés es el idioma universal de la computación, sigue siendo una barrera para muchos administradores de sistemas Unix en México.

Estos problemas pueden evitar el correcto aprovechamiento de las herramientas de seguridad, e incluso su utilización. Por lo tanto, es importante atacarlos dentro de este proyecto.

## 10.2 La solución

Para atacar los problemas descritos, es necesaria la existencia de herramientas que realicen de forma automática la conjunción, interrelación y análisis de la información, y la presenten al administrador en un formato entendible y en su propio idioma.

Con el fin de cubrir esta necesidad, el autor de esta tesis ha trabajado en el desarrollo de dos herramientas, llamadas **New CARP** y **SAINT**. Cabe mencionar que estas herramientas se hacen disponibles no solamente a la comunidad de cómputo de la UNAM, sino que se pretende ponerlas a disposición de la comunidad internacional a través de Internet. Es por esta razón que los nombres de los programas están en idioma inglés, la “lengua universal” en cómputo, y mucho mas en Internet.

## 10.3 New CARP

### 10.3.1 Antecedente: COPS y CARP

Una de las herramientas de seguridad principales utilizadas durante este proyecto es el **COPS**. Este programa genera reportes que constan de mensajes cortos que indican los problemas encontrados, como se puede apreciar en la figura 10.1.

Estos reportes pueden resultar bastante crípticos para quien no esta familiarizado con la utilización del programa, y tampoco sugieren posibles soluciones para los problemas encontrados. En la distribución de **COPS** se incluye un archivo en el que se explica el significado de cada uno de los mensajes generados, pero se ha visto en la practica que el consultar dicho archivo mientras se lee el reporte resulta incómodo, y la gran mayoría de los administradores no lo hace, prefiriendo hacer caso omiso de los mensajes que no entiende, con lo cual se pierde el objetivo inicial del **COPS**, que es alertar a los administradores sobre posibles problemas de seguridad.

Para solucionar en parte este problema, en la distribución de **COPS** se incluye un programa llamado **CARP** (*COPS Analysis and Report Program*) que genera, con base en los reportes generados por el **COPS**, una tabla que resume los problemas encontrados por **COPS** en varias maquinas, y que contiene la siguiente información (cada renglón de la tabla contiene la información correspondiente a una maquina):

- Nombre de la maquina. **CARP** tiene la capacidad de almacenar los reportes generados por **COPS** en varias maquinas, siempre y cuando esos reportes sean concentrados en una sola de ellas.
- Los problemas de seguridad encontrados por **COPS** en dicha maquina. Para cada problema, se indica su gravedad en uno de los siguientes niveles:
  - 0: Problemas muy graves, que permiten a un intruso acceso inmediato a la maquina.

```
ATTENTION:
Security Report for Tue Jan 24 05:33:46 PST 1995
from host ds5000

**** root.chk ****
**** dev.chk ****
Warning! /dev/dsk/dks0d3s3 is _World_readable!
**** is_able.chk ****
Warning! /usr/lib/aliases.pag is _World_writable!
**** rc.chk ****
**** cron.chk ****
**** group.chk ****
**** home.chk ****
Warning! User nobody's home directory /dev/null is not a directory! (mode 02066
6)
Warning! User fred's home directory /usr/users/claudia is mode 0777!
Warning! User tmpu's home directory /tmpu is mode 01777!
**** passwd.chk ****
Warning! Password file, line 2, user sysadm has uid = 0 and is not root
sysadm:*:0:0:System V Administration:/usr/admin:/bin/sh
Warning! Password file, line 3, user diag has uid = 0 and is not root
diag:*:0:996:Hardware Diagnostics:/usr/diags:/bin/csh
Warning! Password file, line 16, negative user id:
nobody:*:-2:-2:/:dev/null:/dev/null
**** user.chk ****
Warning! User jim: /usr/users/jim/.netrc is readable; mode 0755!
**** misc.chk ****
**** ftp.chk ****
Warning! /etc/ftpusers should exist!
**** pass.chk ****
**** kuang ****
**** bug.chk ****
Warning! /usr/lib/sendmail could have a hole/bug! (CA-88:01)
Warning! /bin/login could have a hole/bug! (CA-89:01)
Warning! /usr/etc/ftpd could have a hole/bug! (CA-89:01)
Warning! /usr/etc/fingerd could have a hole/bug! (CA-89:01)
```

Figura 10.1: Fragmento de un reporte generado por COPS

hostname	rep date	crn	dev	ftp	grp	hme	is	pass	msc	pwd	rc	root	usr	kng
polaris	1995_Jan_24		2	2		1	2			2				
mira	1994_Oct_12		2					2	2	2				
hardy	1994_Jul_25			2						2		2		
escher	1995_Feb_13		2						2				1	
ds5000	1995_Apr_29			1					2		1		2	
deneb	1995_Jan_07									2				
capella	1994_Aug_18		2	2		2			2	2				
andromeda	1995_Apr_20		2	2		1	2	2	2	2	1	2	1	

Figura 10.2: Ejemplo de tabla generada por CARP

- 1: Problemas que pueden ocasionar accesos no autorizados al sistema, pero que no son demasiado peligrosos.
- 2: Situaciones anormales cuya gravedad no se puede evaluar, pero que deben ser investigadas por el administrador.
- Sin número: Sin problemas detectados.

La figura 10.2 muestra un ejemplo de una tabla emitida por CARP. Las columnas *crn*, *dev*, etc. corresponden a cada una de las pruebas realizadas por COPS, y los números en cada columna indican la gravedad de los problemas encontrados en esa prueba: 0—Un problema que puede dar acceso inmediato a la clave de **root**, 1—un problema que puede dar acceso al sistema en una clave de usuario, 2—un problema cuya gravedad no puede ser determinada por CARP.

COPS y CARP se complementan en dos aspectos (descripción de los problemas y gravedad de los mismos), pero dejan sin cubrir un punto importante: proporcionar una descripción detallada de los problemas y sugerir posibles soluciones.

### 10.3.2 Una versión mejorada de CARP

New CARP (NCARP) es la solución propuesta por el autor de esta tesis a los problemas planteados. Se trata de un programa que, con base también en los reportes generados por COPS, genera la siguiente información:

1. La misma tabla generada por CARP, indicando la gravedad de los problemas encontrados en cada maquina (fig. 10.2).
2. Un reporte detallado, por maquina, de los problemas encontrados, que a su vez incluye, para cada problema, la siguiente información:
  - (a) El mensaje generado por COPS.
  - (b) Una descripción en español del problema y cuales pueden ser sus consecuencias.
  - (c) Descripción de posibles soluciones al problema, también en español.

La figura 10.3 muestra un ejemplo de un reporte generado por NCARP.

## Resumen de mensajes de COPS

```

hostname      rep date      crn dev ftp grp hme is pass msc pwd rc root usr kng
=====
polaris       1995_Jan_24 | | 2 | 2 | | 1 | 2 | | | 2 | | | | |
alpha1       1994_Jan_06 | | | 2 | 2 | 1 | 2 | 2 | 2 | 2 | | | | |

```

```

#####
polaris
#####

```

## MENSAJE GENERADO:

```
-----
```

Warning! /etc/ftpusers should exist!

## DESCRIPCIÓN DEL PROBLEMA:

```
-----
```

El archivo /etc/ftpusers indica que usuarios NO pueden entrar por medio de ftp a la maquina. Normalmente ninguna de las cuentas del sistema (root, bin, sys, appl, shutdown, etc.) hacen esta clase de accesos, por lo que es recomendable que dicho archivo exista y contenga los nombres de todas las cuentas del sistema, mas aquellas a las que no se desee dar servicio de ftp.

## SOLUCIÓN RECOMENDADA:

```
-----
```

Crear el archivo /etc/ftpusers, que contenga al menos lo siguiente:

```

root
bin
sys

```

...

```

#####
alpha1
#####

```

## MENSAJE GENERADO:

```
-----
```

Warning! Password Problem: Guessed: jsanche shell: /bin/csh

## DESCRIPCIÓN DEL PROBLEMA:

```
-----
```

El password de la cuenta mencionada ha sido adivinado. Solamente se menciona la cuenta, y no el password en cuestion. Esto es peligroso porque significa que el password de dicha cuenta es sencillo de adivinar, lo cual abre la posibilidad de que alguien que no sea el dueo de la cuenta pueda entrar a ella, comprometiendo así la seguridad del sistema.

## SOLUCIÓN RECOMENDADA:

```
-----
```

Los pasos a seguir son:

- 1) Verificar que la cuenta realmente este siendo usada por su dueo.
- 2) Si no es asi, desactivar o borrar la cuenta. Para desactivarla se tiene que poner un "\*" en el campo del password y un "/bin/false" en el campo del shell.
- 3) Si la cuenta si esta siendo utilizada, recomendarle al dueo de la cuenta que cambie su password inmediatamente.

...

Figura 10.3: Ejemplo de reporte generado por NCARP

Como se puede observar, en un mismo reporte se tiene toda la información que puede ser obtenida a partir de las pruebas realizadas por COPS en un sistema Unix. New CARP genera reportes individuales por maquina, que son almacenados en el disco y opcionalmente enviados por correo electrónico a alguna persona (presuntamente el administrador de la maquina), así como un reporte completo, que incluye la información sobre todas las maquinas analizadas.

Para detalles sobre la utilización del New CARP, consultar el apéndice J. Las dos secciones siguientes describen el diseño de NCARP.

### 10.3.3 Estructura de CARP

NCARP esta basado completamente en CARP, por lo que entender el funcionamiento de éste fue el primer paso en la elaboración de NCARP.

CARP esta compuesto principalmente por los siguientes componentes:

1. Un programa llamado `carp.anlz`, escrito en **AWK**, que analiza el reporte generado por COPS en una maquina, y produce como salida datos a partir de los cuales es posible construir un renglón de la tabla final producida por CARP (ver fig. 10.2).
2. Un programa llamado `carp.table`, también escrito en **AWK**, que toma el resultado generado por `carp.anlz`, y produce el renglón apropiado de la tabla.
3. Un programa llamado `carp`, escrito en **Bourne Shell**, que coordina la ejecución de los dos anteriores para generar la tabla completa, correspondiente a varias maquinas.

A continuación de describen con detalle cada uno de estos módulos.

#### Módulo de análisis (`carp.anlz`)

Este programa analiza, una por una, las líneas de un reporte de COPS que se le proporciona (ver figura 10.1 para un ejemplo de este tipo de reporte). Para cada línea, puede tomar una de tres acciones:

1. Detectar el inicio de una nueva sección del reporte, cuando detecta una línea que tiene el siguiente formato:

```
**** módulo ****
```

Donde *módulo* representa el nombre del módulo de COPS que detectó el problema.

Esta información es importante porque, como se puede ver en la figura 10.2, CARP menciona, en la tabla que produce, qué módulo de COPS detectó cada problema.

2. Detectar un posible problema, y generar la salida correspondiente. Esto se hace de acuerdo a expresiones regulares encontradas en el mensaje generado por COPS, y que identifican algún tipo de problema. Esto se basa en el hecho de que, para el mismo tipo de problema, COPS siempre produce un mensaje con la misma estructura general. Cuando se detecta una línea de este tipo, se imprime lo siguiente a la salida estandar del programa:

- El nombre del archivo que esta siendo revisado.
- El módulo de COPS en el que se produjo el problema.
- La gravedad del problema, de acuerdo a los números indicados anteriormente.

Por ejemplo:

```
# Un problema de nivel 0:
/A "+" entry in/ {print FILENAME, check, testing "0"; next }
# Un problema de nivel 1:
/User.*home directory.*is mode/ {print FILENAME, check, testing "1"; next }
# Un problema de nivel 2:
/is _World_ writable!/ {print FILENAME, check, testing "2" }
```

La primera regla detecta líneas del reporte de COPS que contengan la cadena **A "+"entry in**, y lo clasifica como un problema de nivel 0 (acceso inmediato a **root**). Con esto se detecta el problema de encontrar un signo "+" en el archivo `/etc/hosts.equiv` o en algún archivo `.rhosts`.

La segunda línea detecta problemas con los permisos de acceso en los directorios *home* de los usuarios, clasificándolo como un problema de tipo 1. La tercera línea detecta archivos y/o directorios que tengan permisos de escritura para todos los usuarios. Esto es una condición bastante inusual en un sistema Unix, por lo que se reporta como un problema de tipo 2.

Para cada problema detectado, se imprime la información mencionada arriba, en campos separados por espacios (la variable *testing* tiene propósitos de depuración del programa, y normalmente tiene un valor nulo, por lo que no aparece en la salida).

El orden en el que aparezcan las condiciones dentro de `carp.anlz` es muy importante, pues si una línea de un reporte de COPS coincide con mas de una condición, solamente se reportara de acuerdo a la primera de ellas. Esto permite también reportar problemas genéricos, pero al mismo tiempo detectar condiciones específicas que se quieran monitorear.

3. Detectar una condición reportada por COPS que no representa un problema, e ignorar la línea. Estas condiciones también funcionan en base a expresiones regulares, pero no generan ninguna salida. Por ejemplo:

```
/Warning!  \usr\spool\mail is _World_ writable!/ {next}
```

Esta línea detecta la condición de que el directorio `/usr/spool/mail` tenga permisos de escritura para todos los usuarios. En muchos sistemas Unix, esta es su condición normal, de manera que el mensaje puede ser ignorado.

Una vez más, el orden en el que aparezcan las líneas es importante. La línea anterior tiene que estar colocada antes de la siguiente:

```
/is _world_writable!/ {print FILENAME, check, testing "2" }
```

Esto es debido a que esta última representa el caso general, y la primera es un caso particular que se desea ignorar.

### Módulo de generación de la tabla (`carp.table`)

Este programa recibe la salida producida por `carp.anlz`, para un archivo de COPS completo, y la convierte en un renglón de la tabla final, correspondiente a una máquina. Los pasos que se siguen durante este proceso son:

1. Del nombre del archivo que se está revisando, se extrae el nombre de la máquina y la fecha del reporte. Esto es gracias a que COPS almacena sus reportes en directorios con el nombre de la máquina a la que corresponde cada uno de ellos, y dentro de esos directorios, en archivos cuyos nombres indican la fecha en la que fue creado. Por ejemplo: `ds5000/1995_Jan_05`.
2. La gravedad del problema se almacena en un arreglo indexado de acuerdo al nombre del módulo de COPS que lo detectó. Si el mismo módulo detectó varios problemas, se almacena el nivel de gravedad mayor.
3. Una vez que se ha recibido toda la salida de `carp.anlz`, se genera un renglón de salida con el formato apropiado, incluyendo en él toda la información recolectada en los dos pasos anteriores.

### Módulo principal (`carp`)

El programa `carp` es el que el usuario ejecuta para generar la tabla. Su función es ejecutar `carp.anlz` y `carp.table` repetidamente para generar la tabla completa, que contenga la información de todas las máquinas. Su funcionamiento se puede resumir como sigue:

1. Recibir como argumento el directorio debajo del cual se van a buscar los reportes que serán analizados.
2. Buscar, dentro de ese directorio, todos los subdirectorios que contengan archivos con nombre de la forma `aaaa_mmm_dd` (donde `aaaa` es el año, `mmm` es el mes y `dd` es el día), lo cual indica que esos subdirectorios contienen reportes de COPS.
3. Dentro de cada subdirectorio, buscar el reporte más reciente.
4. Imprimir el encabezado de la tabla.
5. Ejecutar `carp.anlz` y `carp.table` para cada reporte a analizar.

### 10.3.4 Diseño e implementación de NCARP

NCARP esta basado completamente en CARP, por lo que su diseño se basa en modificaciones hechas a este programa.

Las tareas que realiza NCARP adicionalmente a las realizadas por CARP son:

- Generación de mensajes descriptivos para cada problema encontrado.
- Generación de un reporte global conteniendo los resultados de todas las maquinas.
- Generación de reportes independientes para cada maquina, que son almacenados en disco y opcionalmente enviados por correo electrónico a personas especificadas en un archivo de configuración.

A continuación se describen los elementos que permiten realizar estas tareas, y cómo fueron integrados a CARP.

#### Base de datos de mensajes

El componente central de NCARP es una base de datos que contiene los mensajes explicativos para todos los problemas reportados por COPS. Cada registro de esta base de datos contiene:

- Un identificador del tipo de problema al que hace referencia.
- El texto explicativo del mensaje, que comúnmente consta de:
  - Una descripción detallada del problema y sus consecuencias.
  - Soluciones recomendadas al mismo.

Puesto que la versión original de CARP esta escrita principalmente en **AWK**, las extensiones hechas en NCARP también están realizadas en este lenguaje. Además, **AWK** es un lenguaje orientado al procesamiento de archivos de texto, lo que lo hace particularmente apropiado para procesar la base de datos, que no contiene mas que texto.

Por facilidad de procesamiento, los dos elementos de cada registro de la base de datos deben ser claramente identificables al ser leídos. Arbitrariamente, el autor eligió para la base de datos el formato ilustrado en la figura 10.4. Cada registro comienza con una línea que contiene 3 asteriscos (\*), después el identificador del mensaje, terminando con otros 3 asteriscos. La última línea del archivo debe contener exactamente tres asteriscos. El final de cada registro se reconoce por el inicio del siguiente o, en el caso del último registro, por la línea final del archivo.

El módulo de generación de reportes extrae de la base de datos la información correspondiente a cada tipo de problema en base a su identificador.

```
*** identificador 1 ***  
  
texto descriptivo 1  
  
*** identificador 2 ***  
  
texto descriptivo 2  
  
...  
  
***
```

Figura 10.4: Formato de la base de datos de mensajes de NCARP.

### Módulo de análisis (*ncarp.anlz*)

Este módulo tiene la misma estructura general de *carp.anlz*, es decir, esta compuesta por líneas que identifican expresiones regulares dentro del reporte generado por COPS, y que toman acciones de acuerdo al tipo de problema identificado. Sin embargo, las acciones tomadas se han extendido a las siguientes:

1. Evita interpretar líneas al inicio del archivo que no correspondan al reporte de COPS. Con esto se abre la posibilidad de analizar reportes que hayan sido enviados por correo electrónico, proceso en el cual se le añaden al archivo encabezados que son utilizados por el sistema de correo electrónico de Unix, pero que no forman parte del reporte que se va a procesar.
2. Imprimir el nombre del archivo que se está analizando, el módulo de COPS que detectó el problema, y la gravedad del mismo.
3. Imprimir el mensaje descriptivo apropiado para el problema encontrado.

El segundo punto funciona exactamente igual que en el programa *CARP* original, pero el tercero es el que le añade a *NCARP* su funcionalidad adicional.

Un problema inmediato que se observa al tener esta funcionalidad añadida es que ahora se necesitan producir dos resultados distintos, y que no pueden ser mezclados entre sí: el segundo punto proporciona la información para generar la tabla, mientras que el tercero proporciona la información para generar los reportes detallados de cada problema. Por esta razón, el autor decidió añadir al programa *ncarp.anlz* un nuevo argumento, en el que se especifica el nombre del archivo en el que serán depositados los mensajes descriptivos generados. Los datos generados por el segundo punto se siguen imprimiendo a la salida estándar, al igual que en *CARP*.

La generación de los mensajes descriptivos se implementa mediante una nueva función de *AWK* definida dentro de *ncarp.anlz*, llamada *print\_msg()*. Esta función es eje-

cutada por la acción asociada a cada tipo de problema, y recibe como argumento el identificador del mismo.

La función *print\_msg()* ha tenido dos versiones muy diferentes entre sí. En su versión original, esta función realizaba lo siguiente:

1. Recibir como argumento el identificador del tipo de problema deseado.
2. Buscar en la base de datos de descripciones el registro correspondiente al identificador proporcionado.
3. Imprimir al archivo de resultados la línea del reporte de COPS que había sido identificada, seguida por la descripción de ese problema extraída de la base de datos.

Aunque este algoritmo cumple con la función básica necesaria, en las primeras etapas de prueba del programa se observó que tiene dos deficiencias importantes:

1. En muchas ocasiones, COPS detecta, dentro de la misma maquina, varios errores del mismo tipo. Al ejecutarse *print\_msg()* para cada uno de ellos, se incluía en el mismo reporte varias veces la misma descripción, lo cual contribuye a confundir a quien lo lee.
2. Algunos de los mensajes generados por COPS constan de dos líneas y no solamente de una, como se contempló en el diseño original de *print\_msg()*. Al encontrarse con este tipo de mensajes, *print\_msg()* generaba información incompleta, y era incapaz de procesar correctamente la segunda línea del mensaje.

Esto condujo al diseño de una nueva versión de la función *print\_msg()*. En ésta, se requiere de una etapa de post-procesamiento al final del programa, pero se corrigen los dos problemas mencionados arriba. El funcionamiento de *print\_msg()* bajo este nuevo esquema es como sigue:

1. Se recibe como argumento el identificador del problema cuya descripción se desea, y una bandera indicando si el mensaje de COPS es de una o de dos líneas.
2. Se almacena el mensaje completo (leyendo la siguiente línea del reporte de COPS, en caso de que sea necesario), conjuntamente con su identificador, en un arreglo en la memoria del sistema. En este punto todavía no se imprime nada al archivo de resultados.

Al final del programa (cuando se han procesado ya todas las líneas del reporte de COPS) se realiza lo siguiente:

1. Se toma el primer identificador almacenado en el arreglo por la función *print\_msg()*.
2. Se imprimen al archivo de resultados todos los mensajes correspondientes a ese identificador que hayan sido encontrados en el reporte de COPS.
3. Se busca en la base de datos el registro correspondiente a dicho identificador.

4. Se imprime la descripción apropiada al archivo de resultados, a continuación de los mensajes.
5. Se repite desde el punto 2 para todos los identificadores almacenados en el arreglo.

Con esto se logra que todos los mensajes de un mismo tipo queden agrupados en el reporte final, con una sola descripción asociada a todos ellos.

En la implementación de este algoritmo se hace uso de las características especiales de **AWK**, como la posibilidad de manejar arreglos con índices alfanuméricos. De hecho, el arreglo en el que se almacenan los mensajes generados por **COPS** se indexa de acuerdo al identificador de los mensajes. Esto permite agrupar todos los mensajes de un mismo tipo en el mismo elemento del arreglo, y posteriormente acceder a ellos de forma fácil.

#### **Módulo de generación de la tabla (ncarp.table)**

Este módulo es exactamente igual al de **CARP**, pues su funcionalidad no cambia. Recibe los mismos datos de entrada, y en base a ellos produce un renglón de la tabla que resume los problemas encontrados.

#### **Módulo principal (ncarp)**

Este es el programa que el usuario ejecuta. Coordina las actividades de los otros dos módulos, y manipula la salida producida por éstos para generar los reportes finales. Su funcionalidad, por lo tanto, es considerablemente mayor que la del programa **carp** original. El algoritmo que sigue este programa para generar todos los reportes necesarios es el siguiente:

1. Recibir como argumento el nombre del directorio debajo del cual están almacenados los reportes de **COPS**.
2. Opcionalmente, recibir el nombre del archivo de configuración (en el que se indica a quién se le enviara el reporte de cada maquina analizada) y el directorio debajo del cual se almacenaran los reportes individuales generados por **NCARP**.
3. Buscar, debajo del directorio de **COPS** especificado, todos los subdirectorios que contengan reportes.
4. En cada uno de los subdirectorios encontrados, buscar el reporte de **COPS** mas reciente.
5. Crear un archivo temporal vacío, al que llamaremos **TMP\_FILE2** en esta descripción.
6. Para cada uno de los reportes a analizar, realizar lo siguiente:
  - (a) Crear dos archivos temporales vacíos, a los que llamaremos **TMP\_RPT** y **TMP\_FILE** en esta descripción.

```

maquina1          email1[,email2,...]
maquina2          email3[,email4,...]
...

```

Figura 10.5: Formato del archivo de configuración de NCARP.

- (b) Ejecutar `ncarp.anlz` sobre el archivo, y pasarle su salida estándar a `ncarp.table`. Se le indica a `ncarp.anlz` que deposite las descripciones generadas en `TMP_FILE`, y la salida de `ncarp.table` se almacena en `TMP_RPT`, además de imprimirse en la salida estándar.
  - (c) El contenido de `TMP_FILE` se añade al contenido de `TMP_FILE2`. De esta forma, en este archivo se van acumulando las descripciones de problemas generadas para todas las maquinas analizadas.
  - (d) El contenido de `TMP_FILE` también se añade al contenido de `TMP_RPT`. De esta forma, en este archivo queda el reporte individual completo de la maquina que se esta analizando, incluyendo el renglón correspondiente de la tabla y las descripciones de los problemas encontrados.
  - (e) Comparar `TMP_FILE` con el último reporte generado anteriormente para esa maquina. Solamente si son diferentes se procede con los puntos siguientes. De esta forma se evita que se almacenen o envíen por correo electrónico reportes repetidos.
  - (f) Almacenar `TMP_FILE` debajo del directorio de reportes de NCARP, en un directorio con el nombre de la maquina analizada, y en un archivo con nombre `ncarp.yyyy_mmm_dd`, donde `yyyy` (numérico), `mmm` (alfabético) y `dd` (numérico) corresponden al año, mes y día en que se generó el reporte, respectivamente.
  - (g) Analizar el archivo de configuración de NCARP para ver si se debe enviar el reporte a alguien. Si es así, realizar el envío.
7. Imprimir `TMP_FILE2`. Con esto se imprimen, después de la tabla completa, los mensajes descriptivos de todas las maquinas analizadas.

#### Archivo de configuración (`ncarp.mail`)

En este archivo se especifica a quién o quiénes se va a enviar, por correo electrónico, el reporte individual generado para cada maquina. El archivo tiene un formato muy sencillo, como se observa en la figura 10.5. Cada línea contiene la información correspondiente a una maquina, en dos columnas: la primera columna contiene el nombre de la maquina, y la segunda contiene una o mas direcciones electrónicas, separadas por

comas o espacios en blanco. El programa `ncarp` revisa este archivo, y los reportes individuales de aquellas maquinas para las que exista un registro son enviados por correo electrónico a las direcciones especificadas. Los reportes de maquinas que no aparezcan en el archivo de configuración solamente son almacenados en el disco.

Este archivo puede incluir comentarios en cualquier línea, comenzando con el carácter “#”. Todo lo que está entre este carácter y el final de la línea sera ignorado. Las líneas en blanco también se ignoran.

Esto completa la funcionalidad de **NCARP**: hacer llegar los reportes a las personas apropiadas. **NCARP** es ya una herramienta utilizable, aunque siempre se considera lo que puede —y debe— suceder con ella a futuro.

### 10.3.5 El futuro de **NCARP**

En términos generales, **NCARP** se puede considerar como una herramienta terminada, a excepción de corrección de errores que se vayan detectando. Sin embargo, hay dos características que es necesario añadir para hacer su utilización completamente confiable y factible a nivel internacional.

Un problema que se presenta es que el correo electrónico que viaja a través de la red es relativamente fácil de interceptar. El enviar reportes de seguridad por correo electrónico puede resultar, por esta razón, muy peligroso. Los reportes pueden contener información sobre problemas de seguridad en las maquinas, y si caen en manos no autorizadas, esta información puede utilizarse para aprovechar esos huecos en vez de corregirlos, como es su propósito original.

La solución mas efectiva a este problema es la utilización de mecanismos de cifrado de datos para el envío de los reportes. De esta forma, **NCARP** enviaría los reportes cifrados de tal manera que solamente el destinatario autorizado (en este caso, el administrador del sistema) pueda descifrarlo y ver su contenido.

El programa mas utilizado para llevar a cabo cifrado de datos es **PGP** [Zim94, Gar95]. Este programa permite enviar por correo electrónico mensajes cifrados de tal manera que no puedan ser leídos mas que por el destinatario autorizado, y que no puedan tampoco ser modificados sin autorización o falsificados.

En este momento, el autor se encuentra trabajando para dotar a **NCARP** de mecanismos que permitan la utilización de **PGP** para cifrar los reportes que son enviados a través de correo electrónico.

El otro problema es que, en su versión actual, **NCARP** genera resultados únicamente en español. Aunque esto es parte de su propósito original, hace difícil su utilización por parte de personas que no sean de habla hispana. Por esto, otra tarea en la que el autor esta trabajando en este momento es añadir a **NCARP** soporte para la generación de reportes en distintos idiomas.

## 10.4 SAINT: Una herramienta de integración de análisis de seguridad

### 10.4.1 Antecedente: utilización de herramientas de seguridad

El esquema de seguridad básico implementado en DGSCA incluye la utilización de las siguientes herramientas:

- COPS
- TCP-Wrapper
- passwd+
- Crack
- TripWire

La utilización de estas herramientas presenta el mismo problema ya mencionado: todas ellas generan información que en algunos casos puede estar relacionada, pero no existe forma de integrar los diferentes reportes, como no sea mediante un análisis “manual” de los mismos por parte de un ser humano.

### 10.4.2 Una nueva forma de ver los datos: SAINT

Por esta razón, se pensó en el diseño de SAINT (*Security Analysis INtegrator Tool*): un programa altamente configurable y extensible, que integre, en su forma inicial, los datos generados por las herramientas mencionadas, pero que pueda ser ampliado para analizar la información generada por cualquier número de herramientas.

### 10.4.3 Estado actual de SAINT

Al momento de finalizarse la escritura de esta tesis, SAINT se encuentra todavía en una etapa de desarrollo. Se espera la liberación de la primera versión en un futuro cercano. Esta herramienta se encontrará disponible a través de FTP anónimo en `ftp://ftp.super.unam.mx/pub/security/tools/saint.tar.gz`.

### 10.4.4 Funciones y características de SAINT

Las principales funciones son:

1. Análisis cruzado de los reportes generados por diferentes herramientas en varias máquinas, para detectar patrones de acción que puedan indicar problemas o accesos ilegales al sistema.
2. Estimación, con base en la misma información, de las posibles causas de un problema.

3. Generación de advertencias cuando se detecten problemas graves o accesos ilegales al sistema.
4. Sugerencia de posibles soluciones a los problemas encontrados.
5. Presentación de toda la información en un reporte en español.

Sus principales características son:

**Extensible** Es fácil añadir nuevos módulos al programa que sirvan para analizar e integrar la información generada por otras herramientas, o para añadir nuevas formas de análisis. Esto se logra mediante la modularización del programa.

**Configurable** Fácilmente se puede modificar el comportamiento de SAINT frente a problemas encontrados, así como su percepción de lo que son los problemas. Esto es necesario, puesto que una situación alarmante para alguna organización puede ser parte normal de las políticas de seguridad para otra, dada la subjetividad de la seguridad en cómputo.

**Fácil de utilizar** Una vez instalado, SAINT es prácticamente transparente a los administradores, a no ser por los reportes que genera periódicamente. Se contempla que la ejecución de SAINT esté sincronizada con la de las otras herramientas de seguridad, de manera que siempre actúe sobre los últimos reportes generados de forma automática.

**Portable** En estos días en los que la norma es la heterogeneidad de los ambientes de cómputo, es muy importante contar con herramientas que puedan utilizarse con el menor número de problemas en todo tipo de plataformas. La mayor parte de SAINT esta escrito en **perl** [WS92], un lenguaje cuya utilización es cada vez mas común en sistemas Unix, y que es prácticamente estándar, asegurando casi completamente la compatibilidad entre diferentes sistemas operativos.

Para detalles sobre la utilización de este programa, consultar el apéndice K.

Las siguientes secciones describen algunos aspectos involucrados en el diseño de SAINT.

#### 10.4.5 ¿Qué hace SAINT?

La operación de SAINT puede dividirse en cuatro grandes etapas:

1. Recolección y homogeneización de datos.
2. Ordenamiento de los eventos registrados.
3. Análisis de los eventos.
4. Presentación de resultados.

A continuación se describe cada una de estas etapas en detalle.

### 10.4.6 Recolección y homogeneización de datos

SAINT necesita interpretar información producida por múltiples herramientas, posiblemente en múltiples máquinas. Lo más común es que cada herramienta almacene los resultados que produce en archivos diferentes, y con formatos diferentes.

Para facilitar el procesamiento posterior, la primera etapa en la ejecución de SAINT consiste en leer los resultados de las diferentes herramientas y archivos que se van a monitorear, y convertirlos a un formato común que pueda ser interpretado por las etapas posteriores del programa.

Estas tareas presentan varios problemas:

1. La información producida por cada herramienta proviene de diferentes archivos, y esta en diferentes formatos. Ciertos datos incluso pueden requerir algún preprocesamiento antes de poder ser útiles. El proceso de conversión al formato común será diferente para cada tipo de datos.
2. Los formatos de datos utilizados por alguna herramienta pueden cambiar en versiones futuras de la misma.
3. Debe ser posible añadir soporte para nuevas herramientas (o nuevas versiones de las mismas) sin necesidad de realizar modificaciones a SAINT.
4. El formato común debe contener toda la información que pueda ser útil para su procesamiento posterior.

El análisis muestra que estos problemas pueden ser resueltos mediante las siguientes dos técnicas:

#### Modularización

Los primeros 3 puntos pueden ser resueltos mediante una técnica muy sencilla: utilizar módulos diferentes para cada herramienta. Estos módulos son programas independientes, que son utilizados por SAINT para realizar el procesamiento de los datos correspondientes a cada herramienta.

Al utilizar programas independientes, se tienen las siguientes ventajas:

- Cada módulo está especializado en procesar un solo tipo de datos, por lo que el código para hacerlo puede ser muy sencillo.
- Cada módulo puede estar escrito en el lenguaje más apropiado de acuerdo al tipo de procesamiento que se tenga que realizar. Lo único que se tiene que definir es la interfaz que va a utilizar SAINT para comunicarse con todos los módulos de manera consistente.
- Cada módulo puede obtener los datos de donde sea necesario, de acuerdo a la herramienta que los haya generado. Puede ser que estén almacenados en un archivo en la misma máquina en la que se ejecuta SAINT, o en una diferente, lo cual haría necesario establecer una conexión de red para obtenerlos. En cualquier caso, esto es transparente para SAINT, pues el módulo de obtención de datos se encarga de eso.

- Los módulos pueden ser reemplazados o modificados para tratar con nuevas versiones de las herramientas, corregir errores en el procesamiento, o realizarlo de manera mas eficiente.
- Resulta sencillo añadir soporte para nuevas herramientas, pues basta escribir el módulo apropiado e indicarle a SAINT que lo utilice.

Los módulos a utilizar se especifican en el archivo de configuración, `saint.cf`, descrito posteriormente.

### Diseño del formato común de datos

La mejor forma de resolver el problema planteado en el punto 4 es diseñar bien desde el principio el formato común de datos. Todos los módulos de recolección tienen que producir como salida los mismos datos que reciben, pero en un formato común que pueda ser entendido por las etapas posteriores del programa.

Para evitar tener que realizar modificaciones posteriores a este formato, se tiene que poner especial cuidado en la información que debe contener, de manera que cumpla con las siguientes características:

**Completo:** Que incluya toda la información necesaria para analizar los datos.

**Extensible:** Que permita cubrir cualquier tipo de datos, de manera que la aparición de nuevas herramientas en cuyos formatos no se haya pensado inicialmente no implique un cambio en el formato común.

**Sencillo:** Que sea fácilmente procesable de forma automática. Esto implica que sus distintos campos sean fácilmente distinguibles, y que el contenido de cada uno de ellos pertenezca, hasta donde sea posible, a un conjunto de datos finito y predecible.

Con base en estas consideraciones, el autor decidió que cada registro del formato común de datos debe contener los siguientes campos:

**Tipo de evento.** Una “palabra clave” que permite clasificar al evento. Éstas son asignadas arbitrariamente, de acuerdo a los tipos de eventos que sea necesario identificar. En el diseño original de SAINT se tienen los siguientes tipos de eventos:

*telnet* Establecimiento de sesiones interactivas remotas.

*ftp* Establecimiento de sesiones de FTP.

*rlogin* Establecimiento de una conexión utilizando el comando **rlogin**.

*rsh* Ejecución de un comando remoto utilizando el comando **rsh**.

*su\_root* Ejecución del comando **su** para obtener acceso a la clave de **root**.

*su\_user* Ejecución del comando **su** para obtener acceso a alguna clave que no sea **root**.

*reboot* Una reinicialización del sistema.

Estos tipos de eventos permiten acelerar el procesamiento posterior de los datos al asignarle una categoría, que puede ser utilizada para decidir qué método de procesamiento es el ideal para ese registro.

**Fecha y hora en que ocurrió el evento.** Para poder hacer un análisis cronológico de los eventos ocurridos, que permita detectar eventos relacionados, es importante que cada evento tenga una etiqueta de tiempo, siempre que sea posible.

**Maquina que originó el evento.** Cuando se trata de eventos que involucran un acceso a través de la red, es importante identificar su origen para poder realizar verificaciones de validez del acceso en cuestión.

**Maquina que recibió el evento.** También es importante, en el caso de eventos de red, qué maquina fue la destinataria del evento registrado (por ejemplo, en el caso de una sesión remota).

**Usuario que generó el evento.** Para algunos tipos de eventos, es posible determinar qué usuario lo generó, y esta información puede ser útil también en su análisis. Por ejemplo, en algunos casos es posible averiguar qué usuario inició una sesión interactiva remota.

**Usuario que recibió el evento.** También puede ser útil saber qué usuario fue el destinatario del evento registrado, en caso de que este dato sea aplicable. Por ejemplo, en un evento de tipo *su\_user*, es posible averiguar a qué cuenta se realizó el cambio.

**Campo de uso general.** Se consideró conveniente la existencia de un campo cuyo significado dependa del tipo de evento que se esté registrando. Algunos ejemplos de lo que este campo puede contener, dependiendo del tipo de evento, son:

- Identificador del proceso que proporcionó un servicio.
- Bandera que indique si un acceso a través de la red fue exitoso o no.
- Valores de variables de ambiente que afecten el resultado del evento.
- Nombre de un archivo transferido por FTP.
- Comando ejecutado con **rsh**.
- Terminal en la que se estableció una sesión remota, o desde la que se originó.
- La razón por la que se reinicializó el sistema.

**Mensaje original.** Es imposible prevenir y clasificar toda la información que en un momento dado pudiera ser necesaria para analizar un evento registrado. Por esta razón, se consideró incluir en el formato común el mensaje original mediante el que se detectó el evento. Esto permite realizar sobre este campo cualquier análisis posterior que no haya sido previsto.

El formato que el autor decidió utilizar para cada registro es el siguiente:

```
tipo|fecha_hora|sys_orig|sys_dest|usr_orig|usr_dest|gral|msg
```

Donde los campos aparecen en el orden en el que fueron mencionados anteriormente.

### 10.4.7 Ordenamiento de los eventos

El análisis que se realiza sobre los eventos es necesariamente cronológico, pues ésto es lo que permite detectar las relaciones existentes entre ellos. Por esta razón, una vez que se han recolectado los datos correspondientes a todos los eventos registrados, estos datos son ordenados de acuerdo a sus etiquetas de tiempo (el segundo campo del formato común de datos).

También es importante hacer mención de un posible problema: si los relojes de todas las maquinas que se estén monitoreando no están correctamente sincronizados, SAINT podría registrar los eventos en un orden diferente al que realmente ocurrieron. Aunque no es un factor vital para el funcionamiento de SAINT, la desincronización de relojes, si es mayor a unos pocos segundos, puede afectar los resultados obtenidos.

### 10.4.8 Análisis de los eventos

Esta es la parte del programa que realiza el trabajo central de SAINT. En base a los datos recolectados y procesados en las dos etapas anteriores, es necesario realizar un análisis que permita detectar relaciones entre los eventos registrados, y decidir, hasta donde sea posible, si esos eventos forman parte de la operación válida de los sistemas o si representan posibles problemas.

Se consideraron dos posibles técnicas para realizar este análisis: análisis gramatical de los eventos y simulación de los mismos. A continuación se describen ambas técnicas, justificando la decisión final tomada.

#### Análisis gramatical

Esta técnica implica realizar un análisis gramatical semejante al que se realiza para reconocer un lenguaje de programación. En esta técnica se elaboraría una gramática que detectara secuencias de eventos relevantes, y realizara acciones con base en ellas. Cada evento se detectaría como un *token* (una unidad léxica), y la gramática reconocería secuencias de *tokens* que tuvieran algún significado especial.

Esta técnica se descartó por las siguientes razones:

- Las secuencias relevantes de eventos son demasiadas, y por lo tanto la gramática sería muy compleja.
- Añadir soporte para nuevas herramientas o nuevos problemas implicaría muy posiblemente modificaciones al código, tanto del analizador léxico (el que identificara los eventos como *tokens*) como del analizador gramatical (el que identifica las secuencias de eventos). Esto va en detrimento de la capacidad de expansibilidad planteada como objetivo de SAINT.
- En la vida real, se puede dar más de una secuencia relevante de eventos de forma simultánea. Esto significa que los eventos correspondientes a una de las secuencias pueden estar intercalados con los de alguna otra. Los tipos de gramáticas utilizados normalmente (por ejemplo, para el reconocimiento de lenguajes de programación) no fueron diseñados para reconocer secuencias “paralelas” *tokens*. Por esta razón, una gramática para hacerlo sería excesivamente compleja.

- Dos secuencias de eventos pueden coincidir en algunas secciones, y aún así ser diferentes. Para un analizador gramatical automático, sin embargo, estas coincidencias pueden hacer muy difícil o imposible la distinción entre una secuencia y otra.
- Dos secuencias idénticas de eventos pueden indicar cosas completamente distintas, dependiendo de otros factores no directamente indicados en la secuencia. Es decir, se necesita tener en cuenta el contexto en el que suceden los eventos. Esto resulta en una gramática no determinística, que es muy difícil de analizar.

### Simulación de los eventos

Esta técnica consiste en realizar una simulación, dentro del programa, de los eventos ocurridos en cada una de las máquinas, y de los efectos que cada uno de ellos va teniendo. En cierta forma esta técnica es similar a la mencionada anteriormente, pero las secuencias posibles de eventos no están determinadas previamente en una gramática, sino que solamente se especifica la forma en la que el sistema reacciona a cierto evento bajo ciertas circunstancias, y el orden en que estos eventos aparezcan puede ser completamente aleatorio.

Una simulación se basa en elementos que responden a ciertos estímulos, y que interactúan entre sí. En este caso, los elementos de la simulación serían los siguientes:

**Maquinas:** Son los elementos centrales, pues es donde se generan y reciben los eventos analizados. Cada máquina permanece en un cierto estado hasta que ocurre algún evento al que la máquina está programada para responder. En ese momento cambia el estado. En base al estado en el que se encuentra, una máquina puede tomar cierta acción (como reportar un posible problema).

**Dominios de seguridad:** Para el análisis de eventos relacionados con seguridad, es importante el concepto de dominios. Éstos se refieren a conjuntos de máquinas que confían entre sí, lo cual puede alterar en cierta forma su respuesta a eventos generados dentro del mismo dominio de seguridad.

**La red:** A través de la red se realizan las interacciones entre las distintas máquinas. En este caso, la red sirve solamente como un medio de transporte pasivo, pues no se considera que pueda generar eventos por sí misma.

Dado este planteamiento, la técnica de la simulación parece ser la apropiada para atacar el problema de analizar los eventos.

Todavía falta por definir el método utilizado para implementarla, pero éste parece sugerirse por sí mismo. Cada elemento que interviene en la simulación, ya sea una máquina o un dominio (la red no se cuenta por ser un elemento pasivo) es un objeto independiente, con ciertas características, y que en un momento dado se encuentra en un estado específico que determina sus respuestas posteriores. Los objetos responden a los eventos, que pueden generarse dentro del mismo objeto o en otros, lo cual indica que los objetos deben tener la capacidad de comunicarse entre sí.

Todo esto puede ser implementado de forma relativamente sencilla utilizando diseño orientado a objetos. En éste, cada objeto es una entidad independiente (tal como lo son las máquinas y los dominios de seguridad), que puede guardar un estado propio, responder a estímulos tanto internos como externos, y establecer comunicación con otros objetos mediante el paso de mensajes.

Por esta razón, el autor decidió utilizar técnicas y herramientas de programación orientadas a objetos para implementar la etapa de análisis de la información en SAINT. Un problema comúnmente asociado a la programación orientada a objetos es que las herramientas disponibles todavía no se encuentran en un estado de madurez que permita la completa portabilidad de los programas. Los lenguajes tradicionales de programación orientada a objetos (C++ [Str91], **C Objetivo** [CN91, PW91] y **Smalltalk** [GR89], por ejemplo) no han alcanzado el grado de madurez necesaria para garantizar la portabilidad. Aunque existen compiladores o intérpretes de estos lenguajes para la gran mayoría de las plataformas de cómputo, la falta de estandarización hace que existan diferencias que hacen que, en ocasiones, los programas escritos para un compilador no puedan ser ejecutados en otro, aunque se trate del mismo lenguaje.

Existe, sin embargo, un lenguaje de programación que cobra cada vez mas importancia en el mundo de los sistemas Unix (e incluso en otros sistemas operativos) como un lenguaje poderoso y, sobre todo, portable. Se trata de **perl** [WS92]. En su versión 5, este lenguaje incluye la capacidad de realizar programación orientada a objetos. Los programas escritos en **perl** son, en la gran mayoría de las ocasiones, completamente portables entre diferentes plataformas de cómputo. Además, **perl** es accesible sin costo, por lo que es ideal para su utilización masiva.

Por estas razones, el autor de esta tesis decidió utilizar **perl5** para la elaboración de la etapa de simulación de eventos en SAINT. De hecho, se utiliza este lenguaje en casi todos los módulos de la versión inicial de SAINT.

Se consideró que los dos tipos de objetos utilizados en la simulación deben ser los siguientes:

**Maquinas:** Estos objetos deben contener una representación de todos los factores que puedan afectar, en un momento dado, la seguridad del sistema, o formar parte de una secuencia relevante de eventos. El estado de un sistema Unix se puede resumir normalmente con base en los siguientes elementos:

- Sesiones establecidas, ya sea por medio de **telnet**, **ftp**, **rlogin** o **rsh**. Cada sesión tiene asociada información acerca del usuario al que pertenece, así como la terminal, fecha y hora en la que se estableció.
- Procesos existentes. No todos los procesos son relevantes para el estado del sistema en términos de análisis de eventos, pero sí se debe tener el registro de los que sean significativos. Cada proceso tiene asociada información acerca del usuario al que pertenece, la terminal a la que esta asociado, los privilegios con los que se ejecutó, y la fecha y hora de ejecución.
- Archivos. Una vez mas, no todos los archivos del sistema intervienen en la representación del estado del sistema, pero, de acuerdo a los eventos registrados, puede haber archivos que sean importantes para la detección de un problema. Cada archivo tiene asociada información acerca del usuario y

grupo al que pertenece, su tamaño, sus permisos de acceso, su ruta completa y su fecha y hora de última modificación.

Además, una maquina y su reacción a los eventos están definidos por las siguientes características:

- Nombre de la maquina.
- Maquinas y dominios confiables. Una maquina puede estar configurada para “confiar” en otras maquinas, o incluso en dominios completos. A estas maquinas y dominios confiables se les pueden otorgar ciertos permisos especiales, como el establecimiento de sesiones con privilegios especiales.
- Usuarios privilegiados. En toda maquina existe uno o mas usuario que tienen autorización para realizar tareas administrativas con privilegios especiales. La ejecución de este tipo de tareas por parte de un usuario que no esté registrado como privilegiado puede ser señal de actividad sospechosa.
- Horarios especiales. Por lo regular, existen horarios definidos en los que se realizan tareas fuera de lo común en el sistema. Estas tareas normalmente incluyen labores de mantenimiento, como reinicializar el sistema o ejecutar programas privilegiados de monitoreo o mantenimiento. La ejecución de estas tareas especiales fuera de los horarios definidos también puede ser señal de actividad sospechosa.
- Mecanismo de selección de eventos. Este es el “punto de entrada” de los eventos al análisis por parte de una maquina. Esta rutina, con base en las características del evento recibido, lo manda para su procesamiento a la rutina de reacción a eventos apropiada.
- Rutinas de reacción a eventos. Cada maquina posee, dentro de sí, la capacidad de reaccionar a los distintos eventos registrados. De acuerdo al tipo de evento, la reacción puede consistir en ignorarlo, modificar de alguna forma el estado de la maquina, o tomar alguna acción, como notificar de un posible problema o de cierta actividad sospechosa.

Estas rutinas, junto con el mecanismo de selección de eventos, son las que determinan los eventos a los que va a reaccionar una maquina, y por lo tanto los problemas que es capaz de detectar. Por lo tanto, para añadir a SAINT soporte para nuevos tipos de problemas o nuevas herramientas, es necesario añadir nueva rutinas de reacción, y nuevas reglas al mecanismo de selección. Aquí también es útil el utilizar un diseño orientado a objetos. Utilizando las posibilidades de herencia y definición dinámica de subrutinas, SAINT permite agregar o modificar las rutinas que sean necesarias para procesar el nuevo tipo de información.

**Dominios de seguridad:** El estado de un dominio de seguridad en un momento dado esta definido por los siguientes factores:

- Conexiones establecidas desde el exterior del dominio.

- Conexiones establecidas hacia el exterior del dominio.
- El estado de cada una de las maquinas que lo componen. Este estado es mantenido, en forma independiente, por los objetos que representan a dichas maquinas.

Además, un dominio mantiene información sobre las siguientes características:

- Nombre del dominio.
- Maquinas que lo componen. Cada maquina esta representada por un objeto del tipo descrito en la página anterior.

Una vez que están definidos los objetos, los pasos para llevar a cabo la simulación de los eventos son:

1. Crear los objetos que representan a todas las maquinas y dominios que intervienen en la simulación.
2. En orden cronológico, comunicarle a cada objeto los eventos en los que haya estado involucrado. Cada objeto reaccionara a los eventos de acuerdo a sus características, y al estado en el que se encuentre.

Cuando un objeto detecta un evento que considere relevante, debe producir una salida que sirva para la posterior etapa de presentación de la información. Para facilitar la tarea de dicha etapa, la salida producida por los objetos también debe estar en un formato común que permita su fácil interpretación. La información que debe estar contenida en este formato es la siguiente:

**Tipo de evento:** Se utiliza la misma clasificación que en el formato de registro de eventos, descrito en la pagina 102.

**Maquina que reporta el evento:** Cada elemento del reporte es producido por alguna maquina en particular, pues son las maquinas las que tienen la capacidad de detectar eventos relevantes.

**Gravedad del evento:** Cada evento reportado debe estar clasificado de acuerdo a su gravedad. Los niveles de gravedad definidos por el autor son:

- 0** Informativo, lo suficientemente notorio como para ser reportado, pero que no representa ningún problema en particular.
- 1** De advertencia. Un evento que se sale de lo común, pero que no indica un problema de forma directa.
- 2** De alerta. Un posible problema de seguridad, que debe ser investigado de inmediato.
- 3** De emergencia. Un claro problema de seguridad.

Debe observarse que estos niveles de gravedad son asignados por el objeto que reporta el evento, por lo que su determinación se deja a discreción de dicho objeto.

**Fecha y hora en que se produjo el evento:** Aunque los eventos ya están ordenados cronológicamente, esta información es importante para su posterior análisis.

**Resumen del evento:** Una cadena corta de caracteres generada por el objeto, y que describa al evento en resumen. Normalmente solo debe incluir el tipo de evento, y la maquina que lo generó.

**Mensaje descriptivo:** Cada objeto debe generar una cadena de caracteres que describa con detalle al evento, y que sera incluida en ciertos formatos del reporte final.

El formato elegido por el autor, de forma arbitraria, para cada registro de la salida es el siguiente:

```
tipo|maquina|gravedad|fecha_hora|resumen|descripción
```

### 10.4.9 Presentación de resultados

La última etapa en la ejecución de SAINT es la interpretación de la salida generada por los objetos, y su presentación en un reporte legible por un ser humano.

El tener la parte de presentación de la información en un módulo separado de la parte de generación de la misma hace que sea sencillo modificar la presentación de los resultados sin necesidad de modificar el funcionamiento del programa. Algunos de los posibles formatos de salida son:

1. Reportes ordenados cronológicamente.
2. Reportes ordenados de acuerdo a la gravedad de los eventos.
3. Reportes ordenados de acuerdo a la maquina en la que fueron registrados.
4. Reportes ordenados de acuerdo al tipo de eventos.
5. Reportes que incluyan solamente la información de ciertas maquinas.
6. Reportes que incluyan solamente la información de eventos ocurridos en un cierto intervalo de tiempo.
7. Reportes en formato de texto.
8. Reportes que sean enviados automáticamente a los administradores de las maquinas afectadas.
9. Reportes escritos en HTML, con características de hipertexto, para ser examinados con un visualizador de WWW.

Los primeros 7 puntos son incluidos en la primera versión de SAINT.

Es importante hacer notar que todos los filtros en la información que se presenta, como restringir los eventos por maquina o por intervalo de tiempo, son realizados en esta etapa, y no en la de recolección de los datos. Esto se consideró apropiado debido

a que las secuencias de eventos que se pueden dar entre un conjunto de maquinas son completamente impredecibles. Un evento que sea detectado como significativo puede ser el resultado final de una cadena de eventos que haya comenzado varias horas antes. Si los eventos se filtran antes de hacer el análisis, posiblemente el evento final nunca se detecte debido a estar incompleta, al momento del análisis, la secuencia que le dio origen.

#### 10.4.10 Módulo de control

Al igual que en NCARP, es necesaria la existencia de un módulo que controle las actividades de todos los demás, de manera que la información fluya de manera correcta y los resultados se generen apropiadamente.

La tarea de este módulo principal se ve facilitada considerablemente gracias al diseño altamente modular y orientado a objetos que tiene el programa. Sus actividades se pueden resumir en:

1. Interpretar los argumentos de línea de comandos.
2. Leer el archivo de configuración.
3. En base a la información recolectada en los puntos 1 y 2, fijar los parámetros y leer del disco la información necesaria (por ejemplo, para cargar módulos de análisis adicionales).
4. Ejecutar, una por una, los programas que implementan las diferentes etapas de SAINT. El flujo de información entre estos programas se lleva a cabo utilizando tuberías de Unix.

#### 10.4.11 Archivo de configuración

Casi todo el comportamiento de SAINT es configurable. Esto es necesario si se espera que la herramienta pueda ser utilizada en múltiples ambientes de trabajo, con necesidades y políticas diferentes.

Un subconjunto reducido de las opciones se puede configurar a través de los argumentos de la línea de comandos de SAINT (ver apéndice K). Sin embargo, el verdadero poder de configuración en SAINT se alcanza a través del archivo de configuración, llamado por default `saint.cf`, y localizado en el mismo directorio que el programa `saint`.

El archivo de configuración es un archivo de texto que contiene líneas que le especifican a SAINT sus parametros de configuración. Este archivo en realidad es código escrito en `perl5`, pero su sintaxis es sumamente sencilla, puesto que se trata, en su gran mayoría, de asignaciones de variables.

En el archivo de configuración es posible definir los siguientes parámetros:

- Módulos de soporte de herramientas que se deben cargar. De esta forma, SAINT se puede configurar para analizar solamente los resultados de las herramientas que sí se estén utilizando, o para cargar módulos de soporte a otras herramientas o a otro tipo de problemas.

- Horarios normales de trabajo. Esto se puede utilizar para detectar actividad poco común, y que pueda indicar problemas (por ejemplo, una secretaria estableciendo una sesión interactiva a las 3 de la mañana). Estos horarios se pueden establecer en términos generales y de forma individual para ciertos usuarios (para indicar usuarios que normalmente trabajan fuera de los “horarios normales”).
- Horarios de mantenimiento de los sistemas. En estos horarios normalmente es permitido realizar tareas poco comunes (como reinicializar los sistemas), mientras que fuera de estos horarios, dichas tareas pueden resultar sospechosas.
- Usuarios autorizados a utilizar la clave de **root**. Normalmente, la utilización de **root** por parte de un usuario que no esté autorizado a hacerlo es indicación clara de problemas.
- Direcciones electrónicas desde las cuales es normal acceder a los sistemas. Así, se pueden detectar conexiones desde sistemas extraños, lo cual podría también ser indicación de actividad sospechosa.
- Maquinas confiables. Esto se utiliza principalmente para determinar desde qué maquinas un usuario autorizado puede entrar a la clave de **root**.
- Dominios de seguridad. Es decir, conjuntos de maquinas entre las cuales se permiten ciertas acciones (como utilización de los comandos **rlogin** y **rsh** para el establecimiento de sesiones sin proporcionar *passwords*) que normalmente serían consideradas como sospechosas.
- Maquinas y dominios no confiables. Es decir, conjuntos de maquinas que se sepan causa frecuente de problemas. Esta información le sirve a SAINT para marcar conexiones desde dichas maquinas inmediatamente como posibles problemas.
- Módulo a utilizar por default para la presentación de los resultados finales.

#### 10.4.12 Consideraciones sobre SAINT

La cuestión mas importante que debe tenerse en cuenta al utilizar SAINT es que no es una solución mágica. De hecho, SAINT es una herramienta cuya única función es analizar información proveniente de otras herramientas. Por esto, se deben considerar los siguientes factores:

- SAINT no puede inventar información. Toda la información con la que trabaja SAINT proviene de otras herramientas. La información que no se le proporcione no puede ser extraída de ningún lado, aunque SAINT si puede hacer ciertas “deducciones” en base a elementos parciales de información. Por otro lado, mientras mas herramientas de monitoreo se utilicen, SAINT tendrá mas información con la cual realizar análisis mas precisos.

- **SAINT** es tan confiable como la información que recibe. Si los datos que **SAINT** recibe para su análisis ya han sido modificados por un intruso para esconder sus pistas, **SAINT** jamás lo podrá detectar. Por esta razón, se debe tener en cuenta, en las máquinas que van a ser monitoreadas por **SAINT**, la seguridad de los archivos utilizados por este programa. Por ejemplo, puede ser una buena idea enviar toda la información, a medida que va siendo generada, a una sola máquina, desde la cual sea tomada por **SAINT** para su análisis. El sistema **syslog** de Unix facilita esta tarea al permitir el registro remoto de información. Ver la página de manual **syslog(8)** para mayor información al respecto.

### 10.4.13 El futuro de **SAINT**

En la primera versión de **SAINT**, cuya liberación se espera en un futuro cercano, se implementara casi toda la funcionalidad descrita en esta sección. Sin embargo, hay muchas cosas que se pueden considerar para el futuro. Por ejemplo (se mencionan en el orden probable de su implementación):

- Generación de reportes en varios idiomas. En su versión original, y como parte de su propósito inicial, **SAINT** genera todos sus resultados en español. Sin embargo, para hacerlo completamente utilizable por parte de la comunidad internacional, se planea añadir soporte para la generación de resultados en múltiples lenguajes.
- Generación de información útil para otras herramientas. Por ejemplo, un programa podría sugerir el contenido de los archivos de configuración de **TCP-Wrapper** en base a la información de máquinas y dominios confiables contenida en el archivo de configuración de **SAINT**. Esto permitiría no solamente detectar eventos, sino prevenirlos.
- Interfaz gráfica. El programa **SATAN** sentó un precedente en la utilización de un visualizador de **WWW** para presentar sus resultados con una interfaz de hipertexto, lo cual facilita enormemente su lectura e interpretación. En pláticas con los autores de **SATAN** (Dan Farmer y Wietse Venema), el autor ha convenido la factibilidad de adaptar la interfaz gráfica de **SATAN** para su utilización en **SAINT**.
- Monitoreo en tiempo real, es decir, detección de los eventos a medida que van siendo registrados por las herramientas.

## 10.5 Dónde obtener las herramientas

Las herramientas de seguridad descritas en este capítulo pueden obtenerse por FTP anónimo en el servidor del Departamento de Supercómputo (`ftp.super.unam.mx`), en el directorio `/pub/security/tools/dgsca/`.

# Capítulo 11

## Servicios de seguridad

La prestación de servicios de seguridad forma parte muy importante del proyecto. Mediante estos servicios se pretenden lograr dos objetivos principales: difundir información sobre seguridad, e impulsar la formación de una infraestructura sólida de seguridad.

Por esta razón, este capítulo se divide en dos secciones: difusión de información y revisión y consultoría de seguridad.

### 11.1 Difusión de Información

Nunca serán demasiadas las veces que se repita la importancia de la difusión de información para lograr esquemas sólidos de seguridad, tanto en el aspecto técnico como en el humano. En el proyecto de seguridad se ha echado mano de múltiples recursos tecnológicos para lograr la difusión de dicha información. En esta sección se describe el trabajo realizado utilizando dichos recursos para auxiliar en la difusión de información de seguridad.

#### 11.1.1 Lista de correo electrónico *gasu*

Este medio de comunicación es vital para el proyecto de seguridad. Baste decir que *gasu* ha sido el medio de comunicación más importante en la comunidad de cómputo de la UNAM y algunas otras instituciones, sirviendo para la difusión de información de toda clase sobre administración y seguridad en Unix, y sin el cual muchas de las actividades que se han realizado, incluyendo la organización del DISC, habrían sido mucho más difíciles o incluso imposibles.

En el apéndice G se describen los detalles técnicos de esta lista, así como algunas estadísticas de su funcionamiento hasta la fecha.

### 11.1.2 Lista de correo electrónico *cert-advisory*

El CERT (*Computer Emergency Response Team*) [CER90] es un organismo encargado de difundir información sobre seguridad en cómputo, así como proporcionar ayuda a quienes lo soliciten en caso de un incidente de seguridad.

Una de las actividades de difusión que lleva a cabo el CERT es la difusión de los *CERT Advisories* (consejos del CERT). Estos son comunicados que se publican cada vez que se descubre algún problema de seguridad en un programa o sistema operativo. Un *CERT Advisory* es un documento que consta generalmente de las siguientes partes:

1. Descripción general del problema.
2. Impacto.
3. Soluciones propuestas.
4. Apéndices opcionales.

Estos documentos son distribuidos a través de una lista de correo electrónico manejada por el CERT/CC (*CERT Coordination Center*), cuya dirección electrónica es `cert-advisory@cert.org`.

Con el fin de promover la difusión, conocimiento y utilización de dichos documentos, se decidió, como parte del proyecto de seguridad, crear en la UNAM una lista de correo electrónico que sirviera como punto de redistribución de los consejos del CERT. La idea es que las personas dentro de la UNAM (o incluso dentro de todo México) que deseen recibir de forma automática los avisos del CERT, pueden suscribirse a la lista local, llamada `cert-advisory@listas.unam.mx` (apéndice G) en vez de suscribirse a la lista original.

A través de la lista local se distribuyen los mismos mensajes que en la lista original, y exactamente al mismo tiempo. La ventaja que se tiene, entonces, es sencillamente disminuir la cantidad de información que tiene que viajar a través de la red internacional, pues solamente una copia del mensaje viaja de Estados Unidos a México, y desde la lista local es distribuida a todos los suscriptores de la UNAM.

Para la puesta en marcha de este servicio se contó con el apoyo del equipo administrativo de la lista *cert-advisory* original, y especialmente de Ed DeHart, quien registró la lista `cert-advisory@listas.unam.mx` como suscriptor de `cert-advisory@cert.org`, de manera que se pudieran recibir los mensajes en la lista local.

Cabe mencionar que la creación de esta lista fue también producto de una discusión en *gasu* acerca de las ventajas y desventajas de distribuir los avisos del CERT en *gasu*. Después de realizar un sondeo entre los integrantes de GASU, se llegó a la decisión de crear una lista separada con tal propósito, que se materializó con la creación de la lista *cert-advisory* (ver apéndice G).

### 11.1.3 FTP anónimo de seguridad

Para difusión de información de seguridad también se ha hecho uso extensivo del servicio de FTP anónimo. Para detalles técnicos sobre el servidor utilizado, referirse al apéndice B.

Dentro del servidor de FTP anónimo del Departamento de Supercómputo se tiene una sección importante dedicada a información de seguridad. Se describe a continuación la estructura general de este servicio. Todos los elementos mencionados a continuación se refieren a directorios dentro del servidor. Para información sobre la utilización de este servicio, consultar el apéndice I. En todos los directorios existe un archivo llamado README que describe el contenido de dicho directorio.

`/pub/security/` Nodo raíz del servicio de FTP anónimo de seguridad. Debajo de este directorio se encuentra todo el material relacionado con seguridad en cómputo.

`/pub/security/tools/` Herramientas de seguridad. Debajo de este directorio se encuentran almacenadas todas las herramientas mencionadas en el apéndice D, así como algunas otras y programas auxiliares. Constantemente se añaden archivos a este directorio, a medida que se publican en el dominio público nuevas herramientas de seguridad.

`/pub/security/doc/` Documentos sobre seguridad. Debajo de este directorio se almacenan documentos de todo tipo sobre seguridad que están en el dominio público. Las categorías cubiertas en estos documentos son principalmente:

- Descripción de problemas de seguridad y soluciones propuestas.
- Descripción de herramientas de seguridad.
- Análisis de problemas de seguridad.
- Tutoriales sobre temas relacionados con seguridad.

`/pub/security/doc/lit/` Bibliografía sobre seguridad. Contiene diversas recopilaciones de bibliografía sobre seguridad en diversos temas.

`/pub/security/crypto/` Archivos sobre criptografía. Es el nodo raíz de la sección de criptografía del FTP anónimo de seguridad. Es una de las secciones más nuevas de este servidor, por lo que todavía no cubre todas las áreas básicas de la criptografía.

`/pub/security/crypto/tools/` Herramientas de criptografía. Contiene programas diversos que permiten hacer uso del cifrado de datos como mecanismo para proteger la información. Casi todas las herramientas que aparecen en este directorio se encuentran también en `/pub/security/tools/`.

`/pub/security/crypto/doc/` Documentos sobre criptografía. Contiene documentos de dominio público que discuten aspectos específicos de criptografía, como algoritmos, técnicas de criptoanálisis, etc.

`/pub/security/disc/` Nodo del DISC. Debajo de este directorio está almacenada toda la información relacionada con este evento, clasificada por años. Por el momento solamente contiene un directorio, `1994/`, que contiene la información y las ponencias del *Día Internacional de la Seguridad en Cómputo 1994*.

/pub/security/gasu/ Nodo de GASU. En este directorio se encuentra información general sobre GASU, las actividades realizadas, etc.

#### 11.1.4 *World Wide Web* (WWW)

A continuación se describen los usos que se le han dado al *World Wide Web* (WWW) en el proyecto de seguridad.

##### Utilización de WWW para difusión de información

Durante el transcurso del proyecto, se han ido desarrollando páginas de WWW sobre seguridad en cómputo. El objetivo de estas páginas es poner a disposición de la comunidad universitaria la mayor cantidad posible de información sobre el tema.

Actualmente, la página de WWW sobre seguridad es mantenida por el autor de esta tesis, y cubre los siguientes temas principales:

- Preguntas frecuentes sobre diversos temas de seguridad.
- Consejos de seguridad (*advisories*) generados por diversos organismos como CERT, CIAC, etc.
- Documentos sobre diversos temas de seguridad.
- Organizaciones de seguridad a nivel nacional e internacional.
- Herramientas de seguridad.
- Parches de seguridad para diferentes sistemas operativos.
- Ligas a otras páginas de seguridad en el mundo.

Cada uno de los temas contiene ligas a documentos que se encuentran en el servidor local del Departamento de Supercómputo, o a otras páginas en diferentes partes del mundo.

También se ha desarrollado una página de WWW sobre GASU. En esta página se puede encontrar información general sobre el grupo, así como acerca de las actividades realizadas y por realizarse, información generada, y el archivo de todos los mensajes distribuidos en la lista de correo electrónico *gasu*.

Los URLs de estas páginas son:

**Página de seguridad:** <http://www.super.unam.mx/seguridad/>

**Página de GASU:** <http://www.super.unam.mx/seguridad/gasu.html>

También se utilizó WWW para dar difusión al DISC. Con este fin se creó una página dedicada a este evento, y disponible en <http://www.super.unam.mx/seguridad/disc/1994/>.

### Utilización de WWW para recolección de información

La posibilidad de recolectar información a través de WWW abre horizontes nunca antes vistos en la recopilación y procesamiento de la información. La información puede ser recolectada de múltiples maneras, y posteriormente verificada y procesada de forma automática. La necesidad de un ser humano que esté recibiendo la información se hace prácticamente nula.

Durante el proyecto de seguridad se han utilizado formas de HTML en los siguientes puntos:

**Organización y coordinación del DISC** Durante la planeación de este evento se hizo uso de las capacidades interactivas de WWW para:

1. Registro de asistentes.
2. Registro de ponentes.

A pesar de que estos registros también se llevaron a cabo por los medios “tradicionales” (a través de correo electrónico, e incluso a través de formas impresas en papel), se ofreció la posibilidad de hacerlo a través de WWW, lo cual ofreció ventajas también para los organizadores, al realizarse automáticamente el procesamiento de la información. En la figura 11.1 se observa la pantalla de introducción de datos para el registro de asistencia al DISC.

**Cuestionario sobre GASU** Con motivo del primer aniversario de la formación de GASU, en Enero de 1995 se realizó una encuesta entre los integrantes, en la que se preguntaba la opinión sobre el funcionamiento de GASU durante este año, así como información general sobre el estado de la seguridad en las distintas instituciones adscritas al grupo. En la figura 11.2 se puede apreciar la pantalla inicial de dicha encuesta.

### 11.1.5 Boletín de Supercómputo

El *Boletín de Supercómputo* es una publicación aperiódica del Departamento de Supercómputo, en el que se tratan diversos temas relacionados con la supercomputadora, sus aplicaciones y sus usuarios, así como algunos otros temas de interés general. Hasta el momento han sido publicados dos números de este boletín.

Este medio también se utiliza para difundir información sobre seguridad en cómputo. En el número 2 del Boletín apareció el primer artículo de una serie sobre seguridad, escrita por el autor de esta tesis. En éste se tratan los conceptos básicos de seguridad, y posteriormente se discutirán aspectos más específicos de seguridad en Unix y en la supercomputadora.

### 11.1.6 Asistencia a foros externos

Se considera de suma importancia difundir el trabajo realizado en la UNAM en cuanto a seguridad en cómputo en instituciones externas a la máxima casa de estudios. Así

NCSA Mosaic: Document View

File Options Navigate Annotate News Help

Title: Forma de registro al Día de la Seguridad en Cómputo

URL: http://www.super.unam.mx/seguridad/disc/1994/registro.p1

### Forma de registro para el Día Internacional de la Seguridad en Cómputo

5 de Diciembre de 1994  
Departamento de Supercómputo  
Dirección General de Servicios de Cómputo Académico  
UNAM

**Instrucciones**

Escriba los datos correspondientes en las ventanillas, o seleccione la respuesta adecuada cuando se trate de una pregunta de opción múltiple. Cuando termine, oprima el botón de ENVIAR que se encuentra al final de la forma, el botón de CANCELAR si no desea enviar los datos.

**Datos personales**

Nombre:

Dirección Electrónica:

Sitio de trabajo:

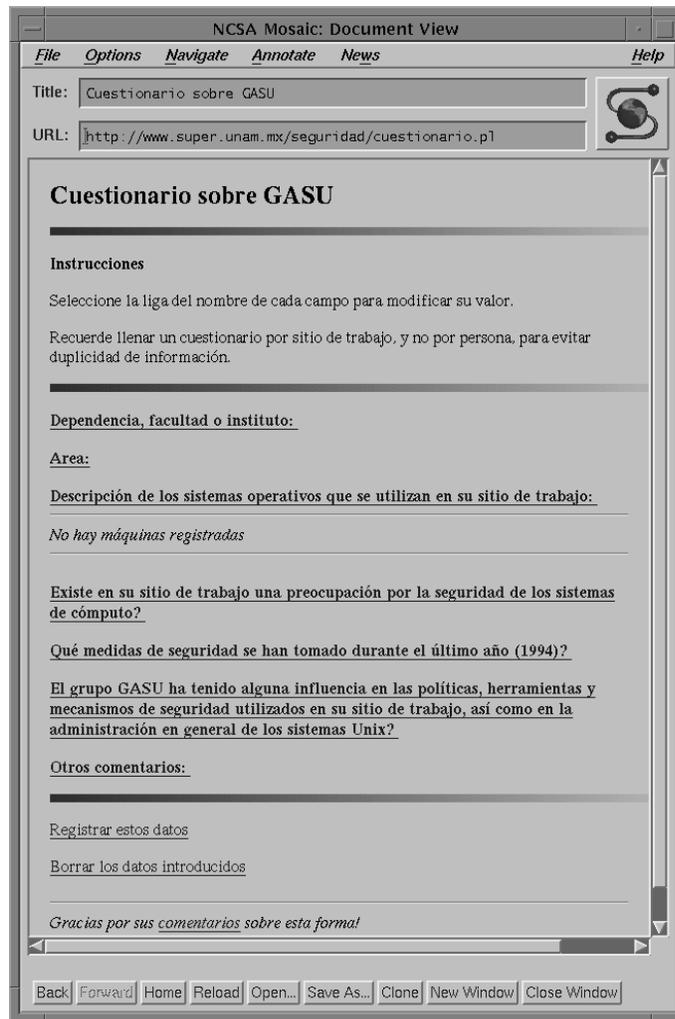
Puesto:

**Datos sobre su trabajo**

Es usted *usuario* de un sistema de cómputo?  SI

Back Forward Home Reload Open... Save As... Clone New Window Close Window

Figura 11.1: Forma de registro de asistencia al DISC en WWW



The image shows a screenshot of the NCSA Mosaic web browser. The title bar reads "NCSA Mosaic: Document View". The menu bar includes "File", "Options", "Navigate", "Annotate", "News", and "Help". The address bar shows the URL "http://www.super.unam.mx/seguridad/cuestionario.p1". The main content area displays the following text:

**Cuestionario sobre GASU**

---

**Instrucciones**

Seleccione la liga del nombre de cada campo para modificar su valor.

Recuerde llenar un cuestionario por sitio de trabajo, y no por persona, para evitar duplicidad de información.

---

Dependencia, facultad o instituto:

Area:

Descripción de los sistemas operativos que se utilizan en su sitio de trabajo:

*No hay máquinas registradas*

---

Existe en su sitio de trabajo una preocupación por la seguridad de los sistemas de cómputo?

Qué medidas de seguridad se han tomado durante el último año (1994)?

El grupo GASU ha tenido alguna influencia en las políticas, herramientas y mecanismos de seguridad utilizados en su sitio de trabajo, así como en la administración en general de los sistemas Unix?

Otros comentarios:

---

Registrar estos datos

Borrar los datos introducidos

---

*Gracias por sus comentarios sobre esta forma!*

At the bottom of the browser window, there is a navigation bar with buttons for "Back", "Forward", "Home", "Reload", "Open...", "Save As...", "Clone", "New Window", and "Close Window".

Figura 11.2: Encuesta sobre GASU en WWW

se logra ampliar aún más el alcance del proyecto. Aunque el equipo de coordinación del proyecto no puede tener influencia directa sobre las acciones realizadas fuera de la Universidad, estas acciones pueden, por sí mismas, ejercer una influencia al establecer antecedentes de trabajo sobre seguridad en México y proporcionar ideas sobre las acciones que pueden tomarse. Además, siempre se ha considerado la posibilidad de realizar trabajo conjunto con otras instituciones, ya sean de la UNAM o externas.

El acto de difusión más importante en el que se ha participado es el *Foro de Consulta Popular sobre Informática*, organizado por el INEGI (Instituto Nacional de Estadística, Geografía e Informática), y llevado a cabo el 18 de abril de 1995 en la Ciudad de México.

El autor de esta tesis participó en dicho foro con la ponencia *La Seguridad en Cómpu-to en México*, que fue presentada en la mesa de trabajo “Mercado informático y uso de la informática en el sector privado”. De esta forma, fue posible plantear las acciones realizadas en la UNAM ante representantes de múltiples sectores de la sociedad mexicana, desde otras instituciones educativas hasta bancos. Esta presentación causó gran interés en algunas personas, y los comentarios recibidos han sido muy valiosos para darle al proyecto de seguridad en la UNAM su contexto apropiado dentro de un marco más amplio.

De esta presentación se derivó la posibilidad de trabajo conjunto, y posteriormente se presentó un artículo que será integrado a la *Reseña de Informática en el Ámbito Académico*, que está siendo elaborada por el Departamento de Monitoreo Tecnológico del INEGI.

## 11.2 Revisión y consultoría de seguridad

Uno de los obstáculos más notorios en la utilización de herramientas y mecanismos de seguridad por parte de los administradores de sistemas Unix es la dificultad implícita en su instalación, configuración y utilización, así como en la comprensión de los conceptos necesarios para su aprovechamiento efectivo. Muchas personas sienten que es necesario ser un experto en seguridad para implementarla, y el resultado es que la seguridad nunca encuentra un lugar en sus labores cotidianas.

Para disminuir este problema, el autor de esta tesis ha iniciado, desde la coordinación del proyecto de seguridad, la prestación de servicios de seguridad a dependencias universitarias y externas, que se describen a continuación.

### 11.2.1 Revisión remota utilizando SATAN

SATAN es una de las más recientes e innovadoras herramientas de seguridad (ver apéndice D). Tiene una gran utilidad para los administradores de los sistemas en la detección de problemas de seguridad, pero también tiene una gran utilidad para alguien que esté buscando puntos fáciles de acceso no autorizado.

Debido al gran peligro que implica la no utilización de SATAN de forma oportuna por parte de los administradores, el primer servicios de seguridad que se ofreció fue la revisión remota de sistemas utilizando dicho programa.

En este servicio, el interesado envía a la coordinación del proyecto de seguridad (el autor) una solicitud por escrito, indicando qué máquinas se desea revisar. Utilizando SATAN, se revisan remotamente los sistemas solicitados, y los resultados se hacen llegar, por escrito, al solicitante.

Este servicio fue anunciado el 5 de abril de 1995 a través de GASU. El anuncio oficial se puede consultar en el apéndice M de esta tesis, donde también se detallan las condiciones para la revisión, y los pasos que hay que seguir para solicitarla.

Hasta la fecha, se han revisado los sistemas de las siguientes áreas y dependencias:

- Departamento de Supercómputo, DGSCA.
- Laboratorio de Visualización, DGSCA.
- Departamento de Redes, DGSCA.
- Cray Research de México.
- Instituto de Física, UNAM.
- Aula de Cómputo, Departamento de Matemáticas, Facultad de Ciencias, UNAM.

De acuerdo a los términos en los que se presta el servicio, la información obtenida es confidencial, por lo que no es posible divulgar detalles sobre las vulnerabilidades encontradas en estas revisiones. Se puede mencionar, sin embargo, que en todos los casos se ha hecho todo lo posible para que los interesados corrijan los problemas detectados.

### 11.2.2 Análisis remoto utilizando NCARP

NCARP es un programa desarrollado por el autor de esta tesis que permite el análisis de los reportes generados por COPS. Dado que su instalación y utilización puede estar fuera de los intereses, los conocimientos o el tiempo de muchos administradores, se encuentra en desarrollo un servicio de análisis remoto de los reportes generados por COPS, utilizando NCARP.

De esta forma, el interesado envía por correo electrónico, a una dirección determinada (`oss@ds5000.dgsc.unam.mx`) los reportes generados por COPS en su máquina. Estos reportes son analizados de forma automática por NCARP, y el resultado del análisis se envía de regreso, también por correo electrónico.

Un problema que se presenta en este esquema es la seguridad de los mensajes enviados por correo electrónico. Es conocida la facilidad con la que un mensaje de correo electrónico puede ser interceptado y leído por personas que no sean el destinatario autorizado. Por esto, si se envían reportes de seguridad a través de correo electrónico, se está abriendo la posibilidad de que dichos reportes sean leídos por personas no autorizadas, y que en un momento dado puedan utilizar la información contenida en ellos para obtener acceso a los sistemas de cómputo.

Para solucionar este problema se piensa utilizar mecanismos de cifrado de información en dichos mensajes. El programa más adecuado para este fin es PGP (*Pretty Good Privacy*) de Philip Zimmermann [Zim94, Gar95], y actualmente se encuentra en estudio su utilización para el cifrado de los mensajes de forma automática.

Este servicio no ha sido liberado oficialmente a la comunidad universitaria y externa, pero ya se encuentra en utilización en los sistemas de Supercómputo y Visualización, donde, debido a tratarse de redes locales, el problema de la interceptación de correo electrónico no es demasiado grave. Su liberación completa se espera, en el mes de julio de 1995.

### **11.2.3 Otros servicios, el futuro**

Hasta el momento, los servicios ofrecidos se concentran en el área de revisión y análisis remotos de seguridad. Sin embargo, dentro de los planes del proyecto de seguridad se encuentra el ofrecimiento de servicios más variados y completos. Uno de estos servicios podría ser la realización de consultorías completas sobre seguridad, que cubrieran los siguientes puntos:

- Revisión de la seguridad en los sistemas.
- Instalación de herramientas de seguridad.
- Integración de los sistemas a los servicios de revisión remota.
- Capacitación del personal de la dependencia o institución.

Para poder proporcionar estos servicios, sin embargo, se requiere de recursos humanos con los que todavía no se cuenta, pero que se espera desarrollar a través de las actividades de capacitación realizadas.

## Capítulo 12

# Actividades Institucionales y Legales

Aunque esta tesis se concentra sobre los aspectos técnicos y humanos de la seguridad en cómputo, es importante reconocer la necesidad de actividades logísticas y legales a nivel institucional, así como el trabajo conjunto con otras organizaciones que se dedican a la seguridad en cómputo. Estas medidas permitirán el establecimiento de la seguridad en cómputo como un problema reconocido oficialmente, y sobre el cual se puedan tomar medidas concretas a nivel Universitario.

En este capítulo se describen las actividades que se han realizado en estos aspectos.

### 12.1 Asesorías a otras dependencias universitarias

Al difundirse las actividades que se estaban llevando a cabo a través del proyecto de seguridad, algunas dependencias universitarias mostraron interés en tomar medidas al respecto. A continuación se describen las acciones tomadas.

#### 12.1.1 Instituto de Astronomía

Este instituto solicitó la ayuda de la DGSCA para realizar una revisión de incidentes de seguridad en los que se realizaron accesos no autorizados a una supercomputadora del North Carolina Supercomputing Center (NCSC) desde sistemas de esa dependencia. Con este motivo, en enero de 1994 se llevó a cabo una revisión detallada de los sistemas de esta dependencia, obteniéndose los siguientes resultados.

- Los accesos a la supercomputadora del NCSC se realizaron desde la sesión de un investigador de dicho Instituto.
- Los accesos no se realizaron por parte del dueño de la clave.
- El acceso a la clave se hizo posible debido a que dicho investigador dejó su sesión activa en una oficina abierta, durante varios días.

Se notificó de estos hechos a la directiva del Instituto de Astronomía, y aunque no fue posible detectar a la persona responsable de los accesos no autorizados, se tomaron medidas para evitar la reincidencia de este tipo de acciones en el futuro.

### **12.1.2 Unidad de Servicios de Cómputo Académico, Facultad de Ingeniería(USCAFI)**

Esa dependencia, a través del Ing. Victor Pinilla, Jefe de la misma, también solicitó la ayuda del equipo de seguridad de la DGSCA. El problema que se tuvo en este sitio fue de un tipo muy común: problemas con personal interno de la organización. Los hechos, según fueron planteados por el Ing. Pinilla, fueron los siguientes:

1. Todas las estaciones de trabajo de dicho centro de cómputo eran administradas por solamente dos personas, que no aceptaban ayuda de nadie más.
2. Al asumir el cargo el Ing. Pinilla e intentar distribuir la administración de los equipos, comenzaron a darse fricciones con los administradores.
3. Finalmente, las dos personas que administraban los equipos tuvieron que dejar CECAFI.
4. Después de que dichas personas se fueron, comenzaron a notarse problemas extraños en los equipos. Al hacer una revisión, se cayó en la cuenta de que los administradores anteriores, antes de irse, habían dejado plantadas una serie de “bombas lógicas” en los sistemas, que estaban causando daños de forma automática. Esto llevó incluso a la pérdida total de la información almacenada en algunos de los sistemas.

Ante esta situación desesperada, el personal de CECAFI decidió reinstalar el sistema operativo en todas las máquinas. Después de esto, se le solicitó a la DGSCA ayuda para implementar en los sistemas los mecanismos básicos de seguridad. Con este motivo, en junio de 1994 se llevó a cabo en las máquinas de CECAFI la instalación de las siguientes herramientas de seguridad:

- COPS
- TCP-Wrapper
- passwd+

Asímismo se dieron una serie de recomendaciones para seguridad, y la invitación de integrarse a GASU para obtener mayor información.

## **12.2 Consideración de la seguridad en cómputo por parte de las autoridades universitarias**

Como en muchos otros sitios, la preocupación por la seguridad en cómputo en la UNAM era prácticamente nula. Por esto, como parte del proyecto, se llevaron a cabo algunas

acciones para atraer la atención de las directivas correspondientes sobre este importante tema.

La formación de GASU ha llamado la atención de las autoridades de la DGSCA, que le dieron su apoyo desde el inicio. Asimismo, las autoridades de otras dependencias universitarias han visto en GASU una buena idea y han aprovechado los seminarios y las actividades realizadas para lograr una mejor capacitación del personal a su cargo encargado de la administración de sistemas Unix.

Sin embargo, estas son acciones aisladas. Todavía falta mucho por hacer a nivel UNAM y en el campo legal para que se pueda considerar que existe un verdadero interés directivo por la seguridad. En cuanto a cómputo administrativo, en el que sí se maneja información confidencial, las consideraciones de seguridad se han dejado siempre al final y “si queda tiempo”. La integración de mecanismos de seguridad en el cómputo administrativo de la UNAM es un área de acción muy amplia, y que no está contemplada dentro de los objetivos de este proyecto.

### 12.3 Legislación universitaria sobre seguridad en cómputo

Uno de los pasos que más ha costado dar en cuanto a seguridad en cómputo a nivel mundial es integrarla en las leyes. Resulta imposible tomar acciones civiles o penales contra quienes cometen delitos de seguridad en cómputo sin contar con leyes al respecto, y ha sido un proceso muy lento el convencer a los legisladores de que las violaciones a la seguridad también son delitos ([RG92, pp. 38–52], [GS92, cap. 17]).

En México no ha sido la excepción. De hecho, en nuestro país todavía no existe legislación sobre seguridad en cómputo a ningún nivel.

Dentro de la UNAM, los incidentes ocurridos causaron un “despertar” en las autoridades universitarias, que durante un período corto de tiempo se dieron cuenta de la importancia de la seguridad en cómputo. Después de este incidente se habló de la creación de una legislación universitaria que tratara sobre seguridad en cómputo y los castigos asociados a los delitos de esta índole. Sin embargo, dicho proyecto se ha estancado, y hasta el momento no se cuenta en la UNAM con ninguna legislación que defina y castigue los delitos de seguridad en cómputo.

### 12.4 Trabajo con otros organismos de seguridad

La seguridad en cómputo es un área en la que se puede aprender mucho del trabajo realizado por otras personas. También es un campo en el que la cooperación vale mucho. Sobre todo en el mundo del cómputo actual, en el que la comunicación a nivel global se ha hecho posible gracias a Internet, los problemas de seguridad pueden extenderse rápidamente. En muchas ocasiones, trabajar unidos es la única forma de solucionar un problema ([Sto89], [GS92, pp. 314–318], [Sto88], [Spa91]).

Por esto, no resulta extraño que con el paso del tiempo hayan surgido equipos de trabajo especializados en seguridad, llegando a formar incluso organizaciones enteras dedicadas a este campo.

El grupo de seguridad en cómputo en la UNAM se encuentra aún en estado de desarrollo. Sin embargo, se ha seguido de cerca el trabajo de otros organismos, con el fin de aprender las técnicas, mecanismos y políticas que ya han sido probadas en otras partes del mundo, así como de mantenerse al día en información sobre seguridad en cómputo.

### 12.4.1 CERT

Uno de los organismos de seguridad en cómputo más relevantes a nivel mundial es el CERT [CER90]. En la UNAM se ha trabajado de cerca con este organismo en los siguientes aspectos:

- Difusión, estudio y aplicación, en su caso, de los consejos del CERT.
- Utilización de las herramientas de seguridad recomendadas por el CERT.
- Colaboración con el CERT para la solución de los incidentes de seguridad en la UNAM.

### 12.4.2 Laboratorio Nacional de Los Alamos

El Laboratorio Nacional de Los Alamos (LANL) en los Estados Unidos fue en una época el centro de desarrollo militar más avanzado en ese país. Hoy es, sin embargo, uno de los centros de investigación general más avanzados, y uno de los que cuenta con mayores recursos económicos para investigación y desarrollo de vanguardia.

En el mes de junio de 1993, un grupo de personas de DGSCA realizó una visita a este laboratorio con el objeto de conocer sus instalaciones y obtener información que pudiera ser útil para la utilización de la supercomputadora en México, dado que el LANL es uno de los principales usuarios del supercómputo en el mundo.

En esta visita, entre muchos otros temas, se tocó el de la seguridad. El autor de esta tesis preparó para dicha visita un reporte del estado de la seguridad en la supercomputadora de la UNAM (apéndice L). Este documento describe las acciones que habían sido tomadas en la supercomputadora a la fecha de su realización, y planteaba preguntas acerca de posibles medidas a tomar.

El equipo de seguridad del LANL analizó dicho documento y proporcionó algunas ideas valiosas sobre acciones que vale la pena tomar en cuanto a seguridad, sobre todo en las áreas de capacitación de los usuarios y administradores. En cuanto a la utilización de herramientas, resultó ser una agradable sorpresa averiguar que incluso en laboratorios de cómputo tan avanzados como el LANL se utilizan las mismas herramientas que se han utilizado desde 1993 en la supercomputadora y las estaciones de trabajo de Supercómputo y Visualización.

## Capítulo 13

# Resultados obtenidos y conclusiones

Al respecto de la seguridad en Unix, es difícil hablar de “haber terminado el trabajo”. La tecnología avanza constantemente, y prácticamente todos los días aparecen nuevos sistemas, nuevos productos, nuevos protocolos, nuevos servicios de red. Y acompañándolos, aparecen nuevos problemas de seguridad. Es por esto que el trabajo de un equipo de seguridad no termina, simplemente evoluciona.

En cuanto a este proyecto, se ha logrado “arrancar el juego”: la seguridad, que antes era tema de ciencia ficción para muchos administradores y usuarios de sistemas Unix en la UNAM, ahora tiene un lugar en los planes y acciones de muchos de ellos. En GASU se ha logrado promover la cultura de la seguridad en muchos aspectos. La organización de seminarios y actividades diversas ha despertado al menos la curiosidad de muchas personas que antes utilizaban los sistemas de cómputo sin dedicarle el menor pensamiento a la seguridad.

Con la prestación de servicios a la comunidad universitaria, el proyecto de seguridad ha ido adquiriendo una presencia aún más notoria en el ámbito del cómputo en la UNAM. No son pocas las personas que actualmente se ponen en contacto continuamente para obtener información, pedir ayuda o solicitar un servicio de seguridad.

La creación de nuevas herramientas es una pequeña muestra de lo mucho que todavía se puede hacer en ese aspecto. Las herramientas creadas resuelven problemas reales, se utilizan en situaciones reales, y están comenzando a ser difundidas incluso fuera de DGSCA, que es el círculo directo de acción de este proyecto.

La organización del DISC es un paso importante en la difusión y promoción de la cultura de la seguridad. Se espera que este evento se siga realizando año con año, atrayendo un número cada vez mayor de personas e instituciones.

En DGSCA, las acciones de seguridad han sido notorias. Actualmente, todas las computadoras con sistema operativo Unix del Departamento de Supercómputo y el Laboratorio de Visualización, incluyendo la supercomputadora Cray, están integradas en un sistema de monitoreo automático de seguridad que genera información útil para los administradores, y permite detectar casi cualquier tipo de actividad no autorizada. Los

resultados están a la vista. Desde el mes de julio de 1993 no se ha detectado en DGSCA un solo acceso no autorizado a los sistemas. Ha habido intentos, pero ninguno de ellos ha sido exitoso debido en gran parte a los mecanismos de seguridad implementados en las máquinas, y a la conciencia que se ha logrado crear en los administradores de las mismas.

Se ha logrado recopilar una considerable base de información y conocimiento en materia de seguridad en cómputo. La seguridad se ha convertido en una especialidad de algunos elementos del personal de DGSCA —incluido el autor de esta tesis—, y un elemento real en el esquema de cómputo de muchos otros—incluyendo, afortunadamente, el personal directivo de la dependencia.

El trabajo no está terminado. El proyecto de seguridad no finaliza, sino que apenas nace. Lo que se ha hecho desde el segundo semestre de 1993 a la fecha son acciones importantes, pero es una parte muy pequeña de lo que hay que lograr. En la tarea de la seguridad en cómputo, más que en ninguna otra especialidad, es imposible “dormirse en los laureles”. Si dejamos que eso suceda, el despertar será más duro que nunca. Los problemas de seguridad siempre están al acecho, y es muy importante estar alerta para cuando se presenten, y para preferentemente evitar que se presenten.

Las posibilidades al futuro son promisorias. Aunque es todavía una posibilidad lejana, no se descarta ya, a niveles directivos en DGSCA, la formación de un área especializada en la seguridad en cómputo, que posteriormente podría convertirse en un Programa Universitario de Seguridad en Cómputo. De realizarse esto, la UNAM habría dado un paso muy importante y sin precedente a nivel nacional que significaría un avance histórico en el cómputo de nuestro país.

Mientras tanto, GASU, el DISC y el proyecto de seguridad dentro del Departamento de Supercómputo siguen su paso. El camino es largo, pero los resultados valen la pena.

## Apéndice A

# Servidor de listas de correo electrónico del Departamento de Supercómputo

### A.1 ¿Qué es un servidor de listas de correo electrónico?

Las listas de correo electrónico son uno de los medios más eficientes actualmente para comunicar en Internet a personas que pueden estar a miles de kilómetros de distancia.

El elemento central de una lista de correo electrónico es el servidor. Este es un programa que corre en alguna máquina, y que maneja de forma automática al menos los siguientes aspectos:

- Peticiones de suscripción y desuscripción a las listas.
- Distribución de los mensajes para cada lista.

Existen varios servidores de listas de correo que son libremente utilizables por estar disponibles en Internet. Los principales son:

- ListProc.
- Majordomo.
- LISTSERV.
- SmartList.

Todos estos sistemas ofrecen, en general, funcionalidades similares, pero cada uno de ellos se diferencia en ciertos aspectos menores, así como en la forma de utilización del servidor. Una descripción y comparación detallada de estos servidores y algunos otros se encuentra en *The Mailing List Management Software FAQ* [Ale].

## A.2 ¿Qué servidor se utiliza en el Departamento de Supercómputo?

El servidor utilizado en el Departamento de Supercómputo es ListProc [Kot94]. Este sistema, desarrollado por Anastasios Kotsikonas, es uno de los manejadores de listas de correo más antiguos, y definitivamente el más conocido. Está compuesto por una serie de programas en **C** y **Bourne Shell** que permiten realizar las siguientes funciones principales:

- Manejo de un número prácticamente ilimitado de listas de correo.
- Cada lista tiene su propio “dueño”, que es el responsable de manejar a los suscriptores de la lista. De esta forma, la administración de las listas no recae sobre el administrador del sistema en el que reside el servidor.
- Cada lista puede tener un “moderador”. Esta persona, si la lista es moderada, es responsable de autorizar los mensajes que se manden a la lista, antes de ser distribuidos. Esto permite tener control sobre el tipo y número de los mensajes que se distribuyen.
- Almacenamiento permanente de los mensajes que se distribuyen en una lista (configurable por lista).
- Manejo de listas públicas y privadas. Una lista pública es aquella en la que las peticiones de suscripción se procesan de forma completamente automática. En una lista privada, por el contrario, las peticiones de suscripción tienen que ser aprobadas por el dueño de la lista antes de ser procesadas.
- Manejo de listas moderadas o no moderadas. En una lista no moderada, los mensajes que se envían a la lista son automáticamente distribuidos a los suscriptores. En una lista moderada, por el contrario, cada mensaje enviado debe ser aprobado por el moderador de la lista (cuya dirección electrónica se define al momento de crearla) antes de ser distribuido.
- Generación automática de *digests* (compendios periódicos de todos los mensajes distribuidos en la lista).
- Funcionamiento configurable de acuerdo a la carga de trabajo de la máquina en la que reside el servidor.

De acuerdo a [Ale], ListProc es uno de los servidores que impone menos carga en el sistema, debido a su filosofía de diseño y a que las partes principales están escritas en **C** (por el contrario, Majordomo está escrito en **perl**, que es un lenguaje interpretado).

La versión de ListProc utilizada es la 6.0c. Cabe mencionar que esta es la última versión gratuita de ListProc, pues los derechos de este sistema fueron adquiridos por la empresa CREN (*Corporation for Research and Educational Networking*), convirtiéndolo en un producto comercial a partir de la versión 7.0.

## A.3 Configuración del servidor de listas

El servidor de listas del Departamento de Supercómputo, aunque fue creado inicialmente para el manejo de la lista *gasu*, actualmente maneja también otras listas que se refieren a diferentes temas.

Al momento de escribir esta tesis, el servidor de listas reside en *ds5000*, una de las estaciones de trabajo del Departamento de Supercómputo. Sin embargo, muy pronto el servidor será cambiado a *mezcal*, una máquina dedicada exclusivamente al suministro de servicios de red por parte del Departamento de Supercómputo.

Esto obviamente implica que la dirección electrónica del servidor va a cambiar. Normalmente esto causaría múltiples problemas, sobre todo con los suscriptores de las listas, que ya están acostumbrados a enviar sus peticiones y sus mensajes a *ds5000*.

Sin embargo, desde el principio se tomaron medidas para evitar este tipo de problemas. Con la ayuda del Departamento de Redes de la DGSCA, al momento de instalar el servidor de listas se creó en el Servidor de Nombres (DNS—*Domain Name Service*) un registro MX (*Mail Exchanger*) para el servidor. Este registro establece una dirección “virtual” a la que se puede enviar correo electrónico, que será direccionado a alguna máquina real definida en las bases de datos del DNS.

Para el caso del servidor de listas, se creó un registro MX llamado `listas.unam.mx` y que apunta, por el momento, a `ds5000.dgsca.unam.mx`. Desde la creación del servidor, su dirección electrónica se anunció como `listas.unam.mx`, por lo que es a esta dirección a la que la gran mayoría de los usuarios envía sus peticiones y mensajes.

Al cambiar el servidor de listas a otra máquina, lo único que hay que hacer es modificar las tablas del servidor de nombres para que `listas.unam.mx` apunte a `mezcal.dgsca.unam.mx`. Así, los usuarios que utilicen dicho *alias* ni siquiera notarán el cambio.

## A.4 Cómo se usa el servidor de listas de correo

Toda la interacción de los usuarios, los dueños de las listas y los moderadores (en caso de haberlos) con ListPROC se lleva a cabo a través de correo electrónico. La dirección electrónica del servidor es `listproc@listas.unam.mx`, y a esta dirección deben enviarse todas las peticiones y órdenes. En estos mensajes, el encabezado **Subject:** debe ir vacío, y las órdenes deben estar en el cuerpo del mensaje.

Los mensajes que vayan a ser distribuidos a alguna lista de correo electrónico deben ser enviados a `lista@listas.unam.mx`, donde *lista* es el nombre de la lista de correo.

Para obtener ayuda general sobre el uso del servidor, hay que enviar el siguiente mensaje a `listproc@listas.unam.mx`:

**help**

Para suscribirse a una lista de correo, se utiliza:

**subscribe lista Nombre Real**

Donde *lista* es el nombre de la lista a la que se desea suscribirse, y Nombre Real es el nombre de la persona. No es necesario enviar la dirección electrónica, pues es extraída automáticamente de los encabezados del mensaje.

Para cancelar la suscripción a una lista, se utiliza:

```
unsubscribe lista
```

Se puede enviar más de una orden en un mismo mensaje de correo electrónico.

Por ejemplo, si Diego Zamboni desea suscribirse a la lista *gasu* y obtener ayuda sobre el uso del servidor, debe enviar el siguiente mensaje a `listproc@listas.unam.mx`:

```
subscribe gasu Diego Zamboni  
help
```

## Apéndice B

# Servidor de FTP anónimo del Departamento de Supercómputo

### B.1 ¿Qué es un servidor de FTP anónimo?

FTP es un protocolo estándar definido en el *RFC959* [PR85], y que está formado por una serie de órdenes y mensajes que son enviados entre un cliente (el que está realizando la transferencia de los archivos) y un servidor (el que proporciona acceso a los archivos).

Un servidor de FTP, por lo tanto, es un programa que cumple con las especificaciones de [PR85] y que permite la transferencia remota de archivos.

Un servidor de FTP anónimo es un servidor de FTP que permite la transferencia de archivos en forma pública, sin requerir ninguna clase de autenticación por parte del usuario.

Todos los sistemas Unix incluyen un servidor de FTP, llamado **ftpd**(8), y que implementa las órdenes básicas especificados que debe cumplir un servidor de FTP. Sin embargo, este servidor sufre de muchas deficiencias, las principales de las cuales son:

1. Falta de control de acceso (cualquiera puede utilizar el servidor).
2. Falta de registro de actividades (no es posible saber quién transfirió qué archivos).

Por esta razón, con el transcurso de los años (el protocolo FTP existe desde 1971) han sido desarrollados servidores de FTP alternativos, que implementan el protocolo, pero además ofrecen múltiples facilidades que le hacen la vida más fácil a los usuarios y a los administradores de los servidores.

### B.2 ¿Qué servidor de FTP se utiliza en el Departamento de Supercómputo?

Uno de los servidores de FTP más difundidos en Internet es el desarrollado en la Universidad de Washington en San Luis, llamado **wu-ftpd** [O'C94]. Además de implementar

el protocolo FTP definido en [PR85], añade las siguientes características [O’C94, archivo README]:

- Registro de transferencias y órdenes ejecutadas.
- Compactación y compresión de archivos “al vuelo”.
- Clasificación de los usuarios dependiendo de su tipo y el sitio desde donde se realiza el acceso.
- Límites de acceso por clase de usuarios.
- Cuentas de “visitante” restringidas.
- Configuración de mensajes por directorio y globales.
- Manejo de alias para directorios comunmente usados.
- Filtro de nombres de archivos inválidos.

Una de las características más prácticas para el usuario es la compactación y compresión al vuelo de los archivos. Esto permite a los usuarios comprimir de forma automática archivos e incluso directorios completos. En la figura B.1 se observa una secuencia de órdenes en la que el usuario comprime automáticamente un archivo utilizando `gzip`, y compacta un directorio completo utilizando `tar(1)` y `gzip`.

### B.3 Configuración del servidor de FTP anónimo

El servidor de FTP anónimo del Departamento de Supercómputo fue creado con anterioridad al inicio del proyecto de seguridad, pero ha sido utilizado ampliamente como medio de difusión de herramientas e información sobre el mismo.

En su inicio, el servidor de FTP anónimo residía en *ds5000*. Sin embargo, actualmente ha sido movido a *mezcal*. En esta máquina se están concentrando todos los servicios de red, como FTP anónimo, listas de correo y WWW.

Para evitar problemas al realizar el cambio de máquina, se siguió el mismo proceso que con el servidor de listas: desde su inicio, el servidor de FTP anónimo se accedió a través de un registro MX en el servidor de nombres llamado `ftp.super.unam.mx`. Al realizarse el cambio de *ds5000* a *mezcal*, sencillamente se cambió dicho registro para apuntar al nuevo servidor de FTP.

Ver el apéndice I para instrucciones de uso de FTP anónimo.

```
ftp> dir
200 PORT command successful.
150 Opening ASCII mode data connection for /bin/ls.
total 3
-r--r--r-- 1 ftp      ftp          17982 Jan 30  1993 COPYING
dr-xr-xr-x 2 ftp      ftp           512 Dec 20 21:40 protocols
dr-xr-xr-x 7 ftp      ftp          1024 Feb 26 19:06 tools
226 Transfer complete.
435 bytes received in 0.14 seconds (3.1 Kbytes/s)
ftp> get COPYING.gz
150 Opening BINARY mode data connection for /bin/gzip.
226 Transfer complete.
local: COPYING.gz remote: COPYING.gz
6843 bytes received in 0.18 seconds (37 Kbytes/s)
ftp> get protocols.tar.gz
200 PORT command successful.
150 Opening BINARY mode data connection for /bin/tar.
226 Transfer complete.
local: protocols.tar.gz remote: protocols.tar.gz
641265 bytes received in 7.4 seconds (84 Kbytes/s)
ftp>
```

Figura B.1: Ejemplo de compresión y compactación automática de archivos utilizando wu-ftp



## Apéndice C

# Servidor de HTTP del Departamento de Supercómputo

### C.1 ¿Qué es un servidor de HTTP?

WWW es uno de los mecanismos de comunicación recientemente creados que está tomando más fuerza en Internet. A través de WWW pueden transferirse textos, imágenes, sonidos, animaciones, archivos, y prácticamente cualquier tipo de información imaginable.

La transferencia de información en WWW se realiza utilizando un protocolo conocido como HTTP (*HyperText Transfer Protocol*) [conb]. Los documentos en WWW están escritos en un lenguaje conocido como HTML (*Hyper-Text Markup Language*) [Cona], que contiene elementos para especificar la estructura y contenido del documento, y que serán interpretados por el visualizador de WWW para ser desplegados en la pantalla del usuario.

Un servidor de HTTP es un programa que recibe peticiones de HTTP a través de la red, las interpreta, y transfiere el documento de HTML apropiado, que será recibido por el visualizador que realizó la petición.

Existen múltiples servidores de HTTP, con funcionalidades prácticamente iguales en cuanto a transferencia de documentos de HTML sencillos. En donde varían sin embargo, es en sus capacidades de manejo de formas de HTML. Los servidores más comunes son:

- Servidor httpd de NCSA (*National Center for Supercomputing Applications*).
- Servidor httpd de CERN

## C.2 ¿Qué servidor se utiliza en el Departamento de Supercómputo?

El servidor de HTTP elegido para ser utilizado en el Departamento de Supercómputo es el `httpd` de NCSA. Este es uno de los servidores más “evolucionados” existentes en dominio público, y sus mejores características son su excelente manejo de formas de HTML y sus mecanismos de control de acceso.

## C.3 Configuración del servidor de HTTP

El servidor de HTTP del Departamento de Supercómputo fue creado inicialmente para proporcionar al público información sobre el supercómputo y las aplicaciones que se le dan en la UNAM, así como temas relacionados. Sin embargo, al observar el increíble potencial de WWW como medio de difusión y recolección de información, sus aplicaciones se han ido ampliando.

Actualmente el servidor contiene tres áreas principales:

- Página *home* de la DGSCA.
- Página *home* del Departamento de Supercómputo.
- Página *home* del Laboratorio de Visualización.

Actualmente no se cuenta con restricciones de acceso en ninguna de estas áreas principales. Además, el servidor tiene la capacidad de manejar páginas *home* de cada usuario en forma individual. En estos casos, el usuario tiene control absoluto sobre lo que aparece en su página.

El servidor de HTTP fue implementado inicialmente en *ds5000*, donde reside hasta el momento. Sin embargo, en un futuro cercano será movido, al igual que los otros dos servidores mencionados, a *mezcal*. Se utilizó un registro MX del servidor de nombres para evitar problemas en este cambio. El nombre utilizado fue `www.super.unam.mx`, y actualmente apunta a *ds5000*, pero puede ser cambiado en el futuro, y las personas que utilicen dicho nombre para llegar al servidor no notarán el cambio.

## C.4 Cómo se usa el servidor de HTTP

WWW es el servicio de Internet más fácil de utilizar. Basta con tener algún visualizador de WWW, como NetScape [Cor], Mosaic [fSAb] o Lynx [oK], para poder comenzar a navegar en WWW.

En WWW, la ubicación de un documento se especifica mediante un URL (*Universal Resource Locator*) [fSAa]. Para llegar al servidor del Departamento de Supercómputo, el URL apropiado es `http://www.super.unam.mx/`. Esto desplegará una página de WWW en la que están ligas a las diferentes áreas contenidas en este servidor.

## Apéndice D

# Herramientas de seguridad de dominio público

En este apéndice se describen las principales herramientas de seguridad existentes en el dominio público (accesibles en Internet). Se encuentran clasificadas por áreas de acción, y para cada una de ellas se incluye una breve descripción y el sitio de FTP anónimo en donde puede ser obtenida.

### D.1 Autenticación

#### anlpasswd

Un programa de cambio de *passwords* que impide que el usuario escoja *passwords* débiles.

**Autor:** Laboratorio Nacional de Argonne.

**Disponible en:** `ftp://ftp.super.unam.mx/pub/security/tools/anlpasswd.tar.gz`

#### Crack

El mejor programa de “rompimiento” de *passwords* existente. Intenta adivinar los *passwords* utilizando una serie de reglas configurables.

**Autor:** Alec D. Muffett.

**Disponible en:** `ftp://ftp.super.unam.mx/pub/security/tools/crack.tar.gz`

#### Cracklib

Una biblioteca de funciones que pueden ser utilizadas para impedir que los usuarios elijan *passwords* que podrían ser adivinados por Crack.

**Autor:** Alec Muffett.

**Disponible en:** `ftp://ftp.super.unam.mx/pub/security/tools/cracklib.tar.gz`

### Kerberos

Un sistema de autenticación para redes físicamente inseguras, basado en el modelo de distribución de llaves presentado por R.M. Needham y M.D. Schroeder [NS78]. Permite a los elementos que intervienen en una comunicación identificarse entre sí y al mismo tiempo evitar “espionaje” en la red o ataques de repetición. También proporciona integridad en el flujo de datos (detección de modificaciones) y privacidad, utilizando sistemas criptográficos como DES. *Kerberos tiene restricciones de exportación hacia fuera de los Estados Unidos.*

**Disponible en:** `ftp://athena-dist.mit.edu/pub/kerberos/`

### npasswd

Un programa de cambio de *passwords* que impide que el usuario escoja *passwords* débiles. Incluye soporte para los mecanismos de envejecimiento de *passwords* de System V Release 3 y NIS (*Network Information Service*).

**Autor:** Clyde Hoover.

**Disponible en:** `ftp://ftp.cc.utexas.edu/pub/npasswd/`

### passwd+

Un programa de cambio de *passwords* que impide que el usuario escoja *passwords* débiles. El rechazo de *passwords* se basa en un archivo de configuración que permite la utilización de expresiones regulares, comparación con diccionarios o la ejecución de programas externos para examinar el password.

**Autor:** Matt Bishop.

**Disponible en:** `ftp://ftp.super.unam.mx/pub/security/tools/passwd+.tar.gz`

### PGP

*Pretty Good Privacy*, es un programa que permite realizar cifrado de mensajes y de archivos, utilizando una combinación de criptografía de llave privada (algoritmo IDEA), de llave pública (algoritmo RSA) y de verificación de mensajes (*Message Digest*, algoritmo MD5). También permite autenticar la procedencia de un mensaje mediante la utilización de firmas electrónicas.

**Autor:** Philip Zimmermann.

**Disponible en:** `ftp://ftp.super.unam.mx/pub/security/tools/PGP/`

### pidentd

Una implementación del servidor de identificación de usuario descrito en *RFC1413* [St.93], que permite averiguar la identidad del usuario que está solicitando un servicio remoto.

**Autor:** Peter Eriksson.

**Disponible en:** `ftp://ftp.super.unam.mx/pub/security/tools/pidentd.tar.gz`

### shadow

Un paquete que permite a cualquier sistema hacer uso de *shadow passwords*.

**Autor:** John F. Haugh, II.

**Disponible en:** `ftp://ftp.super.unam.mx/pub/security/tools/shadow.tar.gz`

**S/Key**

Un sistema que implementa *passwords* desechables para Unix. También incluye generadores de *passwords* desechables para PC's y Mac's.

**Autores:** Phil Karn, Neil M. Haller y John S. Walden.

**Disponible en:** `ftp://ftp.super.unam.mx/pub/security/tools/skey/`

**sra**

Un paquete que proporciona autenticación utilizando SECURE RPC para FTP y telnet.

**Autor:** Dave Safford.

**Disponible en:** `ftp://ftp.super.unam.mx/pub/security/tools/TAMU/`

## D.2 Firmas criptográficas

**MD5**

La especificación y código fuente para la función *Message Digest 5*.

**Autor:** Ronald L. Rivest.

**Disponible en:** `ftp://ftp.super.unam.mx/pub/security/tools/md5/`

**PGP**

Ver la sección D.1 en la página 139.

**Snefru**

Documentación y código fuente para la función *Message Digest Snefru (Xerox Secure Hash Function)*.

**Disponible en:** `ftp://ftp.super.unam.mx/pub/security/tools/snefru/`

## D.3 Seguridad en red

**ipacl**

Un sistema que implementa listas de control de acceso para servicios TCP o UDP. Todas las peticiones de este tipo de servicios pueden ser aceptadas, negadas o adaptadas en base a características como su origen, su destino, el tipo de petición, etc.

**Autor:** Siemens.

**Disponible en:** `ftp://ftp.win.tue.nl/pub/security/tools/ipacl.tar.Z`

**logdaemon**

Un paquete que proporciona versiones modificadas de **rshd**, **rlogind**, **ftpd**, **rexecd**, **login** y **telnetd** que registran mucha más información que los normales, permitiendo un mejor monitoreo y seguimiento de problemas de seguridad. También incluye soporte para S/Key.

**Autor:** Wietse Venema.

**Disponible en:** `ftp://ftp.super.unam.mx/pub/security/tools/logdaemon.tar.gz`

#### portmap

Un reemplazo para el programa `portmap` estandar de Unix, que cierra muchos de los huecos de seguridad existentes en este programa.

**Autor:** Wietse Venema.

**Disponible en:** `ftp://ftp.super.unam.mx/pub/security/tools/portmap.shar.gz`

#### SATAN

La última y más controvertida herramienta de seguridad, **SATAN** (*Security Administrator Tool for Analyzing Networks*) permite revisar sistemas Unix de forma remota en búsqueda de la existencia de diversos huecos de seguridad. Ofrece una interfaz gráfica al usuario, lo que hace muy fácil la revisión de los reportes generados.

**Autores:** Dan Farmer y Wietse Venema.

**Disponible en:** `ftp://ftp.super.unam.mx/pub/security/tools/satan.tar.gz`

#### TCP-Wrapper

Uno de los paquetes de seguridad más utilizados, permite monitorear y controlar el acceso a servicios de red en sistemas Unix, tales como FTP, **telnet**, **rsh**, **finger**, etc.

**Autor:** Wietse Venema.

**Disponible en:** `ftp://ftp.super.unam.mx/pub/security/tools/tcp_wrappers.tar.gz`

#### Xinetd

Un reemplazo de **inetd**, el *daemon* de servicios de red en Unix. Permite monitorear y controlar el acceso a todos los servicios de red proporcionados por **inetd**.

**Disponible en:** `ftp://qiclab.scn.rain.com/pub/security/tools/xinetd-2.1.1.tar.gz`

## D.4 Monitoreo de red

#### Courtney

Permite identificar posibles ataques realizados con **SATAN** y la máquina de la que proceden.

**Disponible en:** `ftp://ftp.super.unam.mx/pub/security/tools/courtney.tar.gz`

#### Netlog

Sistema de monitoreo de servicios TCP y UDP.

**Disponible en:** `ftp://ftp.super.unam.mx/pub/security/tools/TAMU/`

## D.5 Monitoreo del sistema

### COPS

El *Computer Oracle and Password System* (COPS) es otro de los paquetes “clásicos” de la seguridad en Unix. Consta de un conjunto de programas que revisan diferentes aspectos de la seguridad del sistema, reportando los posibles huecos de seguridad encontrados.

**Autor:** Dan Farmer.

**Disponible en:** `ftp://ftp.super.unam.mx/pub/security/tools/cops.tar.gz`

### cpm

*Check Promisuous Mode* (cpm) es un programa que revisa la existencia en el sistema de interfaces de red en modo promiscuo, lo cual puede ser indicativo de que algún intruso ha entrado al sistema y lo ha activado para capturar paquetes de red, o de un grave error de configuración.

**Autor:** Carnegie Mellon University.

**Disponible en:** `ftp://ftp.super.unam.mx/pub/security/tools/cpm.tar.gz`

### ISS

*Internet Security Scanner* (ISS) es un programa que revisa un sistema Unix en búsqueda de ciertos huecos de seguridad.

**Autor:** Christopher Klaus.

**Disponible en:** `ftp://ftp.super.unam.mx/pub/security/tools/iss.tar.gz`

### RIACS

RIACS (*Research Institute for Advanced Computer Science*) *Intelligent Auditing and Categorizing System* (RIACS) es un sistema que monitorea las bitácoras del sistema y reporta los cambios.

**Autor:** Matt Bishop.

**Disponible en:** `ftp://coast.cs.purdue.edu/pub/tools/unix/binaudit.tar.gz`

### Spar

*Show Process Accounting Records* (Spar) es un programa que analiza y despliega los registros de contabilidad de procesos de un sistema Unix, de forma mucho más flexible que los programas estándar como **lastcomm**.

**Autor:** Dough Schales.

**Disponible en:** `ftp://ftp.super.unam.mx/pub/security/tools/TAMU/`

### Swatch

Un sistema para monitorear los archivos de bitácoras del sistema y ejecutar acciones específicas en respuesta a ciertos patrones encontrados.

**Autor:** Todd Atkins.

**Disponible en:** `ftp://ftp.super.unam.mx/pub/security/tools/swatch.tar.gz`

### Tiger

Un paquete de monitoreo del sistema similar a COPS, pero está en constante desarrollo.

**Autor:** Dough Schales.

**Disponible en:** `ftp://ftp.super.unam.mx/pub/security/tools/TAMU/`

### TripWire

Un paquete que revisa la integridad de un sistema Unix utilizando firmas criptográficas; primero genera una lista de las firmas correspondientes a los archivos revisados (supuestamente “limpios”), y en corridas posteriores recalcula las firmas de dichos archivos y las compara con las almacenadas, reportando las diferencias encontradas.

**Autores:** Eugene Kim y Gene Spafford.

**Disponible en:** `ftp://ftp.super.unam.mx/pub/security/tools/tripwire.tar.gz`

### Watcher

Un sistema de monitoreo que ejecuta órdenes configurables, analiza los resultados, y reporta los aspectos importantes al administrador del sistema.

**Autor:** Kenneth Ingham.

**Disponible en:** `ftp://ftp.super.unam.mx/pub/security/tools/watcher.tar.gz`

## D.6 Herramientas generales de seguridad

### Dig

Una herramienta de consulta de DNS, semejante a la orden de Unix **nslookup**, pero más flexible y sencilla.

**Autores:** Steve Hotz y Paul Mockapetris.

**Disponible en:** `ftp://ftp.super.unam.mx/pub/security/tools/dig.tar.gz`

### NFS Watch

Una herramienta que permite monitorear los paquetes de NFS que circulan por la red local, y realizar diversos análisis sobre ellos.

**Autores:** David A. Curry y Jeff Mogul.

**Disponible en:** `ftp://ftp.super.unam.mx/pub/security/tools/nfswatch.tar.gz`

### rdist

El programa **rdist** de Unix ha sido la causa de muchísimos problemas de seguridad a lo largo de los años. Esta distribución de la Universidad de California es un reemplazo para dicho programa, que introduce múltiples mejoras y corrige todos los huecos de seguridad conocidos hasta el momento.

**Autor:** Michael Cooper.

**Disponible en:** `ftp://ftp.super.unam.mx/pub/security/tools/rdist.tar.gz`

### sendmail

El mundialmente utilizado programa `sendmail` de Unix, el encargado de enviar y recibir el correo electrónico, ha sido también causante de muchísimos huecos de seguridad. Muchos fabricantes de Unix incluyen en sus distribuciones del sistema operativo versiones viejas del `sendmail`, que contienen múltiples huecos de seguridad. Las últimas versiones corrigen todos los problemas conocidos, por lo que su instalación es muy recomendada.

**Autor:** Eric Allman.

**Disponible en:** `ftp://ftp.super.unam.mx/pub/security/tools/sendmail/`

### tcpdump

Un programa que captura paquetes de red, los almacena y despliega su contenido, filtrando solamente las partes de interés. Es requerido para utilizar `Courtesy`.

**Autor:** Van Jacobson.

**Disponible en:** `ftp://ftp.super.unam.mx/pub/security/tools/tcpdump.tar.gz`

### Traceroute

Un programa que permite monitorear la ruta que siguen los paquetes de IP desde un sistema a otro.

**Autor:** Van Jacobson.

**Disponible en:** `ftp://ftp.super.unam.mx/pub/security/tools/traceroute.tar.gz`

### wu-ftpd

El `daemon` de FTP de la Universidad de Washington es un sustituto del `daemon` de FTP incluido con Unix. Proporciona seguridad adicional y muchas mejoras, como control de acceso, compresión y descompresión de archivos “al vuelo”, etc.

**Autor:** Bryan D. O’Connor.

**Disponible en:** `ftp://ftp.super.unam.mx/pub/net/wu-ftpd-2.4.tar.gz`



## Apéndice E

# Acciones iniciales a tomar sobre seguridad

Este apéndice contiene el documento sobre acciones iniciales de seguridad, que fue distribuido a los asistentes a la primera junta de GASU, así como a directivos y administradores de sistemas Unix en diversas dependencias de la UNAM.

**Nota importante:** Este fue el primer documento sobre medidas de seguridad que se elaboró en el proyecto de seguridad en la UNAM. Posteriormente a su distribución muchas medidas fueron modificadas, así como refinadas las recomendaciones y la información distribuida. Mucha de la información presente en este documento ya no es válida al momento de finalizar esta tesis.

---

Con la finalidad de intentar dar solución al problema de seguridad en las máquinas Unix se pide a los encargados de los sistemas realicen los siguientes puntos:

### E.1 Acciones Correctivas

- Realizar un respaldo completo del sistema, por si es necesario revisar.
- Reinstalación del sistema operativo de la distribución original, no de respaldos.
- Revisión de todos los `.rhosts`. De ser posible, eliminar todos los que no sean absolutamente indispensables (como para la ejecución de procesos automáticos).
- Revisión de `/etc/hosts.equiv`. De ser posible, eliminarlo.

### E.2 Claves

- Revisión MANUAL de las cuentas existentes, para ver que todas sean autorizadas y utilizadas regularmente. Las cuentas que no se estén utilizando borrarlas o

desactivarlas. Para desactivarlas no se les pone un \* en el *password*, sino se les pone el *nullshell* en el campo del *shell*. De esta manera se lleva un registro de los intentos de acceso a dichas claves.

- Abrir una cuenta llamada **dgsc**a para uso de auditoría de seguridad.
- No permita que haya claves sin contraseña. Lleve un control de las claves que se asignan y evite la creación de claves no autorizadas o cambios no autorizados a los nombres de las claves.
- Instalación del **passwd+**, de manera que sustituya totalmente al `/bin/passwd`. Dejar una copia del `/bin/passwd` original, pero sin permiso de lectura ni ejecución para nadie, y con otro nombre. Crear `/bin/passwd` como una liga a `/bin/passwd+` (o donde sea que se ponga el **passwd+**).
- Para evitar intrusos elija *passwords* difíciles de adivinar y tenga en cuenta lo siguiente:
  - Use palabras que no se encuentren en ningún diccionario de cualquier idioma.
  - No use nombres propios, incluyendo nombres de personajes famosos.
  - No use acrónimos comunes en cómputo.
  - No use palabras que tengan alguna relación con usted o su trabajo.
  - Un buen método para elegir una contraseña es tener una frase fácil de recordar (un verso, una frase célebre, etc.). Por ejemplo, en la frase “*No por mucho madrugar...*”, tome las primeras letras de cada palabra, inserte algunos signos de puntuación y trate de mezclar minúsculas y mayúsculas. En la frase anterior, una buena contraseña sería: **N{p}mM@**.
- Revise los permisos de los directorios principal de usuarios (**rwxr-x--**) y los directorios *home* (**rwX----**), de correo electrónico y utilerías reservadas.
- Negar el permiso de realizar FTP a las claves del sistema, como **sys**, **bin**, etc. Para ello se utiliza el archivo `/etc/ftpusers`.
- Implementar un retraso en el *login* de 3 segundos mínimo para que quienes intenten infiltrarse probando múltiples contraseñas con *N* combinaciones tarden  $3 * N$  segundos más que con un *login* normal. Tres segundos son tolerables para un *login* de un usuario autorizado.

### E.3 Programas de seguridad

- En los que sean máquinas Ultrix u otras que no tengan un **syslog** “decente”, instalar el `jdkohl syslog`, que es un **syslog** de dominio público que implementa las funciones de **syslog** como deben de ser. Esto es necesario porque **passwd+**, **TCP-Wrapper** y otros programas utilizan el **syslog** para llevar el registro de sus actividades.

- Utilice herramientas para determinar la integridad del sistema, este *software* está disponible vía FTP anónimo.  
**COPS** Ayuda a encontrar huecos de seguridad. `cert.org:/pub/tools/cops` (192.88.209.5).  
**passwd+** Permite al administrador reforzar el sistema de *passwords*. `dartmouth.edu:/pub/passwd+.tar.Z` (129.170.16.4).  
**TCP Wrapper** Ayuda al administrador a negar el acceso desde otros sistemas o dominios. `cert.org:/pub/tools/tcp_wrapper` (192.88.209.5).
- Instalación del nuevo **rshd**, que lleva un registro de las sesiones realizadas vía **rsh**.
- Ejecutar el *autologout* en **cs**h (primero hacer pruebas).
- Buscar cuáles comandos tienen el bit SUID encendido:  

```
# find / -perm -4000 -print {} \;
```
- Si su sistema tiene herramientas de seguridad integradas, impleméntelas.

## E.4 Comunicaciones

- Evitar NFS a segmentos locales desde el exterior.
- Revisar si **fingerd** no está accesible a los usuarios.
- Restringir el uso de TFTP y UUCP para evitar que extraños obtengan una copia del archivo `/etc/passwd`.
- Verificar que **root** no pueda entrar por red directamente.

## E.5 Procedimientos

- No dejar su terminal o estación de trabajo en sesión sin estar presente.
- Suscribirse al CERT y obtener los programas para revisión de la seguridad, entre ellos realizar la instalación de **COPS** y **Crack**. Ejecución periódica (en **crontab**) de los mismos, y envío de los reportes tanto al administrador de la máquina como a nosotros en casos difíciles.
- Reunión bimensual para revisar cosas que han ocurrido, notificar de nuevas medidas, programas, etc. Creación de un grupo de seguridad y un mecanismo de comunicación via red entre todos los administradores.
- Revisar a fondo los programas que se están ejecutando en el **crontab** de **root**.

- Poner la máscara de permisos de los usuarios (**umask**) a **077** en el `/etc/profile` y en el `/etc/cshrc`.
- Eliminar a los usuarios el permiso de ejecución de **crontab** y **at**, y dejarlos ejecutarlos solamente si lo solicitan.
- Revisar si hay archivos de **root** en otras claves y de preferencia eliminar la creación de archivos de **root** en los sistemas de archivos de usuarios, por medio de las cuotas de disco.
- Si sospecha de algún intruso, debe verificar la integridad del sistema operativo, de ser posible copie los archivos binarios de algún respaldo original. Verifique los permisos de su sistema de archivos. Verifique que las claves de usuario no tengan permisos de superusuario. Cambie las contraseñas de los usuarios. Basta con poner un `*` en el campo de contraseña y dejarle como `shell /bin/false` hasta que su sistema sea seguro.

## E.6 Bibliografía

- *Unix System Security*, David A. Curry. Addison-Wesley, 1992.
- *Site Security Handbook*, P. Holbrook, J. Reynolds. Network Working Group, Request for Comments (RFC) 1244, FYI 8, July 1992. Disponible via FTP anónimo en `funet.fi`.

Preguntas a Diego Zamboni: `diego@ds5000.dgsca.unam.mx`

## Apéndice F

# Manual de instalación de COPS, TCP-Wrapper y passwd+

Este apéndice contiene el documento entregado a los asistentes a la primera reunión de GASU el 6 de Diciembre de 1995, que contiene instrucciones detalladas de instalación de las herramientas de seguridad COPS, TCP-Wrapper y passwd+.

Nota aclaratoria: este documento contiene a su vez un apéndice, dado que originalmente fue distribuido como un documento independiente.

---

Para aclaraciones, correcciones o resolución de dudas sobre este documento, comunicarse con Diego Zamboni al 622-85-29, o por correo electrónico a [diego@ds5000.dgsca.unam.mx](mailto:diego@ds5000.dgsca.unam.mx).

### F.1 Introducción

El sistema operativo Unix es altamente complejo, por lo que es imposible para los diseñadores prever todas las situaciones que se pueden llegar a presentar. Por otro lado, en muchas ocasiones los administradores y/o los usuarios no tienen plena conciencia de la importancia de la seguridad, o no tienen el conocimiento y el tiempo necesarios para preocuparse por dicho aspecto. Estos dos factores conducen al hecho innegable de que, en general, la seguridad de los sistemas Unix es bastante débil.

Esto, al contrario de lo que mucha gente quiere hacer creer, no se debe tanto a debilidades o deficiencias en el sistema operativo, sino a malas prácticas de administración y uso, provocadas por ignorancia, falta de tiempo, de recursos o de interés.

Son tantos los factores que tienen que ver con la seguridad de un sistema Unix que es prácticamente imposible para una persona estar vigilando continuamente y de manera eficiente todos los “rincones” del sistema para ver si no hay nada sospechoso. Es por esto que son de gran ayuda los programas que permiten realizar parte de esa vigilancia

de manera automática, reportando a alguna persona los hallazgos realizados, sobre todo cuando hay algo poco usual en ellos que pudiera indicar un posible hueco de seguridad.

Este documento contiene una breve descripción de las principales herramientas que un administrador puede utilizar en su tarea de vigilar y mejorar la seguridad general del sistema a su cargo. Para cada uno de los programas se da una breve descripción de su funcionamiento, y a continuación los pasos necesarios para su instalación en un sistema Unix típico.

**Nota Importante:** Este documento solamente pretende ser una guía rápida para la instalación de los programas descritos. Para una descripción más detallada de la instalación, uso y posibles opciones de los mismos, es recomendable consultar el documento README o INSTALL que se incluye en todos los programas. Antes de comenzar el proceso de instalación es bueno imprimir una copia de dicho documento y leerlo completo, junto con los pasos indicados en esta guía. Hay que recordar que al estar trabajando bajo una cuenta privilegiada como **root** es posible hacer mucho daño si no se procede cuidadosamente, de manera que es vital revisar varias veces lo que se va a hacer antes de efectivamente llevarlo a cabo.

Los procedimientos de instalación descritos son los necesarios para instalar los programas en un sistema Unix típico. Desgraciadamente, estos sistemas son casi inexistentes, pues cada versión de Unix tiene sus particularidades que afectan la manera en que un programa debe ser configurado e instalado. Por lo tanto, se recomienda nuevamente leer con detalle la documentación incluida en cada programa, para tomar nota de cualquier posible acción especial que sea necesario tomar antes de instalar el programa siguiente el método tradicional, o durante dicha instalación.

**ADVERTENCIA:** Varias de las herramientas descritas tienen que ser instaladas y/o ejecutadas en última instancia por **root**. Sin embargo, NUNCA se debe realizar la compilación y pruebas del programa bajo dicha cuenta, pues root tiene la capacidad de causar daño casi ilimitado en el sistema si algo sale mal. Siempre que sea posible se debe realizar la compilación y pruebas del programa bajo alguna cuenta no privilegiada, y utilizar **root** solamente a la hora de instalarlo de manera definitiva.

## F.2 Obtención de los Programas

Todas las herramientas descritas en este documento pertenecen al dominio público, por lo cual son accesibles a cualquiera que desee tenerlas. Para facilitar su obtención a los administradores de la UNAM, hemos dispuesto un servicio de FTP anónimo, en el cual se pueden obtener todos estos programas y algunos otros de interés general. Los pasos a seguir son los siguientes:

1. Ejecutar **ftp ftp.super.unam.mx**. Si se produce un error de no ubicación del nombre, intentar **ftp 132.248.204.8**.
2. A la pregunta de login, contestar con **anonymous**.

3. A la petición de password, contestar con su dirección electrónica completa (por ejemplo, `diego@ds5000.dgsca.unam.mx`).
4. Ejecutar **cd pub/security**.
5. Ejecutar **dir**. Con esto se obtendrá el listado de todos los programas disponibles en el área de seguridad.
6. Ejecutar **bin** para establecer el modo binario de transferencia de archivos.
7. Ejecutar **get <nombre>**, donde **<nombre>** es el nombre del archivo correspondiente al programa que se quiere transferir.
8. Repetir el paso anterior para todos los archivos que se quieran obtener.
9. Una vez finalizada la transferencia de los archivos, ejecutar **quit** para terminar la conexión.

**Nota:** Todos los programas disponibles en el FTP anónimo de `ftp.super.unam.mx` están comprimidos con el programa de compresión **gzip**, que le pone a los archivos que comprime la extensión `.gz`. Este programa es utilizado por la principal razón de que proporciona mejor compresión que el `compress` de Unix, además de que es software de dominio público. El **gzip** también se encuentra en `ftp.super.unam.mx`, en el archivo `/pub/gnu/gzip-1.2.4.tar.Z`. En caso de no tenerlo todavía, es necesario que también obtenga este archivo y lo instale siguiendo las instrucciones que se encuentran en el archivo `INSTALL`. En este paquete viene también una nueva versión de **zcat**, cuya funcionalidad es equivalente al comando de Unix del mismo nombre, pero con la capacidad de leer archivos creados con **compress**, **pack** o **gzip** de manera automática. A este comando es al que se hace referencia en las instrucciones de instalación.

**Nota para administradores de sistemas DEC (Ultrix):** Algunas de las herramientas descritas hacen uso del programa **syslog** de Unix, que permite el registro de manera automática de mensajes generados por los procesos, de acuerdo a una configuración global del sistema. En Ultrix, el **syslog** no se apega al estándar, por lo que no proporciona toda la funcionalidad necesaria para que algunos programas puedan funcionar correctamente. Antes de instalar en un sistema Ultrix algún programa que haga uso de **syslog**, se recomienda leer el Apéndice de este documento y realizar el proceso de instalación de una nueva versión de **syslog**.

## F.3 Descripción de las Herramientas

### F.3.1 COPS

#### Descripción

COPS (*Computer Oracle and Password System*) es una colección de programas, cada uno de los cuales intenta cubrir un área de seguridad distinta. Los puntos principales que se cubren son los siguientes:

- Permisos de archivos, directorios y dispositivos.
- Passwords débiles (fácilmente adivinables).
- Formato, contenido y seguridad de los archivos de passwords y de grupos.
- Los programas y archivos utilizados en los archivos `/etc/rc*` (de inicialización del sistema) y `cron` (tareas ejecutadas periódicamente).
- Existencia de archivos con el bit SUID activado, su modificabilidad, y si son o no guiones de shell.
- Una verificación de cambios en los archivos binarios más importantes.
- Permisos de los directorios `home` de los usuarios y los archivos de arranque (`.profile`, `.cshrc`, `.login`, etc.).
- Configuración del servicio de ftp anónimo, en caso de existir.
- Existencia de TFTP sin restricciones, el alias `decode` en `sendmail`, problemas con **uudecode**, ejecución de shells en `inetd.conf` y activación de **rexid(8)** en `inetd.conf`.
- Verificación de varios aspectos de la cuenta de root: el directorio actual en la ruta de búsqueda de programas, un “+” en `/etc/hosts.equiv`, directorios exportados via NFS sin restricciones, etc.
- Fechas de algunos avisos del CERT (Computer Emergency Response Team) contra algunos archivos clave del sistema. Esta parte verifica las fechas en que algunos huecos de seguridad fueron reportados contra la fecha de los archivos involucrados. Si el archivo es anterior al aviso, es señal de que puede estar presente dicho hueco, aunque no necesariamente. Sin embargo, si se recibe un aviso positivo, siempre es buena idea obtener el avisador del CERT y revisarlo para obtener más pistas. Por supuesto, si el resultado es negativo, no es indicativo de que el hueco no exista, simplemente que el archivo en cuestión ha sido modificado de alguna manera en una fecha posterior al aviso.

Todos los programas solamente avisan al usuario de los problemas potenciales. COPS no intenta explotar ninguno de los problemas potenciales que encuentra. Solamente envía un mensaje de correo electrónico o crea un archivo conteniendo el reporte de los problemas encontrados.

Como COPS no hace ningún intento de corregir los problemas, no es necesario ejecutarlo con ninguna clase de privilegios especiales. El único programa que es necesario ejecutar con privilegios de root para obtener los mejores resultados es la búsqueda de archivos SUID (para poder revisar todos los subdirectorios existentes) y a veces la verificación de cambios en los ejecutables.

COPS proporciona un método de encontrar errores y malas configuraciones comunes. De ninguna manera puede reemplazar a un administrador con sentido común y que esté alerta a lo que sucede en el sistema. Más que otra cosa, COPS puede ayudar al administrador a protegerse contra la ignorancia, la falta de cuidado y la ocasional acción malintencionada, tanto por parte de los usuarios como de los mismos administradores.

### Archivos de documentación importantes

README.FIRST, README.1, README.2sh, README.2pl, README.3, quickstart.

### Instalación

1. Desempaquetar el archivo `cops-104.tar.gz` con el comando **zcat cops-104.tar.gz | tar xvf -**. Esto creará en el directorio actual un subdirectorio llamado `cops-104`, que contiene todos los archivos correspondientes a este programa.
2. Entrar al directorio `cops-104`.
3. Ejecutar el comando **./reconfig** para establecer apropiadamente las rutas de los comandos utilizados por COPS.
4. Ejecutar el comando **make all** para compilar los programas de C necesarios y crear las páginas de manual.
5. Consultar el archivo `README.2sh` en caso de surgir algún problema durante la compilación.
6. Las líneas 93 y 94 del archivo `cops` contienen los nombres del directorio donde está almacenado el COPS y de la clave a la que hay que enviar los reportes. Originalmente son como sigue:

```
SECURE=/usr/foo/bar  
SECURE_USERS="foo@bar.edu"
```

Es necesario cambiar estos valores a los correctos antes de ejecutar COPS por primera vez. *SECURE* debe ser el directorio que contiene los programas de COPS, y *SECURE\_USERS* debe ser su propia dirección electrónica o la de quien sea designado para recibir los reportes.

7. Ejecutar el COPS con el comando **cops -v**. Después de terminar, deberá existir en el directorio del COPS un directorio con el nombre de la máquina en la que se ejecutó, y dentro de él un archivo que contiene el reporte generado. Consultar la documentación del COPS para mayor información en cómo interpretar dichos reportes y como continuar la utilización del COPS de manera periódica.

## F.3.2 TCP-Wrapper

### Descripción

TCP-Wrapper (**tcpd**) es un programa que permite controlar y monitorear el acceso a varios de los servicios de red ofrecidos por Unix, como **telnet**, **ftp**, **finger**, **talk**, **rlogin** y **rsh**. Una vez instalado el **tcpd**, toda petición de servicio es registrada con fecha, hora, servicio solicitado y máquina desde la cual se hizo la solicitud. Posteriormente al registro se consultan unas tablas de control de acceso que determinan, en base al tipo

de servicio y dirección electrónica de la máquina solicitante, si dicho servicio se debe proporcionar. Finalmente, tanto si el servicio se proporciona como si no, es posible ejecutar algún comando en respuesta a dicho intento de acceso.

En Unix, casi todos los servicios de red son proporcionados por un programa llamado **inetd**(8), que se arranca al inicializar el sistema y está corriendo de forma continua. Cuando el **inetd** detecta alguna petición de servicio, ejecuta el servidor adecuado, tal como **telnetd**(8), **ftpd**(8), etc. El funcionamiento del **tcpd** se basa en modificar el archivo de configuración del **inetd** (`/etc/inetd.conf`) de manera que cuando reciba alguna petición, en vez de ejecutar directamente el servidor correspondiente, se ejecute el **tcpd**, pasándole como argumento el nombre del servicio solicitado. El **tcpd** hará el registro de la petición de servicio, consultará las tablas de acceso y, si se debe proporcionar el servicio, arrancará el servidor apropiado. En caso de que se determine que el servicio no debe proporcionarse, el **tcpd** simplemente corta la conexión. En cualquier caso, opcionalmente arrancará un comando a ejecutar como respuesta a la petición de servicio.

### Archivos de documentación importantes

README.

### Instalación

**Nota:** En la documentación del **tcpd** se describen dos tipos de instalación: fácil y avanzada. En esta guía se describe la instalación avanzada por considerarse la más apropiada y, aunque parezca contradictorio, más sencilla de realizar.

**Nota para sistemas Ultrix:** Este programa hace uso de **syslog**(8), de manera que es necesario instalar una nueva versión de **syslog** antes de instalar el TCP-Wrapper. Consultar el Apéndice de este documento.

1. Crear un directorio llamado `tcp_wrapper`, donde se guardarán los archivos de este programa.
2. Cambiarse a ese directorio y ejecutar `zcat tcp-wrapper.tar.gz | tar xvf -`. Con esto se desempaquetarán en el directorio actual los archivos del **tcpd**.
3. Revisar el archivo `/etc/inetd.conf` para consultar en qué directorio están ubicados los servidores, cuyos nombres normalmente son **telnetd**, **ftpd**, etc., o **in.telnetd**, **in.ftpd**, etc. El directorio en el que están localizados normalmente es `/etc` o `/usr/etc`.
4. En las líneas 34-44 del archivo `Makefile` aparecen distintas opciones para el directorio obtenido en el punto anterior. Editar dicho archivo y dejar activado (quitarle el # al principio de la línea) el valor apropiado.
5. En la línea 274 del `Makefile`, cambiar la cadena `LOG_MAIL` por `LOG_LOCAL0`. Esto es con el fin de que los mensajes generados por el **tcpd** sean registrados por separado de los generados por el **sendmail**, de manera que se pueda hacer un registro y análisis más fácilmente.

6. Ejecutar **make** y seguir las instrucciones que aparecen en la pantalla.
7. Al finalizar la compilación existirá en el directorio actual el ejecutable **tcpd**, que es el que implementa toda la funcionalidad del TCP-Wrapper. En sistemas Ultrix se creará un segundo ejecutable llamado **miscd** que implementa algunas funciones propias de Ultrix.
8. Copiar el ejecutable **tcpd** al directorio donde se encuentran los servidores del **inetd** (probablemente `/etc` o `/usr/etc`). En sistemas Ultrix, copiar el ejecutable **miscd** al directorio `/usr/local/etc`.
9. Hacer una copia de seguridad del archivo `/etc/inetd.conf`. Hay sistemas en donde este archivo no se encuentra en el directorio `/etc` sino en algún otro, como `/usr/etc`. A partir de ahora se hará referencia a este archivo solamente como `inetd.conf`, sin importar el directorio en el que se encuentre.
10. Editar el archivo `inetd.conf`. En este archivo hay que ubicar las referencias a los programas que proporcionan los servicios. Todas las referencias a las rutas de estos programas hay que cambiarlas por la ruta del **tcpd**, dejando las demás opciones sin modificar. En sistemas Ultrix, hay que cambiar todas las referencias a `/etc/miscd` por `/usr/local/etc/miscd`.
11. Grabar el archivo con las modificaciones.
12. En el archivo de configuración del **syslog** (normalmente `/etc/syslog.conf` o `/etc/nsyslog.conf`), añadir la siguiente línea:

```
local0.info                /usr/local/adm/tcpd.log
```

donde `/usr/local/adm/tcpd.log` puede ser sustituido por la ruta del archivo donde se quiera llevar el registro de las actividades del **tcpd**.

13. Rearrancar el **inetd** y el **syslogd** para que entren en efecto los cambios hechos a los archivos de configuración. Esto se puede hacer con la instrucción **kill -HUP <n>**, donde **<n>** es el número del proceso que se quiera reinicializar.
14. Si todo está correcto, a partir de este momento el **tcpd** debe registrar todos los accesos a los servicios para los que se haya activado su ejecución. Puede probarse haciendo un **telnet** o **ftp** a la misma máquina y después revisando el archivo que se haya indicado en `syslog.conf` para recibir el registro.
15. Consultar la documentación del TCP-Wrapper para la información sobre los archivos de control de acceso y otros aspectos de la utilización del **tcpd**.

### F.3.3 passwd+

#### Descripción

El password es la primera línea de defensa de una cuenta contra los intentos de acceso no autorizado. Sin embargo, en muchas ocasiones la gente tiende a elegir passwords débiles, es decir, que son fácilmente adivinables, como el mismo nombre de la cuenta, el nombre o apellido de la persona, su fecha de cumpleaños, el nombre de su esposa o de algún hijo, etc. Esto hace mucho más sencillo para un atacante el acceso a alguna cuenta legítima del sistema, desde donde es mucho más fácil obtener privilegios mayores.

**passwd+** es un programa que reemplaza al **passwd** estándar de Unix, cuya función es permitirle al usuario cambiar su password. Sin embargo, el **passwd+**, a diferencia del **passwd** (con la excepción de algunas versiones), no permite la elección de passwords débiles, basándose en una serie de reglas definidas por el usuario. Por supuesto, esto no elimina los passwords débiles que ya existen en el sistema, pero impide que los nuevos usuarios o lo que cambien su password hagan una selección fácilmente adivinable. Como el archivo de reglas es modificable por el administrador del sistema, la verificación puede hacerse tan estricta u holgada como se desee.

**Nota:** La versión actual de **passwd+** todavía no maneja bien los sistemas con Yellow Pages/NIS. Si en su red se está utilizando este protocolo, es recomendable que no instale **passwd+** hasta que se disponga de una versión que lo maneje correctamente.

**Nota para sistemas Ultrix:** Este programa hace uso de **syslog**, de manera que es necesario instalar una nueva versión de **syslog** antes de instalar el TCP-Wrapper. Consultar el Apéndice de este documento.

#### Archivos de documentación importantes

README, README . IMPORTANT.

#### Instalación

1. Desempaquetar el archivo `passwd+.tar.gz` con el comando **zcat passwd+.tar.gz | tar xvf -**. Con esto se creará en el directorio actual un subdirectorio llamado `passwd+` que contiene todos los archivos de este programa.
2. Editar el archivo `sys.h`, donde se definen varios de los parámetros de operación del sistema. Se puede tomar como ejemplo alguno de los bloques ya existentes y modificarlo para que se ajuste al sistema en particular en el que se está realizando la instalación. El archivo contiene información detallada del significado de cada campo. En caso de que se tenga duda de la existencia de alguna función, un método para averiguarlo es tratar de obtener la página de manual de dicha función. Si la página de manual no existe, lo más probable es que tampoco se cuente con la función. Cabe hacer una aclaración con respecto a la variable *DBMLIB*: Para saber si el archivo de password es almacenado en este formato, basta ver si existen los archivos `/etc/passwd.dir` y `/etc/passwd.pag`. Esto indica que

el archivo de passwords es almacenado en formato DBM, y se debe dar un valor apropiado a *DBMLIB*.

3. Editar el archivo `passwd.h` y cambiar el valor de la variable *ROOTID* en la línea 32 al identificador numérico (UID) del usuario bajo cuya cuenta se está compilando el programa. Esto es con el fin de poder hacer pruebas antes de instalar el `passwd+` para uso de todos los usuarios.
4. Editar el archivo `Makefile` y modificar, en caso de ser necesario, los valores de las variables *SYSTYPE* (línea 35, de acuerdo a lo determinado en `sys.h`) y *DBMLIB* (línea 49; no definirlo si no se definió *DBMLIB* en `sys.h`).
5. Ejecutar **make all**.
6. Una vez terminada con éxito la compilación, existirá en el directorio actual el ejecutable **passwd**. Con este ejecutable se tienen que realizar las pruebas, para después recompilarlo con los parámetros correctos para ser usado por todos los usuarios. Los siguientes pasos se refieren a la manera recomendada para probarlo. Recuérdese que las pruebas no deben realizarse bajo la cuenta de root sino desde la cuenta del usuario que realizó la compilación del programa.
7. Copiar el archivo `/etc/passwd` al directorio del `passwd+`, con el nombre `passwd.data`.
8. Los archivos `pwsample` y `pwsample2` son archivos de reglas de verificación que se proporcionan como ejemplos. Copiar cualquiera de ellos al archivo `passwd.test` en el mismo directorio. Consultar el archivo `README` para información sobre el formato de dicho archivo.
9. Ejecutando el `passwd` generado en la compilación, intentar cambiar algunos passwords de otros usuarios usando el comando `./passwd usuario`, donde **usuario** es el nombre de alguna de las cuentas que se encuentran en `passwd.data`. Verificar que se realicen los cambios. Intentar introducir algunos passwords inválidos (como el mismo nombre de la cuenta) para corroborar que se estén respetando las reglas establecidas.
10. Una vez que se han realizado pruebas suficientes de forma satisfactoria, es el momento de modificar algunos parámetros para la operación “pública” del `passwd+`. Los siguientes pasos se refieren a este procedimiento.
  - (a) En el archivo `passwd.h`, hacer los siguientes cambios:
    - Cambiar el valor de la variable *DEFPWFILE* (línea 22) a `/etc/passwd`.
    - Cambiar el valor de la variable *PWTESTFILE* (línea 23) a `/etc/passwd.test`.
    - Cambiar el valor de la variable *LG\_OUTDEF* (línea 26) a `> /usr/adm/passwd.log`, donde la ruta del archivo se puede cambiar a donde se desee que quede el registro de las actividades del `passwd+`.

- Cambiar el valor de la variable *ROOTID* (línea 32) nuevamente a 0.
- (b) En el archivo `pwd.c`, hacer los siguientes cambios:
- Cambiar el valor de las variables *PF\_PLATE* y *PF\_TEMP* a `/etc/ptmpXXXXXX` y `/etc/ptmp`, respectivamente.
- (c) En el archivo `Makefile`, hacer los siguientes cambios:
- Insertar un `#` al principio de las líneas 41-44, y quitar el que se encuentra al principio de las líneas 37-40, para indicar la ubicación real de los archivos ejecutables, de *passwords* y de reglas de verificación, así como de registro de actividades.
- (d) Ejecutar **make all** para recompilar el programa con los nuevos parámetros.
- (e) Entrar a la cuenta de **root** para llevar a cabo la instalación definitiva del programa.
- (f) Hacer una copia del ejecutable `/bin/passwd` (o cualquiera que sea su ruta en el sistema en el que se está ejecutando) a `/bin/passwd.old`, quitándole todos los permisos de ejecución con el comando **chmod 700 /bin/passwd.old**. Con esto se conserva una copia de seguridad del programa `passwd` original, pero se elimina la posibilidad de que los usuarios lo utilicen.
- (g) Ejecutar el comando **make install** en el directorio del `passwd+`. Con esto se copia el ejecutable al lugar de `/bin/passwd`, y la página de manual correspondiente a su sitio. Es necesario verificar que el archivo `/bin/passwd` quede con permiso de ejecución para todos los usuarios.
- (h) Copiar el archivo `pwsample` o `pwsample2` (éste último contiene más reglas y los letreros están en español) a `/etc/passwd.test`. Posteriormente se puede editar dicho archivo para reflejar políticas de la administración del sistema, pero las reglas proporcionadas en `pwsample2` se consideran un buen inicio.
11. Añadir al archivo `/etc/syslog.conf` (o `/etc/nsyslog.conf` para sistemas Ultrix) la siguiente línea:

```
auth.info                /usr/local/adm/passwd.log
```

donde `/usr/adm/passwd.log` se puede sustituir por la ruta del archivo donde se desee llevar el registro de las actividades del `passwd+`.

12. Reinicializar el **syslogd** con el comando **kill -HUP <n>**, donde `<n>` sea el número de dicho proceso.
13. Probar que el `/bin/passwd` se ejecute correctamente, realizando los cambios y registrando la actividad de manera correcta.

## F.4 Apéndice: Instalación del nsyslog en sistemas Ultrix

El programa **syslog** permite el manejo en sistemas Unix de los mensajes generados por distintos procesos; estos mensajes son clasificados por categorías y prioridades, a los cuales se les da un nombre. En el archivo `/etc/syslog.conf` se especifica qué se debe hacer con los mensajes de cada combinación de categoría/prioridad que nos interese. Esta acción puede ser: almacenar el mensaje en un archivo, enviarlo por correo electrónico a alguna cuenta, o desplegarlo en algún dispositivo (como la consola). El daemon **syslogd** se tiene que encontrar corriendo para que se lleve a cabo el registro de los mensajes.

Por alguna razón no muy clara, el sistema de **syslog** de Ultrix (la variante de Unix de Digital Equipment Corporation) no se apega al estándar establecido, sino que proporciona una interfase diferente, y tiene una funcionalidad muy reducida, pues no permite la clasificación de los mensajes en 2 niveles, y el formato del archivo `syslog.conf` es totalmente diferente.

Sin embargo, muchos programas requieren el uso de **syslog** para el registro de sus mensajes. Por lo tanto, han surgido en el dominio público reemplazos para el **syslog** de Ultrix, que conforman con el estándar, de manera que ya no exista ningún problema a la hora de compilar aplicaciones que lo requieran. El **nsyslog** se puede obtener también en el FTP anónimo de `ftp.super.unam.mx`, en el archivo `pub/sys/dec/jtkohl-syslog-complete.tar.gz`. A continuación se detallan los pasos necesarios para su instalación.

1. Desempaquetar el archivo `jtkohl-syslog-complete.tar.gz` con el comando `zcat jtkohl-syslog-complete.tar.gz | tar xvf -`. Esto creará en el directorio actual un subdirectorio llamado **nsyslog** que contiene todos los archivos correspondientes a este programa.
2. Entrar al directorio **nsyslog** y ejecutar el comando **make** para iniciar la compilación.
3. Entrar a la cuenta de root para llevar a cabo la instalación del programa.
4. Copiar el ejecutable **syslogd** a `/etc/syslogd`.
5. Copiar el archivo `nsyslog.conf` a `/etc/nsyslog.conf`. Este es el archivo de configuración que le indica al `syslogd` a dónde debe ir cada tipo de mensajes.
6. Hacer una copia de seguridad del archivo `/usr/include/syslog.h`, y reemplazarlo con el archivo `syslog.h` proporcionado en el directorio del **nsyslog**.
7. Para implementar la funcionalidad del nuevo **syslog**, es necesario reemplazar el módulo binario del **syslog** en las librerías del sistema. Para ello se ejecutan los siguientes pasos:
  - (a) Entrar al directorio `/usr/lib`.
  - (b) Hacer copias de seguridad de los archivos `libc.a`, `libc_G0.a` y `libckrb.a`. Estas tres librerías son las que contienen el módulo binario `syslog.o`, que hay que reemplazar por la nueva versión.

- (c) Ejecutar el comando **ar r libc.a ruta\_nsyslog/syslog.o**, sustituyendo `ruta_nsyslog` por la ruta donde se llevó a cabo la compilación del **nsyslog**. Con esto se reemplaza el módulo `syslog.o` por el nuevo.
  - (d) Ejecutar el comando **ar ts libc.a**. Con esto se actualizan los encabezados de la librería, de manera que pueda ser utilizada sin problemas.
  - (e) Repetir los pasos 7c y 7d, reemplazando `libc.a` por `libc_G0.a` y `libc_krb.a`, para actualizar el módulo binario en esas librerías también.
8. La instalación está finalizada. Ahora se pueden compilar sin problema programas que utilicen llamadas al **syslog** estándar. Es necesario revisar el contenido del archivo `/etc/nsyslog.conf` para añadir cualquier cosa que pudiera ser necesaria en el sistema. Se puede hacer una comparación con el archivo `/etc/syslog.conf` ya existente, que está en formato del **syslog** de Ultrix, y añadir o quitar las cosas que sean pertinentes.

## Apéndice G

# Detalles técnicos de las listas de correo electrónico *gasu* y *cert-advisory*

Este apéndice detalla, de forma resumida, las características técnicas de las listas de correo electrónico *gasu* y *cert-advisory*, utilizadas en el proyecto de seguridad. Referirse al apéndice A para la definición de los términos utilizados.

### G.1 *gasu*

#### G.1.1 Características

- Pública.
- No moderada.
- Archivada en HTML: <http://www.super.unam.mx/listas/gasu/>.
- Dueño: `diego@ds5000.dgsca.unam.mx`.

#### G.1.2 Estadísticas

- Fecha de creación: 7 de Febrero de 1994.
- Mensajes distribuidos durante el primer año de operación: 353.
- Mensajes distribuidos hasta el 26 de Mayo de 1995: 475.
- Número de suscriptores al 26 de Mayo de 1995: 142.

## **G.2 *cert-advisory***

### **G.2.1 Características**

- Pública.
- Lista de solo lectura (los suscriptores no pueden enviar mensajes).
- Lista de redistribución de la lista `cert-advisory@cert.org`
- No archivada.
- Dueño: `diego@ds5000.dgsca.unam.mx`.

### **G.2.2 Estadísticas**

- Fecha de creación: 23 de Junio de 1994.
- Mensajes distribuidos hasta el 26 de Mayo de 1995: 19.
- Número de suscriptores al 26 de Mayo de 1995: 39.

## Apéndice H

# Programa de conferencias del DISC 1994

Este apéndice contiene el programa de conferencias del *Día Internacional de la Seguridad en Cómputo* 1994, realizado por primera vez en México, en las instalaciones de la DGSCA, el 5 de diciembre de 1995.

---

10:00-10:30

Auditorio de la DGSCA

**Inauguración y bienvenida**

Dr. Victor Guerra Ortiz

Director General de Servicios de Cómputo Académico

---

10:30-11:00

Auditorio de la DGSCA

**Conceptos básicos de seguridad**

Diego Martín Zamboni

Departamento de Supercómputo

En esta plática se tratarán los aspectos teóricos básicos de la seguridad en cómputo: Qué es la seguridad, tipos de seguridad existentes, por qué la seguridad es importante, etc.

---

11:00-11:30

Auditorio de la DGSCA

**Introducción a la seguridad en Unix para usuarios**

Lic. Martha Adriana Sánchez Cerezo

Jefe del Departamento de Supercómputo

En esta plática se darán los conocimientos necesarios, así como consejos prácticos, para que los usuarios de un sistema Unix puedan contribuir a la seguridad del mismo, mediante la protección de su información.

---

11:30-12:00

Auditorio de la DGSCA

**Introducción a la seguridad en Unix para administradores**

Diego Martín Zamboni

Departamento de Supercómputo

Conocimientos y consejos prácticos para que el administrador de un sistema Unix pueda asegurar en cierta medida la seguridad del mismo, mediante el establecimiento de mecanismos y políticas que permitan monitorear y controlar el acceso al sistema y a la información.

---

12:00-12:30

**Descanso**

---

12:30-13:00

Auditorio de la DGSCA

**Características del sistema de seguridad multinivel en UNICOS 8.0**

M. C. Edgardo Román

Cray Research, Inc.

Cray Research diseñó e implementó un sistema de seguridad multinivel sobre Unix estándar para cumplir con los requerimientos del Gobierno de los Estados Unidos. Actualmente es utilizado en muchas partes para proteger información privada y sensible. Cuáles son las partes operacionales que lo componen?

---

13:00-13:30

Auditorio de la DGSCA

**Conceptos básicos de diseño de redes seguras**

Ing. Rafael Lacambra Macedo

Jefe del Laboratorio de Visualización

La seguridad en una red puede clasificarse a nivel de máquina o a nivel de ruteador. Esta plática abordará el tema del diseño de redes teniendo en cuenta desde esta fase la seguridad lógica a nivel de ruteador (independientemente de la implementada a nivel de máquina) y los cambios necesarios en software para obtenerla. Asimismo se tocará el tema de las conexiones físicas necesarias.

---

13:30-14:00

Auditorio de la DGSCA

**Seguridad en los niveles 1 y 2 de TCP/IP**

Jorge Torres Antuñano, Luis E. Rojas Huerta

Departamento de Redes

La seguridad en las redes de cómputo es muy importante, sobre todo en Internet, a la cual están conectados miles de usuarios de todo el mundo. El protocolo que se utiliza en Internet es por excelencia TCP/IP, que es un conjunto de protocolos de comunicación dedicados a diferentes capas en el establecimiento de la misma.

Existen equipos analizadores de redes locales, como el *sniffer*, que decodifican diferentes protocolos de comunicación, entre ellos el protocolo de sesión remota conocido como **telnet**. Se verá cómo este equipo decodifica estos protocolos resultando inadecuado para la seguridad de la red. Se darán algunas soluciones para evitar este problema, que incluirán de manera básica los diferentes métodos criptográficos de servicios de seguridad.

---

14:00-16:00

**Descanso**

---

16:00-17:00

Auditorio de la DGSCA

**Por qué es importante la criptografía?**

Dr. Horacio Tapia Recillas

UAM Iztapalapa

Dentro del marco de Seguridad en Cómputo, el propósito de esta charla es presentar algunas ideas acerca de por qué consideramos que la Criptografía es un mecanismo que, junto con otras medidas, ayuda a tener un mayor grado de seguridad y confiabilidad en el manejo de cualquier tipo de información (transacciones comerciales, pagos electrónicos, confidencial, etc.), que se transmite por los diversos medios de comunicación. En general los medios por donde viaja la información son poco seguros (teléfono comercial, fax, satélite, redes de computadoras, teléfono celular, etc.), de tal manera que personas no autorizadas pueden tener acceso a dicha información y la pueden usar para diversos propósitos.

---

17:00-18:00

Auditorio de la DGSCA

**Estrategia de Seguridad de Hewlett-Packard "Secure Open Computing"**

Ing. Gustavo Adolfo Cordova Rayas

Hewlett-Packard de México, S. A. de C. V.

Presentación de la Estrategia de Seguridad de Hewlett-Packard la cual está basada en una arquitectura que abarca tanto hardware, sistema operativo como "MiddleWare", usada para guiar el desarrollo de servicios y sistemas de cómputo abiertos y distribuidos. El Servidor de Seguridad está basado en DCE (Distributed Computing Environment) de la OSF (Open Software

Foundation). Asimismo se hará una breve presentación del tipo de Servicios Profesionales que ofrece Hewlett-Packard en el área de Consultoría en Seguridad.

---

18:00-18:30

Auditorio de la DGSCA

**"Firewalls": Defensa contra ataques en Internet**

Lic. Aida Ramírez Bárcenas

Consortio Red Uno S.A. de C.V.

En esta plática se describe el concepto de "firewall", la filosofía bajo la que opera y configuraciones básicas. Se presenta información sobre el software "Firewall Toolkit" de Trusted Information Systems, Inc (TIS).

---

18:30-19:00

Auditorio de la DGSCA

**Historia de la criptografía**

Ma. Susana Soriano Ramírez

Departamento de Supercómputo

Los sistemas criptográficos son una de las pocas maneras de asegurar completamente la privacidad de la información. En esta plática se dará una visión histórica de este campo de estudio, recorriendo los diversos sistemas criptográficos que fueron utilizados en la antigüedad, hasta llegar a los de las épocas recientes.

## Apéndice I

# Cómo usar un servidor de FTP anónimo

Como primer paso para utilizar un servidor de FTP anónimo, es necesario contar con un cliente de FTP, que permita establecer la comunicación con el servidor remoto. Existen múltiples clientes para distintos sistemas operativos y ambientes de trabajo, pero los ejemplos en esta guía se limitarán al programa **ftp**(1), que forma parte estándar de todos los sistemas Unix.

1. Establecer la comunicación con el servidor. Por ejemplo:

```
% ftp ftp.super.unam.mx
```

2. Una vez establecida la comunicación, el servidor preguntará por un *login*. Contestar con **anonymous**.
3. Después se preguntará un *password*. En este punto es necesario proporcionar la dirección electrónica completa de la persona que está realizando el acceso. Por ejemplo: `diego@ds5000.dgsca.unam.mx`.  
Obsérvese que como siempre, el *password* tecleado no aparecerá en la pantalla.
4. Una vez concedido el acceso, es importante fijar el modo de transmisión de los archivos a “binario”, para que los archivos binarios (por ejemplo, programas) que sean transferidos no se alteren. Para esto, ejecutar el comando **bin**.
5. En este punto, se cuenta con los siguientes comandos principales:

**cd** directorio Para cambiarse al directorio especificado en el sistema remoto.

**lcd** directorio Para cambiarse al directorio especificado en el sistema local.

**dir** Para listar el directorio remoto.

**pwd** Para imprimir el directorio remoto actual.

**get** archivo Para transferir el archivo especificado del sistema remoto al local.

**put** archivo Para transferir el archivo especificado del sistema local al remoto.

**!comando** Para ejecutar el comando especificado en el sistema local. Si no se especifica un comando, se abre un *subshell* en el que se pueden ejecutar comandos interactivos en el sistema local.

**quit** Para cerrar la conexión con el sistema remoto.

La figura I.1 muestra una sesión de FTP al servidor del Departamento de Supercómputo para obtener el archivo `cops.tar.gz`

```
% ftp ftp.super.unam.mx
Trying 132.248.204.3...
Connected to mezcaldgsca.unam.mx.
220 mezcald FTP server (Version wu-2.4(1) Sun Feb 26 13:23:05 CST 1995) ready.
Name (ftp.super.unam.mx:diego): anonymous
331 Guest login ok, send your complete e-mail address as password.
Password: diego@ds5000.dgsca.unam.mx
230- Bienvenido al servidor de FTP anónimo del Departamento de Supercómputo!
230-
230- Para cualquier comentario o pregunta sobre este servidor, envíe
230- correo electrónico a diego@ds5000.dgsca.unam.mx.
230-
230 Guest login ok, access restrictions apply.
ftp> bin
200 Type set to I.
ftp> dir
200 PORT command successful.
150 Opening ASCII mode data connection for /bin/ls.
total 18
dr-xr-xr-x  3 root    daemon      512 Feb 26 20:12 bin
dr-xr-xr-x  2 root    daemon      512 Feb 26 19:56 dev
dr-xr-xr-x  4 root    daemon      512 Apr  7 22:51 etc
drwxrwxrwx  2 ftp     ftp         512 Apr 11 14:44 incoming
drwxr-xr-x  2 root    daemon      512 Feb 27 22:53 info
dr-xr-xr-x  2 root    wheel      8192 Feb  6 23:44 lost+found
dr-xr-xr-x 24 ftp     daemon      512 Apr 13 19:13 pub
dr-xr-xr-x  2 ftp     ftp         512 Dec 20 21:39 resúmenes
dr-xr-xr-x  3 root    daemon      512 Feb 26 19:55 usr
226 Transfer complete.
687 bytes received in 0.15 seconds (4.5 Kbytes/s)
ftp> cd pub/security/tools
250 CWD command successful.
ftp> lcd /tmp
Local directory now /tmp
ftp> get cops.tar.gz
200 PORT command successful.
150 Opening BINARY mode data connection for cops.tar.gz (289529 bytes).
226 Transfer complete.
local: cops.tar.gz remote: cops.tar.gz
289529 bytes received in 0.68 seconds (4.2e+02 Kbytes/s)
ftp> quit
221 Goodbye.
%
```

Figura I.1: Ejemplo de sesión en FTP



# Apéndice J

## Manual del usuario de New CARP

### J.1 Introducción

New CARP (*New COPS Analysis Report Program*) es un programa desarrollado en el Departamento de Supercómputo de la DGSCA con el fin de cubrir una deficiencia muy importante en las herramientas de seguridad actuales: la falta de presentación de la información obtenida en un formato que haga sencilla para los administradores su análisis y entendimiento. La principal razón por la que comunmente los administradores no toman medidas para corregir los problemas reportados por las herramientas de seguridad es que en muchas ocasiones es difícil entender dichos reportes.

COPS es una de las principales herramientas en el monitoreo de la seguridad interna de un sistema Unix. Sin embargo, normalmente los reportes generados dejan mucho que desear en cuanto a claridad de los problemas encontrados, y no proporcionan ninguna información acerca de cómo solucionar dichos problemas.

CARP es un programa incluido en la distribución de COPS, que intenta cubrir un poco este problema mediante la presentación de los problemas encontrados en un formato tabular, que permite descubrir fácilmente los problemas graves que hayan sido encontrados en una serie de máquinas revisadas por COPS.

New CARP (o simplemente NCARP) es una herramienta diseñada para cubrir los puntos faltantes: proporcionar información detallada sobre los problemas encontrados, así como sugerir soluciones a los mismos.

En su primera versión, los reportes se envían por correo electrónico normal. Sin embargo, en versiones posteriores se utilizarán mecanismos de cifrado de la información para evitar que dichos resultados puedan ser utilizados por personas que no sean el receptor autorizado.

## J.2 Requerimientos para usar NCARP

NCARP analiza la información generada por COPS, de manera que es necesario tener ya instalado y corriendo de forma regular dicho programa. NCARP alcanza su máxima utilidad cuando se tienen los reportes generados por COPS en múltiples sistemas concentrados en una sola máquina (en la cual se ejecutará NCARP). Dichos reportes tienen que estar almacenados debajo del directorio en el que está instalado COPS, en subdirectorios con el nombre de cada uno de los sistemas. Si se está utilizando COPS en su configuración estándar, así es como se estarán almacenando los reportes.

## J.3 Cómo obtener NCARP

El primer paso en la utilización de NCARP es obtenerlo. NCARP no tiene restricciones en cuanto a su utilización, pues está disponible en Internet para cualquiera que quiera obtenerlo.

La última versión de NCARP está disponible en `ftp://ftp.super.unam.mx/pub/security/tools/ncarp.tar.gz`.

## J.4 Cómo instalar NCARP

NCARP es un programa sumamente sencillo de instalar. A continuación se detallan los pasos a seguir:

1. Ejecutar el siguiente comando:

```
% zcat ncarp.tar.gz | tar xvf -
```

Esto creará un directorio llamado `ncarp-x.y`, donde `x.y` es el número de versión del programa que se haya obtenido.

2. Cambiarse al directorio que acaba de ser creado:

```
% cd ncarp-x.y
```

3. Ejecutar el comando **make**. Esto modificará el programa `ncarp` para referirse a las rutas correctas en las que están almacenadas los distintos programas necesarios por NCARP para su ejecución. Si no se cuenta con el comando **make**, se puede ejecutar **./reconfig**, aunque esto no es recomendado.
4. Si no hubo ningún error, NCARP está listo para ser utilizado.

```

hostname      rep date      crn dev ftp grp hme is pass msc pwd rc  rot usr
=====
ds5000        1992_Jan_27  | 1 |   | 2 |   | 1 | 2 |   |   | 2 | 2 | 2 |   |
polaris       1992_Jan_26  |   |   | 2 | 2 | 1 | 2 |   |   | 2 | 2 |   |   | 1 |
capella       1992_Jan_15  |   |   |   | 2 | 1 | 2 |   |   |   |   |   | 0 |   |

#####
ds5000
#####
<descripción de los problemas encontrados por cron.chk>
<descripción de los problemas encontrados por ftp.chk>
<descripción de los problemas encontrados por home.chk>
<...>

#####
polaris
#####
<descripción de los problemas encontrados por cron.chk>
<descripción de los problemas encontrados por ftp.chk>
<descripción de los problemas encontrados por home.chk>
<...>

...

```

Figura J.1: Formato de reporte producido por NCARP

## J.5 Utilización de NCARP

### J.5.1 Utilización básica

La forma mas sencilla de utilizar NCARP es ejecutarlo, indicándole como argumento el directorio debajo del cual están almacenados los reportes que se quieren utilizar (normalmente será el directorio en el que está instalado el COPS. Por ejemplo:

```

% cd /usr/local/sec/ncarp
% ./ncarp /usr/local/sec/cops

```

La ejecución de NCARP generará:

1. En la salida estándar, el reporte completo, conteniendo la información de todos los sistemas cuyos reportes fueron procesados, en el formato que se muestra en la figura J.1. La tabla inicial contiene una línea por cada máquina que fue analizada. Las columnas indican la fecha del reporte de COPS que se analizó, así como los problemas encontrados y la gravedad de los mismos:
  - 0 Un problema que puede dar acceso inmediato a la clave de **root** en el sistema.
  - 1 Un problema relativamente serio, que podría dar acceso a la clave de algún usuario.

```

maquina          email@maquina[,email2@maquina2,...]
maquina2        email3@maquina3[...]
```

Figura J.2: Formato del archivo de configuración de NCARP

- 2 Un problema reportado por COPS, pero cuya gravedad no se puede determinar con precisión.

La segunda sección del reporte contiene, para cada máquina, una descripción detallada de los problemas encontrados, así como posibles formas de solucionarlos.

2. Los reportes individuales para cada una de las máquinas analizadas se almacenarán en archivos llamados `reports/maquina/carp.aaaa_mmm_dd`, donde `maquina` es el nombre del sistema analizado, y `aaaa_mmm_dd` representa la fecha en la que fue generado el reporte. Cada reporte individual contiene la línea correspondiente de la tabla, así como la sección correspondiente de la descripción de los problemas.

El reporte individual para una máquina solamente es generado si difiere del último reporte que se encuentre almacenado para dicho sistema.

### J.5.2 Utilización avanzada

NCARP tiene la posibilidad de enviar por correo electrónico los reportes individuales que genera. Esto permite, por ejemplo, que cada reporte sea enviado de forma automática al administrador del sistema, para que tome las acciones correspondientes.

Para esto, NCARP cuenta con un archivo de configuración llamado `ncarp.mail`. Este archivo debe tener el formato que se muestra en la figura J.2.

Cada renglón debe contener, en la primera columna, el nombre de la máquina a la que se hace referencia, y en la segunda columna, la dirección o direcciones electrónicas a las que deben ser enviados los reportes de dicha máquina, separadas por espacios y/o comas. Dentro de este archivo, las líneas que comiencen con “#” son consideradas como comentarios.

Si existe este archivo, los reportes generados serán enviados por correo electrónico a las direcciones especificadas para cada máquina, además de ser almacenados en el disco. Las máquinas para las que no exista una línea en el archivo de configuración no generarán ningún mensaje de correo electrónico, y su reporte solamente será almacenado.

## Apéndice K

# Manual del usuario de SAINT

### K.1 Introducción

SAINT (*Security Analysis Integrator Tool*) es un programa desarrollado en el Departamento de Supercómputo de la DGSCA con el fin de proporcionar a los administradores un mecanismo eficiente y sencillo de utilizar que permita el análisis conjunto de los datos proporcionados por las diversas herramientas de seguridad utilizadas, tales como COPS, TCP-Wrapper, passwd+, Crack y TripWire. También se realiza un análisis sobre algunos archivos de bitácora del sistema que pueden proporcionar información importante sobre eventos de seguridad.

SAINT analiza la información generada por dichas herramientas en varios sistemas, y produce reportes que pueden ser consultados de acuerdo a los siguientes criterios:

- Gravedad de problemas encontrados.
- Tipo de problemas.
- Máquina analizada.
- Herramienta que generó los datos.
- Lapso de tiempo en que ocurrieron los eventos.

En su forma inicial, SAINT genera sus reportes en formato de texto, pero posteriormente se integrará en el programa la posibilidad de visualizar los resultados utilizando mecanismos de hipertexto, lo que permitirá una mayor flexibilidad en el estudio de los mismos.

### K.2 Requerimientos para usar SAINT

SAINT analiza los datos generados por otras herramientas de monitoreo y control de seguridad, de manera que dichas herramientas ya deben estar en funcionamiento antes de instalar SAINT. Si los resultados de varias máquinas se concentran en una sola, SAINT

permite analizar todos esos datos de manera conjunta, abriendo la posibilidad de investigar eventos que “viajen” a través de la red, como por ejemplo, el establecimiento de sesiones interactivas de un sistema a otro.

Consultar la documentación de las distintas herramientas de seguridad, así como del sistema operativo, para detalles sobre la concentración de mensajes en una sola máquina.

SAINT está escrito en el lenguaje de programación **perl**, versión 5, por lo que este programa debe estar instalado en el sistema en el que vaya a ser utilizado SAINT. Si no se encuentra ya instalado, **perl5** puede ser obtenido por FTP anónimo en `ftp://ftp.super.unam.mx/pub/gnu/perl5*`.

### K.3 Cómo obtener SAINT

El primer paso en la utilización de SAINT es obtener la última versión, disponible en `ftp://ftp.super.unam.mx/pub/security/tools/saint.tar.gz`.

SAINT se encuentra liberado en el dominio público y su utilización es libre para quien lo desee.

### K.4 Cómo instalar SAINT

SAINT es un programa sumamente sencillo de instalar. A continuación se detallan los pasos a seguir:

1. Ejecutar el siguiente comando:

```
% zcat saint.tar.gz | tar xvf -
```

Esto creará un directorio llamado `saint-x.y`, donde `x.y` es el número de versión del programa que se haya obtenido.

2. Cambiarse al directorio que acaba de ser creado:

```
% cd saint-x.y
```

3. Ejecutar el comando **make**. Esto modificará el programa **ncarp** para referirse a las rutas correctas en las que están almacenadas los distintos programas necesarios por SAINT para su ejecución. También se preguntarán las localizaciones de los archivos en los que se encuentra la información que será procesada por SAINT.

Si no se cuenta con el comando **make**, se puede ejecutar `./reconfig`, aunque esto no es recomendado.

4. Si no hubo ningún error, NCARP está listo para ser utilizado.

## K.5 Utilización de SAINT

### K.5.1 Utilización básica

La forma más sencilla de utilizar NCARP es ejecutarlo sin argumentos. Así, el programa generará en la salida estándar un reporte en el que se concentran, cronológicamente, todos los eventos relevantes de seguridad detectados en los archivos analizados. En la figura K.1 se observa un ejemplo de reporte generado de esta manera. Las columnas son:

**SISTEMA** La máquina en la que fue registrado el evento.

**FECHA** Fecha y hora en que sucedió el evento.

**DESCRIPCION** Una descripción del evento, y si es aplicable, su posible impacto en la seguridad del sistema.

**CAT.** Categoría del evento. SAINT reconoce varias categorías de eventos:

**Informativos (columna en blanco)** Eventos relevantes, pero sin ninguna trascendencia en especial.

**De advertencia (1)** Eventos que se salen de lo común, aunque no indican ningún problema de forma directa.

**De alerta (2)** Posibles problemas de seguridad, que deben ser investigados inmediatamente.

**De emergencia (3)** Un claro problema de seguridad.

Como se puede observar, SAINT hace un análisis entre los eventos para detectar posibles secuencias que indiquen problemas de seguridad o comportamiento poco usual

### K.5.2 Utilización avanzada

El poder real de SAINT viene de sus múltiples opciones de funcionamiento, así como de su archivo de configuración. Con estos elementos es posible especificar el tipo de reporte a generar, así como configurar SAINT para las condiciones de operación específicas de los sistemas que se están analizando.

Las opciones de línea de comandos disponibles son:

**-f módulo[(argumento,...)],...** Especifica los módulos de presentación de resultados que serán utilizados. Puede especificarse más de un módulo, separándolos por comas. En este caso, dichos módulos serán aplicados en el orden que aparezcan.

Como los módulos de presentación de resultados son programas independientes, algunos de ellos pueden recibir argumentos que modifiquen su comportamiento (un ejemplo claro es el módulo de selección por intervalos de tiempo, que necesita saber el intervalo deseado). En estos casos, el nombre del módulo puede ir seguido por los argumentos que se le quieran pasar, entre paréntesis y separados por comas. Estos argumentos serán pasados al módulo separados por espacios y

```

Reporte de SAINT correspondiente al Sat May 13 21:10:43 CST 1995
Generado en ds5000.dgsca.unam.mx
Periodo cubierto en este reporte:
    Mon May  8 00:00:01 CST 1995 - Sat May 13 21:05:12 CST 1995

Sistemas analizados:
    ds5000.dgsca.unam.mx
    polaris.labvis.unam.mx
    capella.labvis.unam.mx

-----
SISTEMA      FECHA      DESCRIPCION      CAT.
-----
ds5000      May  8 01:15  telnet desde capella.labvis.unam.mx      1
             Este acceso se realizó en un horario de
             trabajo poco común. Valdría la pena
             revisar más profundamente.

polaris     May  8 01:17  ftp desde ds5000.dgsca.unam.mx          1
             Este acceso se realizó en un horario de
             trabajo poro común. Podría provenir de la
             sesión mencionada anteriormente en
             ds5000.

< ... >

ds5000      May 10 14:25  telnet desde away.crime.lab
             Este acceso se realizó desde una máquina
             externa a la red local. Conviene averiguar
             si fue realizada por un usuario autorizado.

ds5000      May 10 14:30  su a root realizado por lvm              2
             Se entró a la cuenta de root mientras
             estaba activa una sesión desde una red
             externa. Fue realizado por un usuario
             autorizado?

< ... >

capella     May 13 20:14  reboot.
             Un reboot es normalmente un evento poco común
             en un sistema Unix. Fué válido y autorizado?

-----

Total de eventos reportados en el período: 25.

Total de eventos informativos (CAT= ):          20.
Total de eventos de advertencia (CAT=1):        3.
Total de eventos de alerta (CAT=2):            2.

-----

```

Figura K.1: Ejemplo de un reporte de SAINT

sin los paréntesis. Por ejemplo, si se especifica **-f timeinterval(9501,9512)**, el módulo será ejecutado con **timeinterval 9501 9512**.

Los módulos que se incluyen en la distribución estándar de SAINT son:

**full** Genera un reporte completo, incluyendo todas las máquinas analizadas, en todo el período de tiempo disponible en los archivos analizados.

**crono** Ordena los eventos cronológicamente.

**machine(maquina,...)** Restringe el reporte a los eventos generados en las máquinas especificadas.

**domain(dominio,...)** Restringe el reporte a los eventos generados en los dominios especificados.

**timeinterval(inicio,fin)** Permite restringir el reporte a los eventos ocurridos en el período especificado. Tanto *inicio* como *fin* tienen el formato *YY[MM[dd[hh[mm]]]]*, que permite indicar la fecha y la hora deseada. Las partes que aparecen entre corchetes ([...]) son opcionales.

**bybadness** Ordena los eventos de acuerdo a su gravedad, comenzando por los más peligrosos.

**bysystem** Ordena los eventos de acuerdo a la máquina en la que fueron registrados, en orden alfabético.

**bytype** Ordena los eventos de acuerdo a su tipo.

**-c archivo** Permite especificar el archivo de configuración que será utilizado. Si no se especifica, el default es `saint.cf`, ubicado en el mismo directorio en el cual reside el programa.

**-v** Imprime información a la terminal acerca de lo que va haciendo. Normalmente SAINT es bastante silencioso mientras se ejecuta, a excepción de los mensajes de error.

Si no se especifican los módulos de presentación de resultados en la línea de comandos ni en el archivo de configuración, el default es **crono,full**.

En el archivo de configuración se pueden especificar las características del ambiente de trabajo en el que se encuentran los sistemas. Por ejemplo:

- Horarios normales de trabajo.
- Usuarios autorizados a utilizar la clave de **root**.
- Direcciones electrónicas desde las cuales es normal acceder a los sistemas.
- Horarios de mantenimiento de los sistemas, en los que es normal llevar a cabo tareas poco comunes, como reinicializarlos.
- Máquinas desde las cuales es común que un usuario autorizado lleve a cabo tareas administrativas en la cuenta de **root**.

Consultar la página de manual `saint.cf(5)` para mayores detalles sobre el archivo de configuración de SAINT.



## Apéndice L

# Estado de la seguridad en la supercomputadora de la UNAM

Este apéndice contiene el documento que fue presentado al grupo de seguridad en cómputo del Laboratorio Nacional de Los Alamos de los Estados Unidos, durante una visita de personal de la DGSCA a dicha institución.

---

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO  
DIRECCION GENERAL DE SERVICIOS DE COMPUTO ACADEMICO  
DEPARTAMENTO DE ADMINISTRACION DE SUPERCOMPUTO  
**UNIX SECURITY**

### INDEX

ACTUAL STATUS OF SECURITY-RELATED ITEMS  
WHAT HAS BEEN DONE IN SECURITY-RELATED ISSUES  
REFERENCES AND RESOURCES  
QUESTIONS

## L.1 Actual status of security-related items

1. The supercomputer does not have a firewall machine.
2. We are not using an IDcard system.
3. We are not using MLS (Multi-Level Security).
4. fingerd has been deactivated on the supercomputer.
5. Access to the NSC routers from sites outside Mexico is neglected.
6. Non-mexican people are not allowed use of the supercomputer, except for special cases.

This is, the supercomputer is running as an ordinary Unix system, with direct connection to the Internet, and so easily accesible from anywhere. We are not using any supercomputer-specific security measures.

## L.2 What has been done in security-related issues

1. COPS (Computer Oracle and Password System) version 1.04 has been installed on the supercomputer and the workstations directly related to it (Visualization Lab and Supercomputing Administration Department). COPS runs daily on all the machines, and the reports are concentrated in a single machine for further analysis and, based on that, making recommendations to the corresponding administrators. We intend to build tools that, based on the reports generated by COPS, generate reports in spanish so that the feedback to the administrators can be given in a more understandable form and in an automated fashion. Also in the plans is an X-based frontend for COPS, in spanish, for use of the university community in general. This tool could also include things not present in COPS, either from other public tools or things developed in-house.
2. The password cracker program Crack version 4.1 has been installed experimentally on a workstation, and the plans include running it periodically in the supercomputer (for taking advantage of the speed of the supercomputer, since password cracking is a very time-consuming job) over the password files of all the workstations and the Cray itself. This could result in reports that are either mailed directly to the problem users or to a person in charge that would decide what to do.

Since the first lines of defense in a Unix system are the accounts' passwords and file systems access permissions, special attention has been put initially in these aspects. COPS has been very useful in finding wrong file access permissions, ftp configuration flaws, and many other often overlooked details. Both COPS and Crack have been used to find weak passwords, accounts without passwords, and things like that. We have found A LOT of accounts with password-related problems, since people very often choose very poor passwords or even leave their accounts without passwords.

The use of COPS and Crack has been complementary to the almost continuous human inspection of the systems, their log files, etc.

All the documents referenced, but particularly Garfinkel's Practical Unix Security [1] have been used to build a security checklist that we try to accomplish on all the machines under our direct control.

All the security measures mentioned, specially the installation of new software, have been initially tested on a single workstation, and then expanded to other workstations and the supercomputer.

Also high on our list of priorities has been the education of both users and administrators. Making users security-conscious is one of the most important tasks in keeping a system secure, but we have also found it one of the most difficult to accomplish. One severe problem is that many of the people who use Unix systems do not know Unix, they simply use some applications, and consequently don't care for details like file access

permissions and passwords. Our network is partly composed by PCs, and that makes security an even more difficult task.

### L.3 References and resources

- Books and documents
  1. Simson Garfinkel and Gene Spafford PRACTICAL UNIX SECURITY. O'Reilly & Associates, Inc.
  2. David A. Curry IMPROVING THE SECURITY OF YOUR UNIX SYSTEM. SRI International, April 1990.
  3. Russell L. Brand COPING WITH THE THREAT OF COMPUTER SECURITY INCIDENTS: A PRIMER FROM PREVENTION THROUGH RECOVERY. June 1990.
  4. U.S.A. Department of Defense. TRUSTED COMPUTER SYSTEM EVALUATION CRITERIA. December 1985.
  5. Andres Cherry, Mark Henderson, et al. PASS OR FAIL: A NEW TEST FOR PASSWORD LEGITIMACY. Mathematics and Computer Science Division, Argonne National Laboratory. Oct. 1992.
  6. Alec D. Muffett COMPUTER SECURITY FAQ from Usenet newsgroup **comp.security.misc**.
- Software resources
  1. COPS 1.04 - Dan Farmer et al.
  2. Crack 4.1 - Alec D. Muffett.
- Other resources
  1. Usenet newsgroups **comp.security** and **comp.security.misc**.
  2. Documents and security information from CERT (Computer Emergency Response Team).

### L.4 Questions

1. How beneficial is the use of NIS/YP and/or Kerberos in a local network like ours, composed of a supercomputer and some workstations used for visualization?
2. What kind of automated tools do you use and/or recommend for monitoring systems' security? Do you use COPS and Crack or any other similar tools for monitoring security-related details?
3. What kind of security related courses, seminars and workshops do you give users and administrators for making them security-conscious? Do you think it's best to leave in the users' hands some responsibility, or to take totalitarian measures to get them from making mistakes and omissions that could compromise security?

4. What do you think is the best interval for forcing users to change their password? Currently we are using a 15 day period, but some sources say anything less than two months is more damaging than beneficial, since it makes users start repeating passwords, or changing it to something "temporal" and then returning immediately to the old password.
5. Is it convenient to send a user whose password has been guessed a message saying "Hey, your password has been guessed, please change it", or it's best to talk directly to him trying to explain the reasons why he/she should change the password?
6. We want to start several works on security issues, including some thesis works for students. Could you suggest some topics that are not yet very exploited, so working on them can result in something useful for the entire community?

Thank you very much for your attention and your help. Any comments, suggestions or questions on this text will be appreciated. Please send them to Diego Zamboni (diego@ds5000.dgsca.unam.mx).

## Apéndice M

# Anuncio del servicio de revisión remota utilizando SATAN

Este apéndice contiene el mensaje que fue enviado por el autor de esta tesis a la lista de correo electrónico *gasu*, el 5 de abril de 1995, anunciando el servicio de revisión remota de sistemas Unix utilizando el programa SATAN.

---

From diego Wed Apr 5 22:44:12 1995  
To: gasu  
Subject: OJO: Revisión remota de seguridad.  
Date: Wed, 5 Apr 1995 22:44:12 -0600 (CST)

```
#####  
# DGSCA ABRE EL SERVICIO DE REVISION REMOTA DE SEGURIDAD EN SISTEMAS UNIX #  
#####
```

```
----- FAVOR DE DISTRIBUIR LO MAS AMPLIAMENTE POSIBLE ESTE MENSAJE -----
```

5 de Abril de 1995.

Con fecha de hoy fue liberado a la utilización pública la herramienta SATAN (Security Analysis Tool for Auditing Networks), que permite el análisis remoto de la seguridad de sistemas Unix [1].

Este programa puede tener un gran impacto en la seguridad de los sistemas de cómputo de la Universidad, dado que al haber sido liberado en Internet es fácilmente utilizable por cualquier persona, sea un administrador del sistema o alguien que intente obtener a través de él acceso no autorizado a otros sistemas de cómputo.

Ante esto, el Departamento de Supercómputo de la DGSCA a través de la Coordinación del Grupo de Administración y Seguridad en Unix (GASU), y conjuntamente con las autoridades de dicha dependencia, inaugura el servicio de análisis remoto de seguridad en sistemas Unix de la UNAM.

Este servicio se inaugura ofreciendo la realización de análisis remotos utilizando SATAN, aunque posteriormente se extenderá a otros servicios de análisis y monitoreo de seguridad que serán anunciados en su momento.

A continuación se describe este servicio inicial.

Atentamente.

--

Diego Martin Zamboni                    Depto. de Administracion de Supercomputo  
diego@ds5000.dgsca.unam.mx    DGSCA, UNAM, Mexico. Tel. (5)622-85-29  
WWW home page: <http://www.super.unam.mx/~diego/>

[1] La documentación completa de SATAN puede encontrarse en  
[ftp://ftp.super.unam.mx/pub/security/doc/satan\\_doc.tar.gz](ftp://ftp.super.unam.mx/pub/security/doc/satan_doc.tar.gz).

-----

ANALISIS REMOTO DE SEGURIDAD UTILIZANDO SATAN  
=====

Objetivos:

Utilizar la herramienta más nueva de análisis de seguridad (y una de las más poderosas a la fecha) para analizar remotamente la seguridad de sistemas Unix de la forma más rápida y oportuna posible. La pronta corrección de los problemas encontrados por SATAN impedirá la utilización de los mismos para obtener acceso no autorizado a los recursos de las máquinas.

DGSCA cuenta con los recursos de hardware y software, así como con el personal para realizar estas revisiones, ahorrando a los administradores de sistemas Unix de la Universidad la instalación de nuevos programas para poder hacerlo de forma local.

Condiciones:

- El análisis remoto de la seguridad solamente se hará en sistemas en los que se cuente con autorización por escrito de los responsables de los mismos.
- Los reportes sobre los resultados obtenidos se harán llegar también por escrito a los responsables de los sistemas.
- Toda la información obtenida será completamente confidencial, dándose a conocer únicamente a los interesados.
- DGSCA se reserva el derecho de conservar la información obtenida, únicamente con fines estadísticos y de análisis. En ningún momento se divulgará dicha información a persona o institución alguna que no sea la directamente interesada.

Pasos a seguir:

1. Enviar una solicitud por escrito, firmada por el responsable del área a la que pertenecen los sistemas a analizar. La solicitud debe estar dirigida a:

Diego Zamboni  
Departamento de Supercómputo  
Coordinación de GASU.

La solicitud debe contener:

- a) Nombre del área y la dependencia a la que pertenecen los sistemas a analizar.
- b) Direcciones electrónicas (nombres y números) de las máquinas cuyo análisis se solicita. Si se desea el análisis de un dominio completo, indicarlo así, incluyendo únicamente la dirección completa de una máquina dentro de dicho dominio.
- c) Si se desea que se realicen revisiones periódicas de los sistemas indicados.
- d) Datos (nombre, teléfono y dirección electrónica) de una persona a la que se pueda contactar para verificaciones y aclaraciones.

La solicitud puede enviarse por fax al 622-81-49 (solamente 2-81-49 si se envía desde dentro de la UNAM), o entregarse personalmente en el Depto. de Supercómputo en la DGSCA.

2. La solicitud se confirmará telefónicamente con la persona que haya sido especificada como contacto.
3. Una vez realizado el análisis solicitado, los resultados impresos se harán llegar a la persona que firmó la solicitud. Si se solicitan revisiones periódicas, los resultados de cada una se harán llegar oportunamente a la persona responsable.

-----

Para cualquier aclaración, comunicarse con Diego Zamboni al 622-85-29 o por correo electrónico a [diego@ds5000.dgsca.unam.mx](mailto:diego@ds5000.dgsca.unam.mx).



# Bibliografía

- [Ale] Norm Aleks. The mailing list management software faq. Disponible en `ftp://ftp.uu.net/usenet/news.answers/mail/list-admin/software-faq.Z`.
- [AMP74] C. R. Attanasio, P. W. Markstein, and R. J. Phillips. Penetrating an operating system: A study of vm/370 integrity. *IBM Systems Journal*, 15(1), 1974.
- [Bel92] Steven M. Bellovin. There be dragons. In *Proceedings of the Thirs Usenix UNIX Security Symposium*, Murray Hill, NJ, August 15 1992. AT&T Bell Laboratories.
- [BL73] D. E. Bell and L. J. LaPadula. Secure computer systems: Mathematical foundations and model. Doc. No. M74-244, Mitre Corporation, Bedford (MA), 1973. Disponible en NTIS: AD 771543.
- [BPC78] R. Bisbey, G. Popek, and J. Carlstedt. Protection errors in operating systems. Technical report, USC Information Sciences Institute, 1978.
- [Bra90] Russell L. Brand. *Coping with the Threat of Computer Security Incidents. A Primer from Prevention through Recovery*, June 8 1990.
- [CER90] CERT. CERT. *Bridge*, March 1990. Disponible en `ftp://cert.org/pub/cert/_advisories/cert-article`.
- [Che91] Bill Cheswick. The design of a secure internet gateway. Technical report, AT&T Bell Laboratories, Murray Hill, New Jersey 07974, September 10 1991. Disponible en `ftp://ftp.super.unam.mx/pub/security/doc/gateway.dvi.gz` y `ftp://ftp.super.unam.mx/pub/security/doc/gateway.ps.gz`.
- [CHN<sup>+</sup>92] Andrew Cherry, Mark W. Henderson, William K. Nickless, Robert Olson, and Gene Rackow. Pass or fail: A new test for password legitimacy. Technical report, Applied Mathematical Sciences subprogram for the Office of Energy Research, U. S. Department of Energy, September 25 1992.
- [CMQ92] Smoot Carl-Mitchell and John S. Quarterman. Building internet firewalls. *UnixWorld*, pages 93–102, February 1992.

- [CN91] Brad J. Cox and Andrew J. Novobilski. *Object Oriented Programming: An Evolutionary Approach*. Addison-Wesley Publishing Company, Reading, Massachusetts, 1991.
- [Cona] Daniel W. Connolly. Hypertext markup language (HTML). Página de WWW disponible en <http://info.cern.ch/hypertext/WWW/MarkUp/MarkUp.html>. Publicado y mantenido por *The World Wide Web consortium*.
- [conb] The World Wide Web consortium. Hypertext transfer protocol. Página de WWW disponible en <http://www.w3.org/hypertext/WWW/Protocols/Overview.html>.
- [Cor] NetScape Communications Corporation. Welcome to netscape. Documento electrónico disponible en WWW: <http://www.netscape.com/>.
- [Cur90] David A. Curry. Improving the security of your unix system. Reporte ITSTD-721-FR-90-21, SRI International, 333 Ravenswood Avenue, Menlo Park, CA 94025-3493, April 1990.
- [DoD78] Department of Defense. *Security Requirements for Automatic Data Processing (ADP) Systems (DoD 5200.28)*, 1972, revised 1978.
- [DoD79] Department of Defense. *Security Manual—Techniques & Procedures for Implementing, Deactivating, Testing, and Evaluating Secure Resource-Sharing ADP Systems (DoD 5200.28-M)*, 1973, revised 1979.
- [DoD85] Department of Defense. *Department of Defense Trusted Computer System Evaluation Criteria (DoD 5200.28-STD)*, Dec 1985. Disponible en GPO: SN 008-000-00461-7.
- [dRds] Departamento de Redes de DGSCA. REDUNAM. Disponible en el GOPHER del departamento de redes: <gopher://noc.noc.unam.mx/>.
- [Fau84] Lincoln D. Faurer. Computer security goals of the department of defense. *Computer Security Journal*, Summer 1984.
- [FS93] David Ferbrache and Gavin Shearer. *Unix Installation Security & Integrity*. PTR Prentice Hall, Englewood Cliffs, New Jersey 07632, 1993.
- [fSAa] National Center for Supercomputing Applications. A beginner's guide to urls. Página de WWW disponible en <http://www.ncsa.uiuc.edu/demoweb/url-primer.html>.
- [fSAb] National Center for Supercomputing Applications. Ncsa mosaic. Documento electrónico disponible en WWW: <http://www.ncsa.uiuc.edu/SDG/Software/XMosaic/>.
- [Gar95] Simson Garfinkel. *PGP—Pretty Good Privacy*. O'Reilly & Associates, Inc., 103 Morris Street, Suite A, Sebastopol, CA 95472, January 1995.

- [GR89] Adele Goldberg and David Robson. *Smalltalk 80—The Language*. Addison-Wesley, Reading, Massachusetts, 1989.
- [GS92] Simson Garfinkel and Gene Spafford. *Practical Unix Security*. O'Reilly & Associates, Inc., 103 Morris Street, Suite A, Sebastopol, CA 95472, October 1992.
- [Hun94] Craig Hunt. *TCP/IP Network Administration*. O'Reilly & Associates, Inc., 103 Morris Street, Suite A, Sebastopol, CA 95472, first edition, January 1994.
- [Kle92] Daniel V. Klein. "foiling the cracker": A survey of, and improvements to, password security. Technical report, Software Engineering Institute, Carnegie Mellon University, 1992.
- [Kot94] Anastasios Kotsikonas. Listprocessor 6.0c mail list server. Disponible en `ftp://ftp.super.unam.mx/pub/mail/listproc6.0c.940712.0.sh.gz`, 1994.
- [KS74] P. A. Karger and R. R. Schell. Multics security evaluation: Vulnerability analysis. Disponible en NTIS: AD A001120, Electronic Systems Division, U. S. Air Force, Hanscom Air Force Base, Bedford (MA), 1974.
- [KS94] Gene H. Kim and Eugene H. Spafford. Experiences with tripwire: Using integrity checkers for intrusion detection. Purdue Technical Report CSD-TR-94-012, COAST Laboratory, Department of Computer Sciences, Purdue University, West Lafayette, IN 47907-1398, February 21 1994.
- [M<sup>+</sup>93] Alec Muffet et al. Faq: Computer security frequently asked questions. Disponible en los grupos de Usenet comp.security.misc y alt.security, Dec 1993. Versión 2.2.
- [MT79] Robert Morris and Ken Thompson. Password security: A case history. *Communications of the ACM*, 22(11):594–597, November 1979.
- [Nib79] G. H. Nibaldi. Proposed technical evaluation criteria for trusted computer systems. Reporte M79-225, Mitre Corporation, Bedford (MA), 1979. Disponible en NTIS: AD A108832.
- [NS78] R. M. Needham and M. D. Schroeder. Using encryption for authentication in large networks of computers. *Communications of the ACM*, 21(12):993–999, December 1978.
- [O'C94] Bryan D. O'Connor. Washington university ftp server. Disponible en `ftp://ftp.super.unam.mx/pub/net/wu-ftpd-2.4.tar.gz`, 1994.
- [oK] University of Kansas. About lynx. Documento electrónico disponible en WWW: `http://kuhttp.cc.ukans.edu/about\_lynx/about\_lynx.html`.

- [oST89] National Institute of Standards and Technology. Selected bibliography of key computer security literature. Disponible en `ftp.super.unam.mx:/pub/security/lit/800-1complete.txt.gz`, 1980–1989.
- [PR85] J. Postel and J. Reynolds. File transfer protocol (ftp). Request for Comments 959, Network Working Group, ISI, October 1985. Disponible en `ftp://nic.ddn.mil/rfc/rfc959.txt`.
- [PW91] Lewis J. Pinson and Richard S. Wiener. *Objective-C: Object Oriented Programming Techniques*. Addison-Wesley Publishing Company, Reading, Massachusetts, 1991.
- [QCM93] John S. Quarterman and Smoot Carl-Mitchell. Tutorial: Local protection for networked systems. *UnixWorld*, X(7):64–72, July 1993.
- [RG92] Deborah Russel and G. T. Gangemi Sr. *Computer Security Basics*. O'Reilly & Associates, Inc., 103 Morris Street, Suite A, Sebastopol, CA 95472, July 1992.
- [RM80] Z. Ruthberg and R. McKenzie. Audit and evaluation of computer security. Special Publication 500-19, National Bureau of Standards, Gaithersburg (MD), 1980. Disponible en GPO: SN 003-003-01848-1.
- [Rut80] Z. Ruthberg. Audit and evaluation of computer security ii. Special Publication 500-57, National Bureau of Standards, Gaithersburg (MD), 1980. Disponible en GPO: SN 003-003-02178-4.
- [Spa91] Eugene H. Spafford. The internet worm incident. Technical Report CSD-TR-933, Department of Computer Sciences, Purdue University, West Lafayette, IN USA 47907-2004, September 19 1991.
- [St.93] M. St. Johns. Identification protocol. Request for Comments 1413, Network Working Group, US Department of Defense, February 1993. Disponible en `ftp://nic.ddn.mil/rfc/rfc1413.txt`.
- [Sto88] Clifford Stoll. Stalking the wily hacker. *Communications of the ACM*, 31(5):484–497, May 1988.
- [Sto89] Clifford Stoll. *The Cuckoo's Egg: Tracing a Spy Through the Maze of Computer Espionage*. Doubleday, New York (NY), 1989.
- [Str91] B. J. Stroustrup. *The C++ Programming Language*. Prentice Hall, New York, 1991.
- [War79] W. H. Ware. Security controls for computer systems: Report of defense science board task force on computer security. Doc. No. AD-A076-617/0, R-609-1, Rand Corporation, 1970, declassified 1976, reissued 1979.
- [WS92] Larry Wall and Randal L. Schwartz. *Programming Perl*. O'Reilly & Associates, Inc., 103 Morris Street, Suite A, Sebastopol, CA 95472, first edition, March 1992.

- [Zam93] Diego M. Zamboni. *Manual de instalación de COPS, TCP-Wrapper y passwd+*. Departamento de Supercómputo, Dirección General de Servicios de Cómputo Académico, UNAM, Circuito Exterior, C. U., México D. F., Diciembre 1993. Incluido en el apéndice F de esta tesis.
- [Zam94a] Diego M. Zamboni. Comentarios sobre configuración de reglas en *passwd+*. Publicado originalmente en mensajes distribuidos en la lista de correo electrónico *gasu*, 1994. Disponible por FTP anónimo en `ftp.super.unam.mx:/pub/security/doc/passwd+.comentarios.gz`.
- [Zam94b] Diego M. Zamboni. *TCP-Wrapper: Introducción, instalación y uso*. Publicado originalmente en mensajes distribuidos en la lista de correo electrónico *gasu*, 1994. Disponible por FTP anónimo en `ftp.super.unam.mx:/pub/security/doc/tcp\_wrapper.tutorial.gz`.
- [Zim94] Philip Zimmermann. *PGP User's Guide*. Phil's Pretty Good Software, Oct 1994. Disponible por FTP anónimo en `ftp://ftp.super.unam.mx/pub/security/tools/PGP/`.

# Índice de Materias

- .cshrc, 154
- .gz, 153
- .login, 154
- .profile, 154
- .rhosts, 91, 147
- /bin/false, 150
- /bin/passwd, 148, 160
- /bin/passwd+, 148
- /bin/passwd.old, 160
- /etc, 156, 157
- /etc/cshrc, 150
- /etc/ftpusers, 148
- /etc/hosts.equiv, 91, 147, 154
- /etc/inetd.conf, 156, 157
- /etc/miscd, 157
- /etc/nsyslog.conf, 157, 160, 161
- /etc/passwd, xv, 54, 59, 149, 159
- /etc/passwd.dir, 158
- /etc/passwd.pag, 158
- /etc/passwd.test, 160
- /etc/rc\*, 154
- /etc/syslog.conf, 157, 160, 161
- /etc/syslogd, 161
- /pub, xvi
- /pub/gnu/gzip-1.2.4.tar.Z, 153
- /pub/security/, 115
- /pub/security/crypto/, 115
- /pub/security/crypto/doc/, 115
- /pub/security/crypto/tools/  
    , 115
- /pub/security/disc/, 115
- /pub/security/doc/, 115
- /pub/security/doc/lit/, 115
- /pub/security/gasu/, 116
- /pub/security/tools/, 115
- /pub/security/tools/dgsca/  
    , 112
- /usr/adm/passwd.log, 160
- /usr/etc, 156, 157
- /usr/include/syslog.h, 161
- /usr/lib, 161
- /usr/local/adm/tcpd.log, 157
- /usr/local/etc, 157
- /usr/local/etc/miscd, 157
- /usr/spool/mail, 92
- 1994/, 115
- 4D/35, 36
- 4D/420VGX, 36
- 9000/730, 36
- A, 37
- A12, 30–32
- A9, 30
- aaaa, 92
- aaaa\\_mmm\\_dd, 92, 176
- accesos
  - no autorizados, 6
- ACF/2/MVS, 15
- ACM, 21, 75, 76, 79, 193, 194
- administradores, 3, 5, 6, 8–10, 12, 43–  
    44
- capacitación, 44
- como difusores de la seguridad, 44
- de toda la UNAM, 63
- en máquinas grandes, 44
- en máquinas pequeñas, 44
- falta de interés, 47
- grupo en DGSCA, 62
- ignorancia, 47
- importancia, 44
- ireemplazabilidad, 44
- recursos, 44

- responsables de la seguridad, 44
- tareas, 43, 45
- Agencia de Proyectos de Investigación
  - Avanzada, 21
- Agencia Nacional de Seguridad, 23
- AIX, 72
- alarmas, 5
- aldebaran*, 36, 60
- Alejandro, xvi
- Alemania, 24
- Algoritmos de cifrado
  - DES, 80
  - IDEA, 140
  - MD5, 140
  - RSA, 80, 140
- Allman, Eric, 145
- Alonso y Coria, Alberto, xvii, 79
- altair*, 36
- amenazas, 5–8
  - intencionales, 7–8
  - naturales, 6
  - no intencionales, 6–7
- análisis de riesgos, 13
- andromeda*, 36, 55, 60
- animaciones, 48
- anlpasswd*, 57, 139
- anonymous**, cuenta en Unix, 152, 169
- apostar la vida, 14
- archivo*, variable, 181
- archivo, 170
- Archivos
  - .cshrc, 154
  - .gz, 153
  - .login, 154
  - .profile, 154
  - .rhosts, 91, 147
  - /bin/false, 150
  - /bin/passwd, 148, 160
  - /bin/passwd+, 148
  - /bin/passwd.old, 160
  - /etc, 156, 157
  - /etc/cshrc, 150
  - /etc/ftpusers, 148
  - /etc/hosts.equiv, 91, 147, 154
  - /etc/inetd.conf, 156, 157
  - /etc/miscd, 157
  - /etc/nsyslog.conf, 157, 160, 161
  - /etc/passwd, xv, 54, 59, 149, 159
  - /etc/passwd.dir, 158
  - /etc/passwd.pag, 158
  - /etc/passwd.test, 160
  - /etc/rc\*, 154
  - /etc/syslog.conf, 157, 160, 161
  - /etc/syslogd, 161
  - /pub, xvi
  - /pub/gnu/gzip-1.2.4.tar.z, 153
  - /pub/security/, 115
  - /pub/security/crypto/, 115
  - /pub/security/crypto/doc/, 115
  - /pub/security/crypto/tools/, 115
  - /pub/security/disc/, 115
  - /pub/security/doc/, 115
  - /pub/security/doc/lit/, 115
  - /pub/security/gasu/, 116
  - /pub/security/tools/, 115
  - /pub/security/tools/dgsca/, 112
  - /usr/adm/passwd.log, 160
  - /usr/etc, 156, 157
  - /usr/include/syslog.h, 161
  - /usr/lib, 161
  - /usr/local/adm/tcpd.log, 157
  - /usr/local/etc, 157
  - /usr/local/etc/miscd, 157
  - /usr/spool/mail, 92
  - 1994/, 115
  - aaaa, 92
  - aaaa\\_mmm\\_dd, 92, 176
  - archivo, 170
  - cops, 155
  - cops-104, 155
  - cops-104.tar.gz, 155
  - cops.tar.gz, 170
  - dd, 92, 97

- directorio, 169
- ds5000/1995\\_Jan\\_05, 92
- inetd.conf, 154, 157
- INSTALL, 152, 153
- jtkohl-syslog-complete.
  - tar.gz, 161
- libc.a, 161, 162
- libc\\_G0.a, 161, 162
- Makefile, 156, 159, 160
- maquina, 176
- mmm, 92, 97
- ncarp-x.y, 174
- ncarp.yyyy\\_mmm\\_dd, 97
- nsyslog.conf, 161
- passwd+.tar.gz, 158
- passwd.data, 159
- passwd.h, 159
- permisos de acceso, 13
- pwd.c, 160
- pwsample, 159, 160
- pwsample2, 159, 160
- quickstart, 155
- README, xvi, 115, 134, 152, 156, 158, 159
- README.1, 155
- README.2p1, 155
- README.2sh, 155
- README.3, 155
- README.FIRST, 155
- README.IMPORTANT, 158
- reports/maquina/carp.aaaa\\_
  - mmm\\_dd, 176
- saint, 110
- saint-x.y, 178
- saint.cf, 102, 110
- seguridad.html, xvi
- sys.h, 158, 159
- syslog.conf, 157, 161
- syslog.h, 161
- syslog.o, 161, 162
- tcp\\_wrapper, 156
- TMP\\_FILE, 96, 97
- TMP\\_FILE2, 96, 97
- TMP\\_RPT, 96, 97
- udb, 59
- x.y, 174, 178
- yyyy, 97
- Argonne National Laboratory, 185
- argumento, variable, 179
- ARPA, 21
- Association for Computing Machinery,
  - 21, 75
- at, comando, 150
- AT&T, 15
  - Laboratorios Bell, 29
- Atkins, Todd, 143
- autenticidad, 4
- autenticación, 11
  - de procesos, 11
  - de usuarios, 11
- AWK, lenguaje de programación, 90, 93, 94, 96
- awk, comando, 69
- B, 37
- backdoors, véase puerta trasera
- bajo mundo de la computación, 7
- Baz, Fernando, 30, 32
- becarios de supercómputo, xvii
- Bell, D. E., 22, 23
- Beto, xvi
- bin, comando, 169
- bin, cuenta en Unix, 148
- Bishop, Matt, 140, 143
- bitácoras, 11, 13, 40
- Boeing Aerospace, 15
- Bourne Shell, lenguaje de programación,
  - 90, 130
- Brand, Russell L., 43, 185
- Burroughs, 24, 25, 30
- C, lenguaje de programación, 83, 130
- C Objetivo, lenguaje de programación,
  - 106
- C++, lenguaje de programación, 106
- C2, nivel de seguridad, 31
- cable, 8
- candados, 5, 8
- CANDE, 31
- capella, 36, 55, 60
- Carnegie Mellon University, 143
- CARP, v, ix, 86, 88, 90, 93, 94, 96, 173

- estructura, 90–92
- módulo de análisis (*carp.anlz*), 90–92, 94
- módulo de tabulación (*carp.table*), 90, 92
- módulo principal (*carp*), 90, 92, 96
- casiopea*, 36, 60
- castor*, 33
- CECAFI, 124
- Centro de Ecología
  - Departamento de Biología, 72
  - Departamento de Ecología Terrestre, 72
- Centro de Seguridad en Cómputo del DoD, 23
- Centro de Supercómputo de Carolina del Norte, 66
- Centro Nacional de Seguridad en Cómputo, 24
- CERN, 137
- cerraduras, 20
- CERT, vi, 40–41, 114, 116, 126, 149, 185, 191
  - advisories*, 114
  - distribución, 114
  - estructura, 114
  - consejos, 114
  - coordination center*, 114
  - función, 114
- cert-advisory*, lista de correo electrónico, vi, viii, 114, 163, 164
- CERT/CC, 114
- chantaje, 7
- chapas, 5
- Cherry, Andres, 185
- Chiapas, 70
- CIAC, 116
- científicos, 9
- cifrado de datos, 19
- cintas, 5, 7
- Cisneros, Gerardo, xvi
- Ciudad Universitaria, 27
- CMW+, 15
- Coahuila, 70
- color naranja, 14
- comando**, comando, 170
- Comandos
  - at**, 150
  - awk**, 69
  - bin**, 169
  - comando**, 170
  - compress**, 153
  - crontab**, 149, 150
  - cs**, 149
  - finger**, 55, 142, 155
  - fingerd**, 149
  - ftp**, 106, 155, 169
  - ftpd**, 133, 141, 156
  - grep**, 69
  - inetd**, 142, 156
  - lastcomm**, 143
  - login**, xv, 54, 141
  - make**, 174, 178
  - ncarp**, 178
  - nslookup**, 144
  - pack**, 153
  - passwd**, 58
  - rexd**, 154
  - rexecd**, 141
  - rlogin**, 55, 102, 106, 111, 155
  - rlogind**, 141
  - rsh**, 55, 102, 103, 106, 111, 142, 149, 155
  - rshd**, 141, 149
  - saint.cf**, 181
  - sed**, 69
  - sort**, 69
  - su**, 40, 102
  - syslog**, 112, 148, 156
  - talk**, 55, 155
  - tar**, 134
  - telnet**, 55, 106, 141, 142, 155, 167
  - telnetd**, 141, 156
  - umask**, 150
  - uudecode**, 154
  - write**, 40
- comp.security**, grupo de USENET, 185
- comp.security.misc**, grupo de USENET, 185
- compress**, comando, 153
- computadoras
  - facilidad de acceso, 9

- globalización, 20
- inicios, 9
- omnipresencia, 9
- puntos vulnerables, 9
- computadoras personales, xiii
  - abundancia, 32
  - en la UNAM, 32
  - impacto en la seguridad, 32
  - problemas, 39
  - ventajas, 32
- Computer Associates International, 15
- Computer Emergency Response Team, 114, 185
- cómputo
  - administrativo, 30, 37–38
    - cambio a Unix, 32
    - seguridad, 37–38
  - bajo mundo, 7
  - como forma de manejo de información, 19
  - como herramienta para la seguridad, 46
  - como privilegio, 19
  - como recurso útil, 20
  - conocimientos, 20
  - desarrollo en la UNAM, 28
  - en los planes de estudio, 20
  - evolución, 19–20
  - función principal, 46
  - grandes máquinas, 19
  - grandes progresos, 20
  - infallibilidad, 46
  - infatigabilidad, 46
  - inicios, 19
  - redes, 20
  - salas de, 20
  - sistemas poderosos, 20
- COMSEC, 21
- CONACYT, 70
- Conferencia Conjunta de Cómputo de Primavera*, 21
- Conferencia de Seguridad en Cómputo y Comunicaciones*, 79
- confidencialidad, 4
- configuración
  - errónea, 6
- Consejo de Ciencias de la Defensa, 21
- Consejo de Seguridad en Comunicaciones, 21
- consejos
  - del CERT, 114
- consola, 41
- contramedidas, 5, 8–9
- Control Data, 30
- control de acceso, 11
- controles
  - de la información, 19
  - de la tecnología, 19
  - gubernamentales, 19
- convenciones tipográficas, xv
- Cooper, Michael, 144
- COPS, v, vii, ix, xv, 41, 57, 60, 61, 64, 69, 70, 73, 86–88, 90–96, 99, 121, 124, 143, 144, 149, 151, 153–155, 173–177, 185, 195
  - problemas, 86
- cops, 155
- cops-104, 155
- cops-104.tar.gz, 155
- cops.tar.gz, 170
- Corporación Mitre, 23
- Corporación RAND, 21
- Corporation for Research and Educational Networking, 130
- correo electrónico, 47, 61
  - facilidad de interceptación, 121
- Courtney, 142, 145
- cpm, 143
- Crack, 41, 57, 60, 70, 99, 139, 149, 177, 185
- cracker*
  - definición, 7
- crackers, 34
- Cracklib, 139
- Cray, xiii, 12, 28, 30, 32–34, 36, 40, 41, 53, 58–60, 66, 72, 127
- Cray Research de México, xvi, 121
- Cray Research, Inc., 15, 30, 34, 40–42, 55
- CREN, 130
- criptografía, 11
- crontab**, comando, 149, 150

- crypt*, xv  
 CSC, 23, 24  
**cs**h, comando, 149  
 Cuentas en Unix  
     **anonymous**, 152, 169  
     **bin**, 148  
     **dg**sca, 148  
     **god**, 40, 41  
     **jlm**, 40  
     **oss**, 58, 61  
     **root**, xv, 39–41, 58, 88, 91, 102, 111, 149, 150, 152, 160, 175, 181  
     **sys**, 148  
 Curry, David A., 144, 185  
 Cursos  
     Temas Avanzados de Seguridad en UNICOS, 83  
*cvyd*, lista de correo electrónico, 67  
*cygnus*, 36  
  
 Día Internacional de la Seguridad en Computo, xiv  
*Día Internacional de la Seguridad en Computo*, v, ix, 75–77, 113, 115–118, 165  
 daños  
     físicos, 6, 20  
     minimización, 8  
     prevención, 8  
     sin rastro, 20  
 Daltabuit Godás, Enrique, 40, 64, 79, 80  
 Daltabuit Godás, Enrique, xvii  
 DARPA, 21  
 Data Encryption Standard, *véase* DES  
*DBMLIB*, variable, 158, 159  
 DCAA, 32  
 dd, 92, 97  
 De Leo, Mike, xvii  
 DEC, 22, 30, 35, 153  
*DEFPWFILE*, variable, 159  
 DeHart, Ed, 114  
*deneb*, 36, 60  
 Departamento de Apoyo a Sistemas Unisys, 30  
 Departamento de Defensa de los Estados Unidos de Norteamérica, 14, 22  
 Departamento de Defensa de los Estados Unidos, 21  
 Departamento de Matemáticas, 121  
 Departamento de Monitoreo Tecnológico, 120  
 Departamento de Operación, 24  
 Departamento de Redes, 62, 121  
 Departamento de Supercómputo, iv, vii, xi, xvi, 33–36, 40, 41, 45, 53, 55, 57, 58, 60, 62, 112, 115–117, 121, 122, 126, 129–131, 133, 134, 137, 138, 170, 173, 177  
 DEPFI, 41  
 DES, 22, 23, 140  
 DES, algoritmo de cifrado, 80  
 desastres naturales, 8, 13  
 desvío de mensajes, 6  
 DGAE, 31, 32  
 DGSCA  
     Departamento de Redes, 62, 121  
     Departamento de Supercómputo, iv, vii, xi, xvi, 33–36, 40, 41, 45, 53, 55, 57, 58, 60, 62, 112, 115–117, 121, 122, 126, 129–131, 133, 134, 137, 138, 170, 173, 177  
     Laboratorio de Visualización, iv, xi, xvi, 33–36, 40, 45, 53, 55, 57, 58, 60, 62, 121, 122, 126, 138  
 DGSCA, iv, xi, xiv, xvii, 24, 33, 34, 45, 53–55, 57, 58, 60, 62, 64, 66, 76, 79–82, 99, 121, 123–128, 131, 138, 165, 173, 177, 183, 192  
**dg**sca, cuenta en Unix, 148  
 DGSCAD, 31  
 Dig, 144  
 Digital Equipment Corporation, 15, 22, 161  
 dinero, 7, 13  
 Dios, 40  
*diphda*, 36, 60

- Dirección de Cómputo para la Administración Académica, 30, 31
  - Coordinación Técnica de Redes e Interoperabilidad, 71
- Dirección de Sistemas de Seguridad de la Información, 24
- Dirección General de Administración Estudiantil, 31
- Dirección General de Servicios de Cómputo Académico, 45
  - Departamento de Administración de Supercómputo, 71
  - Departamento de Redes y Telecomunicaciones, 71
  - Laboratorio de Visualización, 71
- directorio, 169
- DISC, viii, 127, 128, 165
  - el futuro, 77
  - en 1993, 76
  - en 1994, 76–77
    - convocatoria, 76
    - realización, 76–77
    - respuesta, 76
  - objetivo, 75
  - organización, 75
  - origen, 75
- discos, 5, 7
- diseño orientado a objetos, 106
- diskettes, 5
- disponibilidad, 4
- dispositivos, 5
- dispositivos biométricos, 8
- dispositivos electrónicos
  - ventajas, desventajas y debates, 19
- diversión, 7
- División de Estudios de Posgrado de la Facultad de Ingeniería, 41
  - Departamento de Ingeniería Eléctrica, 72
  - Departamento de Ingeniería en Informática, 72
  - Laboratorio de Cómputo Avanzado, 71
- DNS, 68, 131, 144
- DoD, 14, 15, 21–23
- DoD Computer Security Center, 23
- Domain Name Service, 69, 131
- dominio*, variable, 181
- dominios
  - en SAINT, 105
- ds5000*, xv, 35, 55, 59–61, 131, 134, 138
- ds5000/1995\\_Jan\\_05*, 92
- Durán, Rafael, xvii, 24, 25
- emanaciones electromagnéticas, 6
- Embajada de los Estados Unidos en México, 42
- enlaces satelitales, 28
- enrutadores, 27
- equipos de tigres, 22
- Eriksson, Peter, 140
- errores
  - humanos, 41
- esfuerzo, 13
- estaciones de trabajo, xiii
- Estados Unidos, 21–24, 66, 114, 126, 140, 183
- Estados Unidos de Norteamérica, 14
- estudiantes, 45
- Ethernet, 28, 33, 34
- evolución
  - de la computación, 19–20
- expresiones regulares, 91, 94
- Facultad de Ciencias, 41, 121
- Facultad de Ingeniería, xiii
- fallas de energía eléctrica, 6
- falsificación de mensajes, 6
- FAQ, 74
- Farmer, Dan, 82, 112, 142, 143, 185
- FDDI, 27, 33
- Federal Information Processing Standard, 23
- fibra óptica, 27
- fiebre tecnológica, 12
- File Transfer Protocol, 48
- fn*, variable, 181
- finanzas personales, 12
- finger**, comando, 55, 142, 155
- fingerd**, comando, 149
- FIPS, 23
- firewall, 33

- firewalls*, 10
- formas, 49
- Foro de Consulta Popular sobre Informática*, 120
- FORTTRAN**, lenguaje de programación, 83
- FTP, iii, vi–ix, xv, xvi, 35, 48, 49, 55, 61, 65, 69, 76, 99, 102, 103, 112, 114, 115, 133, 134, 139, 141, 142, 145, 148–150, 152, 153, 161, 169–171, 178, 195
- ftp**, comando, 106, 155, 169
- FTP anónimo, 48
- ftpd**, comando, 133, 141, 156
- Funciones
  - crypt*, xv
  - print\_msg()*, 94, 95
- Garfinkel, Simson, 82, 185
- gas neurotóxico, 14
- gastos
  - millonarios, 14
- GASU, v, ix, 62, 66–71, 73, 74, 76, 114, 116, 117, 119, 121, 124, 125, 127, 128, 147, 151
  - acciones iniciales, 64
  - actividades, 66, 68–69
  - coordinadores, 64
  - difusión de herramientas, 69–70
  - el futuro, 74
  - encuesta, 71
  - enfoque inicial, 63
  - estado actual, 70
  - fundación, 63–65
  - lista de correo electrónico, 66–67
    - antecedentes, 67
    - gasu*, 67
    - ¿por qué?, 66–67
    - unam-admin*, 67
  - medios de difusión inicial, 64
  - nuevas áreas de acción, 68
  - opiniones, 71–74
  - ¿para qué?, 64
  - primera reunión, 64–65
    - resultados, 65
  - reporte de incidentes, 66
    - segunda reunión, 65–66
    - resultados, 66
- gasu*, lista de correo electrónico, vi, viii, xv, 67, 69, 71, 113, 114, 116, 131, 132, 163, 187, 195
- gobierno, 21, 22, 24
- Gobierno de los Estados Unidos, 14, 22, 23
- Gobierno de los Estados Unidos de Norteamérica, 14
- god**, cuenta en Unix, 40, 41
- GOPHER, 49, 192
- Gould, 15
- grep**, comando, 69
- Grupo de Administración y Seguridad en Unix, 67
- Grupo de Administración y Seguridad en Unix, 62
- Grupo de Interés Especial en Seguridad, Auditoría y Control, 75
- Grupos de USENET
  - comp.security**, 185
  - comp.security.misc**, 185
- Guadalajara, 66
- Guanajuato, 70
- GUARD, 25
- guardias, 14, 20
- Guerra Ortiz, Victor, xvii, 76, 79
- gzip, 134, 153
- habilidad, 7
- hacker, 7
- hacker*
  - definición, 7
- Haller, Neil M., 141
- hardware*
  - errores, 4
- Haugh, II, John F., 140
- HBO, 3
- Henderson, Mark, 185
- herramientas
  - análisis, 55, 57
  - análisis necesario, 85
  - asesorías, 69
  - CARP, 86–88
  - comerciales, 46

- concentración de reportes, 60–61
- conjunción de, 85
- COPS, 86–88
- critérios de selección, 57
- dónde obtenerlas, 112
- de difusión de información, 47–49
- de dominio público, 46
- desarrollada en DGSCA, 86
- desarrolladas localmente, 47
- distribución, 69
- distribución por máquinas, 59
- en español, 86
- entendibles, 86
- estado actual, 60
- información, 69
- instalación en DGSCA, 58
- mala utilización, 86
- manuales, 69
- metodo de instalación, 59–60
- NCARP, 86–98
- necesidades en DGSCA, 55–56
- necesidades nuevas, 85
- New CARP, *véase* NCARP
- objetivo, 60
- obstáculos para su utilización, 120
- privilegios, 58
- problema lingüístico, 85
- SAINT, 99–112
- tutoriales, 69
- Hewlett-Packard, 15, 30, 76
- hipertexto, 48
- historias académicas, 30
- Honeywell Federal Systems, 15
- Honeywell Information Systems, 15
- Hoover, Clyde, 140
- Hotz, Steve, 144
- HP, 36
- HP-UX, 30, 36, 72
- HTML, 49, 109, 117, 137, 138, 163, 192
  - formas, 49
- HTTP, vii, xvi, 49, 137, 138
- httpd, 137, 138
- humedad, 5
- HyperText Markup Language, 49
- HyperText Transfer Protocol, 49
- IBM, 15, 30
- IBM-650, 28
- IDEA, algoritmo de cifrado, 140
- ignorancia, 6
- IIMAS, 31
- imágenes, 48
- imanes, 7
- impresos confidenciales, 8
- incendios, 5, 6, 13
- Indigo 2, 36
- Indy, 35, 36
- INEGI, 120
- inetd**, comando, 142, 156
- `inetd.conf`, 154, 157
- InfoGuard, 31
- información
  - abusos, 20
  - cifrado, 31
  - controles, 19
  - en transmisión, 8, 11
  - facilidad de acceso, 20
  - importancia, 12
  - inaccesible, 12
  - medidas gubernamentales, 19
  - origen, 11
  - qué se tiene, 12
  - respaldos, 13
  - valor, 19
- INFOSEC, 24
- Ingeniero en Computación, xiii
- Ingham, Kenneth, 144
- Inglaterra, 24
- Iniciativa de Seguridad en Cómputo del DoD, 22, 23
- inicio*, variable, 181
- INSTALL, 152, 153
- Instituto de Astronomía, vi, 66–68, 123, 124
  - Departamento de Cómputo, 71
- Instituto de Ciencias Nucleares, 40
  - Departamento de Cómputo, 71
- Instituto de Física, 121
- Instituto de Geofísica, 67
- Instituto de Química
  - Departamento de Cómputo, 71

- Instituto Nacional de Astrofísica, Óptica y Electrónica  
 Departamento de Astrofísica, 72  
 Instituto Nacional de Estadística, Geografía e Informática, 120  
 integridad, 4, 11  
 interceptación de mensajes, 6  
 Internet, 3, 28, 33, 34, 37, 46–48, 56, 69, 82, 86, 129, 133, 137  
 intrusos, 8  
     decididos, 7  
     externos, 7  
     internos, 7–8  
     motivos, 7  
     ocasionales, 7  
 inundaciones, 5  
 investigación científica en México, 12  
 investigadores, 45  
 IP, 145  
 ipacl, 141  
 Irix, 30, 35, 36, 72  
 ISI, 194  
 ISS, 143  
 ITAM, 70
- Jacobson, Van, 145  
 Jalisco, 70  
 Japón, 21  
 Jiménez, Victor, xvii  
**jlm**, cuenta en Unix, 40  
**jdkohl syslog**, 148  
**jdkohl-syslog-complete.tar.gz**, 161
- Karn, Phil, 141  
 Kerberos, 57, 80, 140, 185  
*kernel* de seguridad, 22  
 KGB, 3  
 Kim, Eugene, 82, 144  
 Klaus, Christopher, 143  
 Koch, Bryan, 40  
 Kotsikonas, Anastasios, 130
- línea telefónica, 20  
 líneas conmutadas, 27  
 líneas privadas, 27
- Laboratorio de Visualización, iv, xi, xvi, 33–36, 40, 45, 53, 55, 57, 58, 60, 62, 121, 122, 126, 138  
 Laboratorio Nacional de Argonne, 139  
 Laboratorio Nacional de Los Alamos, vi, 126, 183  
 Laboratorios Bell de AT&T, 29  
 Lacambra Macedo, Rafael, 69  
 Lali, xvi  
 LAN Manager, 28, 39  
 LANL, 126  
 Lantastic, 28  
 LaPadula, L. J., 22, 23  
**lastcomm**, comando, 143  
 Lenguajes de programación  
     **AWK**, 90, 93, 94, 96  
     **Bourne Shell**, 90, 130  
     **C**, 83, 130  
     **C Objetivo**, 106  
     **C++**, 106  
     **FORTRAN**, 83  
     orientados a objetos, 106  
     **perl**, 100, 106, 130, 178  
     **perl5**, 106, 110, 178  
     **Smalltalk**, 106  
**LG\_OUTDEF**, variable, 159  
*libc.a*, 161, 162  
*libc\\_g0.a*, 161, 162  
 Libro Naranja, 14–17  
     características de las clases definidas, 15–16  
     divisiones definidas, 14–15  
     motivación, 14  
     propósito, 15–17  
     sistemas aprobados, 15  
 Linux, 30, 72  
*lista*, lista de correo electrónico, 131, 132  
 Listas de correo electrónico, 35, 47–48  
     *cert-advisory*, vi, viii, 114, 163, 164  
     *cvyd*, 67  
     *gasu*, vi, viii, xv, 67, 69, 71, 113, 114, 116, 131, 132, 163, 187, 195  
     *lista*, 131, 132  
     *unam-admin*, 67  
     ventajas, 47–48

- ListProc, 129–131  
 LISTSERV, 129  
 Llamadas al sistema  
   *read*, xv  
 llaves, 8  
*log*, 11, 13  
 logdaemon, 141  
 login, 25  
**login**, comando, xv, 54, 141  
 Lynx, xvi, 49, 138
- Máquinas  
 A, 37  
   *aldebaran*, 36, 60  
   *altair*, 36  
   *andromeda*, 36, 55, 60  
 B, 37  
   *capella*, 36, 55, 60  
   *casiopea*, 36, 60  
   *castor*, 33  
   *cygnus*, 36  
   *deneb*, 36, 60  
   *diphda*, 36, 60  
   *ds5000*, xv, 35, 55, 59–61, 131, 134, 138  
   *mezcal*, 35, 60, 131, 134, 138  
   *mira*, 36, 55, 60  
   *nocdos*, 36, 55, 60  
   *pegasus*, 36, 60  
   *polaris*, 36, 55, 60  
   *pollux*, 33  
   *pulque*, 35, 60  
   *roxanne*, 40  
   *sirio*, 54, 60  
   *tepache*, 35  
   *tequila*, 35, 60  
   *tzetzal*, 32  
   *xtabentun*, 35, 60
- México, 24, 73, 77, 85, 114, 120, 125, 126, 165  
*módulo*, variable, 179  
 Macintosh, 15, 28, 73  
 macrocomputadoras, 28  
 mainframes, 28  
   característica principal, 30  
   en la UNAM, 30–32  
   en uso activo, 30  
   la Cray, 30  
   requerimientos físicos, 30
- Majordomo, 129, 130  
**make**, comando, 174, 178  
 Makefile, 156, 159, 160  
 Mallén, Guillermo, 81  
*maquina*, variable, 181  
 maquina, 176  
 Maquinas  
   en SAINT, 105  
 Martha, xvi  
 Mathematics and Computer Science Division, 185  
 McIver, Jeff, 81  
 MCP, 30  
 MD5, algoritmo de cifrado, 140  
 MD5, 141  
 media noche, 13  
 Medina, Gaby, xvii  
 mensajes  
   desvío, 6  
   falsificación, 6  
   intercepción, 6  
*mezcal*, 35, 60, 131, 134, 138  
 microcomputadoras, 28  
 microondas, 27  
 minicomputadoras, 28  
*mira*, 36, 55, 60  
 Mitre Corporation, 22  
 MLS, 80  
 mmm, 92, 97  
 Mockapetris, Paul, 144  
 modelo de seguridad, 22  
 modificaciones  
   accidentales, 4  
   intencionales, 4  
   no autorizadas, 11  
 modularización  
   usada en SAINT, 101–102  
 Mogul, Jeff, 144  
 Monterrey, 66, 70  
 Mosaic, xvi, 49, 138  
 MPE V/E, 15  
 MS-DOS, 15, 32, 39, 73  
 Muffet, Alec D., 185

- Muffett, Alec, 139
- Muffett, Alec D., 139, 185
- Multics, 15, 22
- multimedia, 48
- MVS, 30
- MVS/ESA, 15
- MVS/RACF, 15
  
- nóminas, 30
- National Bureau of Standards, 21
- National Center for Supercomputing Applications, 137
- National Computer Security Center, 24
- National Security Agency, 23
- NBS, 21, 22
- NCARP, v, vi, viii, ix, 61, 86–98, 110, 121, 173–176, 178, 179
  - archivo de configuración, 97–98
  - archivo de configuración (`ncarp.mail`), 97, 176
  - base de datos, 93
  - el futuro, 98
  - módulo de análisis (`ncarp.anlz`), 94–97
  - módulo de tabulación (`ncarp.table`), 96–97
  - módulo principal (`ncarp`), 96–98, 174
  - qué hace, 88–90, 93
- ncarp**, comando, 178
- `ncarp-x.y`, 174
- `ncarp.yyyy\_mmm\_dd`, 97
- NCSA, 23, 137, 138
- NCSC, 24, 66, 123
- Needham, R.M., 140
- Netlog, 142
- NetScape, xvi, 49, 138
- NetWare, 28, 39
- Network Information Service, 140
- NEXTSTEP, 72
- NFS, 36, 144, 149
- NFS Watch, 144
- niños, 9
- NIS, 36, 140, 185
- Nivel de seguridad
  - C2**, 31
- nocdos*, 36, 55, 60
- North Carolina Supercomputing Center, 123
- NOS/VE, 30
- `npasswd`, 57, 140
- NSA, 23, 24
- NSC, 33, 34
- NSDD 145, 23
- nslookup**, comando, 144
- `nsyslog.conf`, 161
- `nullshell`, 148
  
- O'Connor, Bryan D., 145
- O'Reilly & Associates, Inc., 185
- objetos
  - dominios de seguridad, 107–108
  - en SAINT, 106–108
  - maquinas, 106–107
- Oficial de Seguridad del Sistema, 58
- Oficina del Secretario de la Defensa, 23
- Oficina Nacional de Normas, 21
- operadores, 20
- Orange Book*, véase Libro Naranja
- origen de esta tesis, 25
- Orozco, José Luis, xvii
- OS 1100, 15
- OSF, 29
- OSF/1, 30
- oss**, cuenta en Unix, 58, 61
  
- pérdida
  - de confianza, 13
  - monetaria, 13
- Países
  - Alemania, 24
  - Estados Unidos, 21–24, 66, 114, 126, 140, 183
  - Estados Unidos de Norteamérica, 14
  - Inglaterra, 24
  - Japón, 21
  - México, 24, 73, 77, 85, 114, 120, 125, 126, 165
- pack**, comando, 153
- passwd**, comando, 58

- passwd+, vii, xv, 57–60, 64, 69, 70, 99, 124, 140, 148, 151, 158–160, 177, 195
- passwd+.tar.gz, 158
- passwd.data, 159
- passwd.h, 159
- password, 25
- passwords, 54–59
  - control, 57, 58
  - importancia, 55
  - protección, 56–58
  - revisión, 56
  - rompimiento, 57
- PC's, 28, 32
- PDP-11/45, 22
- PDP-11/70, 22
- pegasus, 36, 60
- periódicos
  - primeras planas, 7
- perl**, lenguaje de programación, 100, 106, 130, 178
- perl5**, lenguaje de programación, 106, 110, 178
- permisos de archivos, 56
- personal
  - ambicioso, 7
  - autorizado, 11
  - de mantenimiento, 8
  - directivo, 13, 45
    - qué es, 45
  - distraído, 8
  - inconforme, 7
  - interno, 7
- PF\_PLATE*, variable, 160
- PF\_TEMP*, variable, 160
- PGP, 98, 121, 140, 141
- pidentd, 140
- Pinilla, Victor, 124
- política de seguridad, 22
- polaris*, 36, 55, 60
- pollux*, 33
- polvo, 5
- portmap, 142
- POSIX, 29
- potencia de cálculo, 10
- Presidente de la República, 12
- print\_msg()*, 94, 95
- prioridades, 12
- privilegios de usuarios, 56
- problemas de frontera, 10
- problemas de seguridad
  - en 1975, 24
  - falta de combate organizado, 24
  - larga existencia, 24
- procesos
  - autenticación, 11
  - autorizados, 11
- productividad, 9
- Programa Universitario de Seguridad en Cómputo, 128
- Programas
  - anlpasswd, 57, 139
  - CANDE, 31
  - CARP, v, ix, 86, 88, 90, 93, 94, 96, 173
  - COPS, v, vii, ix, xv, 41, 57, 60, 61, 64, 69, 70, 73, 86–88, 90–96, 99, 121, 124, 143, 144, 149, 151, 153–155, 173–177, 185, 195
  - Courtney, 142, 145
  - cpm, 143
  - Crack, 41, 57, 60, 70, 99, 139, 149, 177, 185
  - Cracklib, 139
  - Dig, 144
  - gzip, 134, 153
  - httpd, 137, 138
  - InfoGuard, 31
  - ipacl, 141
  - ISS, 143
  - jt Kohl syslog, 148
  - Kerberos, 57, 80, 140, 185
  - ListProc, 129–131
  - LISTSERV, 129
  - logdaemon, 141
  - Lynx, xvi, 49, 138
  - Majordomo, 129, 130
  - MD5, 141
  - Mosaic, xvi, 49, 138
  - NCARP, v, vi, viii, ix, 88–90, 93, 94, 96–98, 110, 121, 173–176,

- 178, 179
- Netlog, 142
- NetScape, xvi, 49, 138
- NFS Watch, 144
- npasswd, 57, 140
- nullshell, 148
- passwd+, vii, xv, 57–60, 64, 69, 70, 99, 124, 140, 148, 151, 158–160, 177, 195
- PGP, 98, 121, 140, 141
- pidentd, 140
- portmap, 142
- rdist, 144
- RIACS, 143
- S/Key, 57–60, 70, 80, 141
- SAINT, vi, viii, ix, 61, 86, 99–102, 104, 106, 107, 109–112, 177–181
- SATAN, vi, viii, 57, 58, 60, 112, 120, 121, 142, 187
- sendmail, 145
- shadow, 140
- SmartList, 129
- Snefru, 141
- Spar, 143
- sra, 141
- Swatch, 143
- TCP-Wrapper, vii, 57, 60, 64, 69, 73, 99, 112, 124, 142, 148, 151, 155–158, 177, 195
- tcpdump, 145
- Tiger, 57, 144
- Traceroute, 145
- TripWire, 57–60, 70, 99, 144, 177
- Watcher, 144
- wu-ftpd, ix, 133, 135, 145
- Xinetd, 57, 142
- Protocolos
  - FDDI, 27, 33
  - FTP, iii, vi–ix, xv, xvi, 35, 48, 49, 55, 61, 65, 69, 76, 99, 102, 103, 112, 114, 115, 133, 134, 139, 141, 142, 145, 148–150, 152, 153, 161, 169–171, 178, 195
  - GOPHER, 49, 192
  - HTTP, vii, xvi, 49, 137, 138
  - requerimientos, 28
  - SECURE RPC, 57, 80, 141
  - TCP, 141, 142
  - TCP/IP, iv, 28, 32, 69
  - TFTP, 149, 154
  - UDP, 141, 142
  - UUCP, 149
- proyecto de seguridad, 29
  - alcance, 53
  - antecedentes, 27
  - apoyo, 79
  - elementos involucrados, 27
  - FTP anónimo, 48
  - inicio oficial, 54
  - Listas de correo electrónico, 48
  - máquina coordinadora, 35
  - objetivo, xiii
  - origen, 25
  - primer objetivo, 34
  - primeras medidas, 53
  - recursos humanos, 43–45
  - recursos legales, 49–50
  - recursos técnicos, 46–49
  - requerimientos para impulsarlo, 79
  - sistemas relacionados, 34–35
- puerta trasera, 40, 41, 54
  - detección, 54
  - técnicas, 54
- puertas, 5
  - pulque*, 35, 60
- puntos de acceso, 10
- pwd.c, 160
- pwsample, 159, 160
- pwsample2, 159, 160
- PWTESTFILE*, variable, 159
- quickstart, 155
- radio módem, 27
- Rafael, xvi
- RAND, 21
- rayos, 6
- RDI, 27
- rdist, 144
- read*, xv

- README, xvi, 115, 134, 152, 156, 158,  
     159  
 README . 1, 155  
 README . 2pl, 155  
 README . 2sh, 155  
 README . 3, 155  
 README . FIRST, 155  
 README . IMPORTANT, 158  
 Reagan, Ronald, 23  
 Real, Nombre, 131, 132  
 recursos, 4  
     comerciales, 46  
     de dominio público, 46  
     desarrollados localmente, 47  
     humanos, 43–45  
     informativos, 47–49  
     legales, 49–50  
     técnicos, 46–49  
 Red Digital Integrada, 27  
 Red Uno, 76  
 redes  
     de coaxial delgado, 28  
     de coaxial grueso, 28  
     de Token Ring, 28  
     en SAINT, 105  
     Internet, 28, 33, 34, 37, 46–48, 56,  
         86  
     nombre de máquinas, 34  
     par trenzado, 28  
     tipo estrella, 28  
 redes de computadoras, 9, 10  
     inexistencia, 12  
 redes locales, 27  
 RedUNAM  
     estructura general, 27–28  
     medios de enlace, 27–28  
     protocolos, 28  
     qué es, 27  
     topología principal, 27  
     topologías comunes, 28  
 REDUNAM, iv, 27, 28, 31–34, 37, 67,  
     192  
 reinstalación de sistema operativo, 53–  
     55  
 rejas, 5  
 reportes  
     concentración, 60–61  
     procesamiento, 61  
     transferencia, 61  
 reports/maquina/carp.aaaa\  
     \_mmm\\_dd, 176  
 respaldos, 13, 20, 31  
 revolución computacional, 14  
**rexd**, comando, 154  
**rexecd**, comando, 141  
 RIACS, 143  
 RIACS, 143  
 riesgos  
     análisis, 12–14  
     comunes, 13  
     reales, 12  
 Ritchie, Dennis, 29  
 Rivest, Ronald L., 141  
**rlogin**, comando, 55, 102, 106, 111, 155  
**rlogind**, comando, 141  
**root**, cuenta en Unix, xv, 39–41, 58, 88,  
     91, 102, 111, 149, 150, 152,  
     160, 175, 181  
*ROOTID*, variable, 159, 160  
*roxanne*, 40  
 RSA, algoritmo de cifrado, 80, 140  
**rsh**, comando, 55, 102, 103, 106, 111,  
     142, 149, 155  
**rshd**, comando, 141, 149  
  
 S/Key, 57–60, 70, 80, 141  
 Sánchez Cerezo, Martha A., 40, 64, 66,  
     69, 79, 80  
 Sacristán Ruiz-Funes, Eduardo, xvii, 67,  
     68  
 Safford, Dave, 141  
 SAINT, vi, viii, ix, 61, 86, 99–112, 177–  
     181  
     análisis de eventos, 104–109  
         técnicas consideradas, 104  
     análisis gramatical, 104–105  
         desventajas, 104–105  
     archivo de configuración, 110–111  
     características, 100  
     consideraciones, 111–112  
     el futuro, 112  
     estado actual, 99

- eventos
  - ftp*, 102
  - reboot*, 102
  - rlogin*, 102
  - rsh*, 102
  - su\_root*, 102
  - su\_user*, 102, 103
  - telnet*, 102
- formato común de datos, 102–103
- homogeneización de datos, 101–103
- limitaciones, 111–112
- módulo de control, 110
- objetos, 106–108
- operación, 100
- ordenamiento de eventos, 104
- ¿por qué?, 99
- presentación de resultados, 109–110
  - formatos, 109
- ¿qué hace?, 99–100
- recolección de datos, 101–103
- simulación de eventos, 105–109
  - elementos, 105
  - implementación, 105–109
  - pasos a seguir, 108
  - qué es, 105
  - resultados producidos, 108–109
- saint*, 110
- saint-x.y*, 178
- saint.cf***, comando, 181
- saint.cf*, 102, 110
- sala de Cray, 34
- salida estándar, 94, 97
- salida estandar, 91
- Santa Cruz Operation, 30
- SATAN, vi, viii, 57, 58, 60, 112, 120, 121, 142, 187
- Schales, Dough, 143, 144
- Schroeder, M.D., 140
- SCO Unix, 30, 72
- SCOMP, 15
- SCT, 70
- señales electrónicas, 6
- Secretaría de la Defensa, 77
- Secretaría de la Defensa para la Investigación y la Ingeniería, 22
- SECURE*, variable, 155
- SECURE RPC, 57, 80, 141
- SECURE\_USERS*, variable, 155
- SecureWare, 15
- sed**, comando, 69
- seguridad
  - en redes, 10–11
  - en supercómputo, 10–12
  - en Unix, 10
  - kernel* de, 22
- seguridad de la información, 19
- seguridad en cómputo
  - accesos no detectados, 53
  - adquisición de equipo, 17
  - alcance, 3
  - alto nivel, 6
  - amenazas, 6–8
    - intencionales, 7–8
    - naturales, 6
    - no intencionales, 6–7
  - amenazas originales, 9
  - análisis de incidentes, 41
  - análisis de riesgos, 12–14
  - asesorías, 123–124
  - ataques a centros de supercómputo, 39
  - atracción del supercómputo, 39
  - becarios, 83
  - beneficios de la información, 63
  - bibliografía, 81
  - bitácoras, 11, 13, 32
  - carpeta de, 82
  - causas de ataques exitosos, 41
  - como actividad independiente, 12
  - como cajas negras, 24
  - como campo teórico, 53
  - como herramienta en aplicaciones
    - reales, 14
  - como moda, 19
  - como pérdida de tiempo, 43
  - como problema cultural, 47
  - como problema humano, 47
  - como solución real, 45
  - compromiso, 13
  - confianza, 37
    - ejemplo, 37
  - contramedidas, 8–9

- control de acceso, 55, 57
- control de passwords, 55
- controles de acceso, 31
- costos, 12–13
- criptografía, 11
- cuentas desprotegidas, 45
- de una planta industrial, 20
- definición, 3–4
- definición del problema, 27
- descuidos, 7–8
- diferentes requerimientos, 14
- discrecional, 14
- divisiones, 14–15
- educación, 9–10
- elemento principal, 43
- elementos, 5–9
- en comunicaciones, 8
- en los bancos, 24
- en los sistemas Unisys, 25
- en los sistemas Wang, 25
- en sistemas Unisys, 30–32
- equilibrio, 13
- equipo especializado, 13
- evaluación, 27
- expertos en, 82
- física, 8, 11, 20
- falta de legislación, 25
- frecuencia de incidentes, 39
- grupos internacionales, 50
- guía, 17
- herramientas, 13
- historia a nivel mundial, 21–24
- historia en la UNAM, 24–25
- humana, 8–9
- impacto de las PC's, 32
- importancia de la comunicación, 47
- importancia de la información, 113
- importancia de los administradores, 44
- importancia de los servicios, 113
- importancia del apoyo directivo, 45
- importancia del trabajo legal, 123
- imposibilidad, 13, 14
- incidentes, 13
- incidentes famosos
  - Ataque a HBO, 3
  - Conexión KGB, 3, 13
  - Gusano de Internet, 3
- incidentes previos en la UNAM, 39–42
- indiferencia, 43
- inicios, 21
- integridad, 56
- interna, 8
- legislación universitaria, 124–125
- mínima, 14
- mecanismos tradicionales, 5
- medición, 15
- moda, 10
- necesidad de ser institucional, 45
- obligatoria, 14
- por byte, 25
- por oscuridad, 38
- preguntas iniciales, 12–13
- prevención de los problemas, 9
- primer estándar, 21
- primera organización gubernamental, 21
- primera presentación detallada, 21
- prioridades, 12
- problemas cotidianos, 3
- problemas debidos a malos diseños, 29
- problemas políticos, 25
- productos, 13
- reglamentación, 14–17
  - Libro Naranja, 14–17
- respaldos, 31
- respuesta a incidentes, 40–41
- revisión de integridad, 57
- revisión general, 57
- seminarios, 80–81
- situación inicial en DGSCA, 53
- situación original, 9
- soluciones tecnológicas, 47
- subjetividad, 14
- tipos, 4
  - autenticidad, 4
  - confidencialidad, 4
  - disponibilidad, 4
  - integridad, 4
- total, 14

- trabajo interorganizacional, 125–126
- verificable, 14
- vulnerabilidades, 5–6
  - almacenamiento, 5
  - comunicaciones, 6
  - emanaciones, 6
  - físicas, 5
  - hardware, 5
  - humanas, 6
  - naturales, 5
  - software, 5
- seguridad multinivel, 22
- seguridad nacional, 13
- seguridad.html, xvi
- seminarios
  - seguridad y criptografía, 80–81
- sendmail, 145
- servicios de seguridad, 113
  - boletín de supercómputo, 117
  - cert-advisory, 114
  - consultorías, 122
  - difusión de información, 113–120
    - importancia, 113
  - foros externos, 117–120
  - FTP anónimo, 114–116
  - gasu, 113
  - importancia, 113
  - necesidades, 122
  - revisión con NCARP, 121–122
  - revisión con SATAN, 120–121
  - revisión y consultoría, 120–122
  - World Wide Web, 116–117
    - para difusión, 116
    - para recolección, 117
- Servidor de Nombres, 131
- SGI, 30
- SGI, 35, 36
- shadow, 140
- Siemens, 141
- SIGSAC, 75, 79
- Silicon Graphics, véase SGI
- sirio, 54, 60
- sistema de cómputo
  - administradores, 3, 5, 6, 8–10, 12
  - mantenerlo funcionando, 4
  - usuarios, 3–13, 16, 17
- sistema operativo
  - reinstalación, 41, 53–55
- Sistemas de cómputo
  - 4D/35, 36
  - 4D/420VGX, 36
  - 9000/730, 36
  - A, 37
  - A12, 30–32
  - A9, 30
  - aldebaran, 36, 60
  - altair, 36
  - andromeda, 36, 55, 60
  - B, 37
  - Burroughs, 24, 25
  - capella, 36, 55, 60
  - casiopea, 36, 60
  - castor, 33
  - Cray, xiii, 28, 30, 32–34, 36, 40, 41, 53, 58–60, 66, 72, 127
  - cygnus, 36
  - deneb, 36, 60
  - diphda, 36, 60
  - ds5000, xv, 35, 55, 59–61, 131, 134, 138
  - IBM-650, 28
  - Indigo 2, 36
  - Indy, 35, 36
  - Macintosh, 15, 28, 73
  - mezcal, 35, 60, 131, 134, 138
  - mira, 36, 55, 60
  - Multics, 22
  - nocdos, 36, 55, 60
  - operación interna, 11
  - parámetros de seguridad, 13
  - PDP-11/45, 22
  - PDP-11/70, 22
  - pegasus, 36, 60
  - personales, 15
  - polaris, 36, 55, 60
  - pollux, 33
  - pulque, 35, 60
  - roxanne, 40
  - sirio, 54, 60
  - Sparc 1+, 36
  - SparcClassic, 35, 36
  - Sun, 59

- tepache*, 35
- tequila*, 35, 60
- totalmente seguros, 14
- tzetzal*, 32
- Unisys, 25, 30–32, 37, 38
- Unisys A12, 31
- Unisys A9, 31
- Wang, 25
- xtabentun*, 35, 60
- Y-MP4/464, 34, 53
- Sistemas operativos
  - ACF/2/MVS, 15
  - AIX, 72
  - CMW+, 15
  - HP-UX, 30, 36, 72
  - Irix, 30, 35, 36, 72
  - LAN Manager, 28, 39
  - Lantastic, 28
  - Linux, 30, 72
  - MCP, 30
  - MPE V/E, 15
  - MS-DOS, 15, 32, 39, 73
  - Multics, 15
  - MVS, 30
  - MVS/ESA, 15
  - MVS/RACF, 15
  - NetWare, 28, 39
  - NEXTSTEP, 72
  - NOS/VE, 30
  - OS 1100, 15
  - OSF/1, 30
  - SCO Unix, 30, 72
  - SCOMP, 15
  - SNS, 15
  - Solaris, 30, 35, 36, 59, 72
  - SunOS, 30, 35, 36, 72
  - SVS/OS CAP 1.0, 15
  - System 7, 15, 73
  - System V Release 3, 140
  - System V/MLS, 15
  - Trusted UNICOS, 15
  - Trusted XENIX, 15
  - Ultrix, vii, 30, 35, 72, 153, 156–158, 160–162
  - UNICOS, 30, 34, 41, 59, 72, 83
  - Unix, xi, xiii–xv, 1, 10–12, 15, 28–30, 32, 38–40, 44, 47, 49, 53–59, 61–74, 85, 90, 92, 94, 100, 106, 110, 112, 113, 117, 120
  - UTX/32S, 15
  - VAX/VMS, 15
  - VMS, 30
  - Windows, 32, 39
  - Windows for Workgroups, 73
  - Windows NT, 28, 39, 73
  - XTS-200, 15
- sistemas operativos
  - control de acceso, 8
- Smalltalk**, lenguaje de programación, 106
- SmartList, 129
- Snefru, 141
- sniffer, 41, 54
- SNS, 15
- software*, 3
  - errores, 4
  - recursos de, 4
- Solaris, 30, 35, 36, 59, 72
- sonidos, 48
- Soriano Ramírez, Ma. Susana, 80
- sort**, comando, 69
- Spafford, Gene, 14, 82, 144, 185
- Spar, 143
- Sparc 1+, 36
- SparcClassic, 35, 36
- Spring Joint Computer Conference*, 21
- sra, 141
- SRI International, 185
- Stoll, Clifford, 13
- su**, comando, 40, 102
- sub-passwords, 25
- suerte, 7
- SUID, 149
- Sun, 30, 35, 36, 59, 81
- SunOS, 30, 35, 36, 72
- supercómputo, 10–12
  - atractivo, 10
  - costo, 10
  - definición, 10
  - en la UNAM, 10
  - plan de becarios, 83

- restricciones de acceso, 10
- resultados, 12
- seguridad, 10, 12
- usuarios en la UNAM, 12
- supercómputo en la UNAM, 33–34
  - conexión a la red, 33–34
  - falta de un firewall, 33
  - ubicación, 33
  - y visualización, 33
- supercomputadora, xiii
- Susana, xvi
- SVS/OS CAP 1.0, 15
- Swatch, 143
- sys, cuenta en Unix, 148
- sys.h, 158, 159
- syslog, comando, 112, 148, 156
- syslog.conf, 157, 161
- syslog.h, 161
- syslog.o, 161, 162
- System 7, 15, 73
- System V Release 3, 140
- System V/MLS, 15
- SYSTYPE, variable, 159
  
- talk**, comando, 55, 155
- Tapia Recillas, Horacio, 80
- tar**, comando, 134
- tarjetas inteligentes, 8
- tarjetas perforadas, 19
- TCP, 141, 142
- TCP-Wrapper, vii, 57, 60, 64, 69, 73, 99, 112, 124, 142, 148, 151, 155–158, 177, 195
- TCP/IP, iv, 28, 32, 69
- tcp\\_wrapper, 156
- tcpdump, 145
- tecnología
  - control de, 19
- tecnologías
  - de protección de la información, 19
- teléfono, 8
- telecomunicaciones, 3, 4
  - líneas, 6
- telnet**, comando, 55, 106, 141, 142, 155, 167
- telnetd**, comando, 141, 156
  
- temblores, 6
- temperatura
  - cambios de, 5
- TEMPEST, 21
- tepache, 35
- tequila, 35, 60
- terminales de video, 20
- terremotos, 5, 6, 13
- testing, variable, 91
- TFTP, 149, 154
- The World Wide Web consortium, 192
- tiempo, 13
- tierra, planeta, 20
- Tiger, 57, 144
- tiger teams, 22
- titanio, 14
- TMP\\_FILE, 96, 97
- TMP\\_FILE2, 96, 97
- TMP\\_RPT, 96, 97
- Token Ring, 28
- Torvalds, Linus, 30
- Traceroute, 145
- tragedias naturales, 5
- transmisión
  - preocupación, 8
- TripWire, 57–60, 70, 99, 144, 177
- Trusted Information Systems, 15
- Trusted UNICOS, 15
- Trusted XENIX, 15
- tzetzal, 32
  
- U. S. Communications Security Board, 21
- U.S.A. Department of Defense, 185
- UAM, 70, 76
- UCLA, 22
- ucdb, 59
- UDP, 141, 142
- UID, 159
- Ultrix, vii, 30, 35, 72, 153, 156–158, 160–162
- umask**, comando, 150
- UNAM
  - sistemas de cómputo, 28–32
- UNAM, iv, v, viii, xi, xiii, xiv, xvii, 9, 12, 24, 25, 27–34, 37–41, 43,

- 45–47, 49, 53, 55, 62–70, 73, 74, 76, 77, 80, 81, 85, 86, 113, 114, 117, 120, 121, 124–128, 138, 147, 152, 183
- unam-admin*, lista de correo electrónico, 67
- UNICOS, 30, 34, 41, 59, 72, 83
- Unidad de Servicios de Cómputo Académico, Facultad de Ingeniería, vi, 124
- Unisys, 15, 25, 30–32, 37, 38
- Unisys A12, 31
- Unisys A9, 31
- Universal Resource Locator, 138
- Universidad de Washington, 145
- Universidad de Washington en San Luis, 133
- Universidad Iberoamericana, 81
- Universidad Nacional Autónoma de México, xiii
- Unix
- condición inusual, 91
  - en la UNAM, 29
  - utilización creciente, 10
- Unix, xi, xiii–xv, 1, 10–12, 15, 28–30, 32, 38–40, 44, 47, 49, 53–59, 61–74, 85, 90, 92, 94, 100, 106, 110, 112, 113, 117, 120
- ambiente de diseño, 29
  - amplitud, 56
  - como estándar, 29
  - consideración original de la seguridad, 29
  - crecimiento, 29
  - diversidad, 29
  - estándares, 29
  - filosofía, 29
  - interacciones inesperadas, 29
  - modificaciones arbitrarias, 29
  - nacimiento, 29
  - responsabilidad de los fabricantes, 29
  - versiones usadas en la UNAM, 29
- URL, iii, xv, xvi, 116, 138
- cómo interpretarlos, xv–xvi
- USCAFI, vi, 124
- usuarios, 3–13, 16, 17, 19, 20, 23, 44–45
- aceptación de la seguridad, 9
  - ambiente de trabajo, 13
  - autenticación, 11
  - autorizados, 20
  - capacidad de ignorar la seguridad, 10
  - conciencia, 9
  - conocimiento sobre seguridad, 9
  - educación, 9
  - importancia de tener conciencia, 10
  - malintencionados, 9
  - privilegios, 56
- UTX/32S, 15
- UUCP, 149
- uudecode**, comando, 154
- Vázquez, David, xvii, 67
- Variables
- archivo*, 181
  - argumento*, 179
  - DBMLIB*, 158, 159
  - DEFPWFILE*, 159
  - dominio*, 181
  - fin*, 181
  - inicio*, 181
  - LG\_OUTDEF*, 159
  - módulo*, 179
  - maquina*, 181
  - PF\_PLATE*, 160
  - PF\_TEMP*, 160
  - PWTESTFILE*, 159
  - ROOTID*, 159, 160
  - SECURE*, 155
  - SECURE\_USERS*, 155
  - SYSTYPE*, 159
  - testing*, 91
  - YY[MM[dd[hh[mm]]]]*, 181
- VAX/VMS, 15
- Vega Hernández, Gerardo, 80
- Venema, Wietse, 82, 112, 142
- ventaja competitiva, 13
- vigilantes, 5
- visualizadores de WWW, 49
- VMS, 30

- vulnerabilidades, 5–6
  - almacenamiento, 5
  - comunicaciones, 6
  - emanaciones, 6
  - físicas, 5
  - hardware, 5
  - humanas, 6
  - naturales, 5
  - software, 5
  
- W3, 48
- Walden, John S., 141
- Wang, 15, 25
- Ware, Willis H., 21
- Watcher, 144
- Windows, 32, 39
- Windows for Workgroups, 73
- Windows NT, 28, 39, 73
- World Wide Web, *véase* WWW
- write**, comando, 40
- wu-ftpd, ix, 133, 135, 145
- WWW, vi, ix, xvi, 35, 48–49, 69, 71, 76, 109, 112, 116–119, 134, 137, 138, 192, 193
  - visualizadores, 49
  
- x.y, 174, 178
- Xinetd, 57, 142
- xtabentun, 35, 60
- XTS-200, 15
  
- Y-MP4/464, 34, 53
- YP, 185
- YY[MM[dd[hh[mm]]]], variable, 181
- yyyy, 97
  
- Zamboni, Diego, xiii, 69, 80, 85, 86, 88, 93, 98, 116, 117, 120, 121, 132, 150, 151, 186
- Zimmermann, Philip, 121, 140