



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

---

FACULTAD DE CIENCIAS

INTROMISIÓN 2008

# REPORTE DE SERVICIO SOCIAL

QUE PARA OBTENER EL TÍTULO DE:  
LICENCIADO EN CIENCIAS DE LA COMPUTACIÓN

PRESENTA:  
PATRICIO LÓPEZ SERRANO ERICKSON

TUTOR:  
FRANCISCO LORENZO SOLSONA CRUZ



2009

1. Datos del alumno. Apellido paterno: Apellido materno: Nombre(s): Teléfono: Universidad: Facultad o escuela: Carrera: No. de cuenta:	López Serrano Erickson Patricio 5659 1132 Universidad Nacional Autónoma de México Facultad de Ciencias Ciencias de la Computación 4 0505869 9
2. Datos del tutor. Grado: Nombre(s): Apellido paterno: Apellido materno:	Lic. Francisco Lorenzo Solsona Cruz
3. Datos del sinodal 1. Grado: Nombre(s): Apellido paterno: Apellido materno:	Dr. Sergio Rajsbaum Gorodezky
4. Datos del sinodal 2. Grado: Nombre(s): Apellido paterno: Apellido materno:	Dr. José de Jesús Galaviz Casas
5. Datos del sinodal 3. Grado: Nombre(s): Apellido paterno: Apellido materno:	Fís. Max Ulises de Mendizábal Carrillo
6. Datos del sinodal 4. Grado: Nombre(s): Apellido paterno: Apellido materno:	Mat. Salvador López Mendoza
7. Datos del trabajo escrito. Título: Subtítulo: Número de páginas: Año:	INTROMISIÓN 2008  62 2009

# **INTROMISIÓN**

Patricio López-Serrano Erickson



# **Agradecimientos**

A mi familia, mi asesor, los sinodales, los patrocinadores del evento y todos los que hicieron posible y divertido este proyecto.



# Índice general

<b>I</b>	<b>Fundamentos</b>	<b>7</b>
1.	Cifrado de datos	11
2.	Seguridad de redes y protocolos	13
2.1.	Redes alambradas . . . . .	13
2.2.	Redes inalámbricas . . . . .	14
2.3.	Domain Name System (DNS) . . . . .	14
2.4.	Address Resolution Protocol (ARP) . . . . .	15
3.	El <i>kernel</i> de Linux	17
3.1.	Protección del espacio ejecutable . . . . .	17
3.2.	<i>Null-pointer dereference</i> . . . . .	18
3.3.	Desbordamiento de búfer ( <i>Buffer overflow</i> ) . . . . .	18
3.4.	Referencia colgante ( <i>Dangling pointer</i> ) . . . . .	18
4.	Programas para auditoría	21
4.1.	Aircrack-ng . . . . .	21
4.2.	Ettercap-ng . . . . .	22
4.3.	Nmap . . . . .	24
<b>II</b>	<b>Análisis y discusión puntual</b>	<b>27</b>
5.	Alcance del evento	29
5.1.	Omisiones . . . . .	29
5.2.	Intromisiones . . . . .	30
6.	Estructura administrativa	31

<b>7. Conferencias</b>	<b>33</b>
7.1. Corrección de protocolos de seguridad . . . . .	33
7.2. Construcción de un plan estratégico de seguridad de la información	34
7.3. La criptografía nuestra de cada día . . . . .	36
7.4. Linux firewall en 10 líneas . . . . .	37
<b>8. Talleres</b>	<b>41</b>
8.1. Administración segura de sistemas Linux . . . . .	41
8.2. Configuración básica de un servidor de nombres (DNS) . . . . .	42
8.3. Uso de Wireshark . . . . .	43
8.4. Uso de ettercap . . . . .	44
8.5. Aircrack-ng . . . . .	45
<b>9. El reto de penetración</b>	<b>47</b>
<b>III Reflexiones</b>	<b>51</b>
<b>10. Notas finales</b>	<b>53</b>
10.1. Logros . . . . .	53
10.2. Metas . . . . .	54




**Parte I**

**Fundamentos**



# Introducción

a necesidad de divulgar las prácticas actuales en el campo de la seguridad informática no está a discusión. Lo verdaderamente interesante comienza con los métodos existentes para incrementar la conciencia sobre éstas. Algunas posibilidades:

- Ser víctima de una intrusión, obligando a tomar medidas para evitar futuros incidentes. Tiene por ventaja la efectividad, a costa del ego.
- Aleccionamiento teórico: tedioso, pero útil para involucrarse en conversaciones con profesionales del ramo.
- Participación directa en demostraciones de seguridad ofensiva (en un ambiente controlado y completamente legal): divertido, didáctico e inolvidable.

INTROMISIÓN se trata justamente de la última opción. Como proyecto, pretende divulgar echando mano de la participación en lo que (de ser otras las circunstancias) sería perfectamente inapropiado. Se inspiró, hasta cierto punto, en un evento anual de seguridad ofensiva llamado DEF CON, donde se reúnen talentos de todo el mundo para exponer sus hallazgos más recientes y participar en una multitud de retos y pruebas de dudosa legitimidad, todo en un ambiente lúdico y altamente competitivo.

Para formular un evento completo, se decidió incluir teoría, práctica y concurso: conferencias, talleres y un reto de penetración. Se buscó tratar temas que decrementaran la sensación de seguridad y confianza de los participantes, fomentando la inquietud por fortalecer sus hábitos de protección de manera distinta a la usual.

Este reporte describe, justifica y analiza lo ocurrido mediante tres partes: los fundamentos teóricos requeridos para la comprensión de los aspectos prácticos subsecuentes a ellos; el análisis y discusión de las ponencias y el reto incluidos en el evento, y las reflexiones finales sobre lo discutido a lo largo del escrito.



## 1. Cifrado de datos

---

Como recurso para mantener la privacidad y confidencialidad de la información sensible hay múltiples implementaciones de sistemas y protocolos que utilizan la criptografía. Dichos medios existen en varios niveles de la comunicación electrónica, atendiendo a cada uno de los puntos donde puede haber participación no autorizada en el intercambio. Para tener una imagen más clara, recuérdese el modelo TCP/IP:

Nivel	Nombre de la capa
4	Aplicación
3	Transporte
2	Internet
1	Red local

A continuación se enumeran brevemente algunos ejemplos de cifrado, y cómo se utilizan en su respectiva capa:


- **Cifrado de enlace (*Link encryption*)**. [22] En este esquema se cifran y descifran todos los datos en cada nodo de un enlace de comunicaciones.
- **Cifrado de capa de transporte (*Transport Layer Security*)**. [16] TLS, el sucesor de SSL (Secure Sockets Layer), proporciona cifrado de conexiones que requieren saltos por más de un ruteador (a diferencia de la capa de enlace). Está diseñado para permitir intercambios libres de intervención y falsificación. Una de sus aplicaciones más comunes es la autenticación unilateral: el cliente verifica la autenticidad del servidor, sin que el servidor verifique la autenticidad del cliente (piénsese en lo que ocurre con la banca en línea, por ejemplo). TLS involucra tres fases:

1. Negociación del algoritmo de cifrado (las opciones en implementaciones actuales son RSA, DIFFIE-HELLMAN, ECDH, SRP O PSK)
  2. Intercambio de llaves y autenticación (utilizando RSA, DSA O ECDSA)
  3. Cifrado simétrico del tráfico (mediante RC4, TRIPLE DES, AES, IDEA, DES O CAMELLIA)
- **Cifrado extremo a extremo o punto a punto (End-to-End Encryption).** [5] [2] Puede ocurrir tanto en la capa de aplicación (en PGP y SMIME, por mencionar algunos) como en la de red local (utilizando soluciones de cifrado en *hardware*). Bajo este esquema se cifra el tráfico en el origen para ser descifrado *sólo* hasta llegar al destinatario. A diferencia del cifrado de enlace, los nodos intermediarios no descifran el contenido del mensaje. Presenta algunas ventajas, a saber:
- Reduce la latencia mediante la selectividad. A diferencia de HTTPS (HTTP sobre SSL), que cifra todo el contenido de una página siendo que algunos elementos podrían no requerirlo (imágenes, código HTML, etc.), el cifrado punto a punto elige sólo los elementos necesarios.
  - Ahorro monetario. La vía más común de autenticación es adquiriendo certificados SSL de una entidad emisora. Este esquema de cifrado elimina dicha necesidad, dado que los únicos capaces de descifrar los mensajes son el emisor y el receptor.
  - Cambio flexible de llaves. La renovación periódica de llaves públicas y privadas es una excelente práctica para elevar la seguridad de la comunicación. El cifrado extremo a extremo permite hacerlo con la frecuencia deseada, a diferencia, nuevamente, de la emisión de certificados estáticos.

La mayoría de los mecanismos descritos arriba son transparentes para el usuario (es decir, no es notoria su actuación en las transacciones cotidianas), lo cual no reduce su relevancia. Las únicas restricciones a su empleo deben ser las determinadas por la importancia de la información que protegen y los recursos disponibles para su implementación (por ejemplo, no muchos usuarios caseros podrán acceder al cifrado de paquetes en *hardware* para su red local).

## 2. Seguridad de redes y protocolos

---

e vio en el capítulo anterior que el cifrado es una herramienta poderosa y necesaria para mantener en secreto la comunicación. Desafortunadamente, no es suficiente; mientras los elementos que subyacen a la criptografía sean débiles, ésta no puede conservar su propiedad de eficacia. En lo sucesivo se analizan algunos componentes encargados de sostener las conexiones cifradas, y que si son vulnerados, hacen fútiles a éstas.

### 2.1. Redes alambradas

El principal caso de estudio para esta categoría son las redes locales (LAN). Están expuestas a intervención física, por lo que deben implementarse medidas de seguridad perimetral para prevenir este tipo de ataques. Una de dichas medidas (tal vez la más costosa) es la inserción de la instalación en tubos con gas presurizado; cualquier atentado a la integridad del tubo causa una variación de presión (la cual se monitorea constantemente), disparando una alarma. [22] Además de los riesgos físicos, existen algunos problemas de implementación, explotables por cualquier equipo autorizado para pertenecer a la red:

- La difusión de mensajes (*broadcast*). Es la debilidad más notoria, dado que cada paquete transmitido es potencialmente accesible por cualquier equipo en el mismo segmento. Esto implica pérdidas de autenticidad y privacidad, constituyendo un riesgo muy severo. La única solución física al problema es utilizar un *switch*, que además de segregar los segmentos de una red, impide el espionaje a causa de la difusión y ayuda a aliviar la congestión de paquetes. [6]

## 2.2. Redes inalámbricas

La gran practicidad en la instalación y uso de las redes inalámbricas las hace una opción muy atractiva para el manejo de información. Los intrusos también creen que son fabulosas, sobre todo por las siguientes razones:

- No es físicamente evidente cuando se está conectado a una red inalámbrica. ¡Aún mejor! - se puede estar conectado a una red inalámbrica desde fuera de un edificio, en un lugar completamente inconspicuo. Las redes inalámbricas se propagan mediante ondas de radio y modificar su trayectoria es prácticamente imposible. Lo único que se puede hacer es limitar la potencia de transmisión del punto de acceso, de tal modo que la señal sea débil fuera de cierto perímetro.
- El protocolo de seguridad predominante, WEP, es sorprendentemente débil. Reside en la capa de enlace de datos y emplea un cifrado de flujo basado en RC4. La lógica del cifrado es muy sencilla: se operan mediante XOR el texto plano y el flujo de llaves. La seguridad de RC4 requiere mantener en secreto la llave de paquete derivada del flujo. Dicha llave de paquete consta de lo siguiente: [8] [17] [23]
  - Una llave previamente compartida (*Pre-shared Key*). La llave de cifrado utilizada por los usuarios partícipes de la comunicación privada (i.e., el punto de acceso y la tarjeta inalámbrica del usuario).
  - Un arreglo de estados. Durante el proceso de cifrado, se coloca un conjunto de valores (generalmente de 1 a 256) en un arreglo de estados, que después se revuelve y contribuye a la construcción del flujo final.
  - Vector de inicialización (*IV*). Número aleatorio de 3 bytes colocado al principio o al final del texto cifrado, para enviarse al destinatario.

## 2.3. Domain Name System (DNS) <sup>1</sup>

El DNS es vital para Internet al proveer el mecanismo que hace conversiones entre nombres mnemónicos (*hostnames*) y direcciones IP. La inseguridad de algunos de los protocolos que lo componen, además de la falta de autenticación y pruebas de integridad de la información inherente amenazan el funcionamiento de dicho sistema. Algunas de las principales vulnerabilidades de DNS son:

---

<sup>1</sup>Davidowicz, D. Domain Name System (DNS) Security.  
<http://compsec101.antibozo.net/papers/dnssec/dnssec.html>.



- **Envenenamiento de caché (*cache poisoning*)**. En el caso que un servidor DNS no posea en su caché la respuesta a una consulta, éste puede referirla a otro servidor, en nombre del cliente. Si el servidor tiene entradas incorrectas en su caché (colocadas de manera intencional o no), se dice que su caché esta *envenenado*. El envenenamiento malicioso también se conoce como *DNS spoofing* (falsificación DNS).
- **Servidores piratas**. Representan un riesgo para la comunidad de Internet al contener entradas espurias. Un usuario malicioso puede montar su servidor DNS pirata y dirigir el tráfico de las peticiones de usuario a réplicas de sitios comunes (correo electrónico, banca en línea, etc.) y recopilar información sensible de las víctimas, sin que éstas siquiera sospechen lo que ocurre.
- **Enmascaramiento**. Este ataque pretende acreditar como entidades confiables y válidas a aquellas que no necesariamente lo son. Las consecuencias son interceptación, análisis y corrupción de la comunicación entre la víctima y el sitio con el que cree estar tratando.

## 2.4. Address Resolution Protocol (ARP)

Cada dispositivo conectado a una red tiene por lo menos dos direcciones: una del controlador de acceso al medio (MAC) y una del protocolo de internet (IP). La dirección MAC corresponde a la interfaz física de red del dispositivo y no cambia a lo largo de la vida del mismo (a menos que el usuario conozca las técnicas de falsificación adecuadas). La dirección IP, no obstante, puede cambiar al moverse el dispositivo dentro de la red, o si la red implementa el protocolo DHCP, por ejemplo. ARP, uno de los protocolos de IP, se utiliza para convertir entre direcciones MAC e IP. Funciona difundiendo un paquete con la dirección IP del destinatario, que todos los dispositivos *deberían* ignorar, exceptuando al interesado. Además de esto, cada dispositivo posee una *tabla ARP*, que almacena las direcciones MAC e IP que ya ha relacionado en su red, con el fin de no repetir peticiones en vano. [11] En lo siguiente se listan los ataques más comunes al protocolo ARP: [21]

- **Negación del servicio (*Denial of Service*)**. Un usuario malicioso podría asociar cualquier dirección MAC bien formada con una dirección IP arbitraria - por ejemplo, la dirección IP del ruteador de la red con una MAC falsa. El resto de los dispositivos creen estar intercambiando datos con el enlace, cuando en realidad, todo se está yendo a la basura.
- **Espionaje (*Man in the Middle*)**. El intruso intercepta el tráfico entre dos dispositivos. Lo logra haciéndole creer al ruteador que su dispositivo es el de la

víctima mediante una “respuesta” ARP (que el ruteador ni siquiera solicitó), con lo cual el tráfico será redirigido. Para terminar el ataque, deberá cambiar la tabla ARP de la víctima, haciendo que dirija el tráfico al dispositivo atacante, creyendo que es el ruteador. Nótese que se puede hacer que *todos* los dispositivos de una red estén intervenidos simultáneamente de esta forma.

- **Inundación** (*Flooding*). Este ataque está dirigido específicamente a los *switches* de red. Como se vio en la sección de redes alambradas, los *switches* hacen privada la comunicación con cada dispositivo, aunque algunos de ellos pierden esta capacidad cuando el tráfico de paquetes es excesivo (al estar demasiado ocupado con la integridad de los datos, en vez de las políticas de exclusión), posibilitando el espionaje.


Como solución a estos problemas se pueden implementar medidas adicionales de seguridad, tales como: [21]

- Direcciones IP fijas para cada dispositivo.
- Tablas ARP fijas para cada dispositivo, establecidas por la administración.
- Control de exclusión de direcciones MAC para el ingreso a la red.
- Herramientas de monitoreo de ARP, como ARPwatch.

Los dos tipos de redes que se enumeraron en este capítulo, además de los protocolos mencionados, pertenecen a la experiencia común de cualquier usuario de computadoras (aunque el nivel de conciencia que se tiene de ellos puede ser distinto). Con ello, debe quedar claro el gran número de oportunidades fáciles que existen para llevar a cabo ataques contra la privacidad, confidencialidad e integridad de la información en cuestión.

## 3. Aspectos de (in)seguridad en el *kernel* de Linux

---

n el cumplimiento de las políticas de seguridad, el *kernel* de un sistema operativo es el último delegado en la cadena de protección. Precisamente esta cualidad lo hace un blanco muy perseguido por los ataques, además de sujeto de revisiones y auditorías exhaustivas de código. Al estar encargado de las funciones de más bajo nivel (y, principalmente, el manejo de la memoria), hay una búsqueda constante de maneras nuevas para engañarlo y hacer que lleve a cabo diversas acciones que normalmente serían indeseables. En lo siguiente se exponen las clases generales de vulnerabilidades y ataques atingencia con el *kernel* de Linux, teniendo en cuenta que son consecuentes para la seguridad en la medida en que las pueda explotar un atacante habilidoso. En el caso de la ejecución de código arbitrario (el efecto más temido de implementaciones incorrectas), es necesario que el atacante introduzca una carga útil (*payload*, en inglés) a la región vulnerada para poder proceder con el ataque; en otras palabras, no ocurre por sí solo.

### 3.1. Protección del espacio ejecutable

En términos generales, esta técnica consiste en etiquetar ciertas regiones de la memoria como *no ejecutables*, de modo que cualquier intento de ejecutar código de máquina en ellas causará una excepción. Típicamente utiliza facilidades de *hardware* para funcionar así (el bit NX), aunque existen maneras de hacer que el sistema operativo se encargue del etiquetado. El *kernel* de Linux es capaz de utilizar el bit NX en procesadores de las arquitecturas x86-64 y x86 que lo

soporten [7]. Con esto se pretende limitar las regiones de memoria susceptibles a ataques como los que se describen aquí.

### 3.2. *Null-pointer dereference*

Ocurre cuando un apuntador nulo se utiliza como si apuntara a una dirección válida de memoria. La consecuencia más usual es la terminación anormal del proceso, aunque en algunas ocasiones puede conducir a la ejecución de código arbitrario. [12] La única manera de evitar este tipo de ataque es asegurarse que no haya cabida para él en el código de un programa. Un ejemplo en el lenguaje C: [3]

```

1   if (pointer1 != NULL) {
2       /* utilizar pointer1 */
3       /* ... */
4   }
```

Al liberar apuntadores, asegurarse que no apunten a NULL; una vez liberados, verificar que sí lo hagan:

```

1   if (pointer1 != NULL) {
2       free(pointer1);
3       pointer1 = NULL;
4   }
```

### 3.3. Desbordamiento de búfer (*Buffer overflow*)

Esta falla de seguridad acontece cuando un proceso almacena datos en un búfer **fuera** de la memoria que se le asignó originalmente. Los datos sobrantes sobrescriben la memoria adyacente (que puede o no contener otros datos), ocasionando comportamiento errático, errores de acceso a la memoria, terminación del proceso y ejecución de código arbitrario. [1] [14]

### 3.4. Referencias colgante (*Dangling pointer*)

Las referencias colgantes son apuntadores que no apuntan a un objeto válido del tipo apropiado. Nótese el uso general de la palabra “objeto”, implicando que este tipo de falla no es exclusiva de los lenguajes orientados a objetos. Las referencias colgantes surgen al eliminar o quitar la asignación a un objeto sin modificar el valor de su apuntador (dejándolo *colgando*, haciendo referencia a una dirección de memoria en desuso). Este problema incumbe a la seguridad en cuanto se

puede utilizar una referencia colgante para llamar a una función virtual o una dirección de memoria distinta (que puede contener código malicioso). Nuevamente, la solución a este problema es asegurarse que los apuntadores en desuso hagan referencia a NULL. [4]



## 4. Programas para auditoría

---

Quizá lo más interesante de las herramientas para auditoría de seguridad es su cualidad de arma de doble filo. Los programas más comunes son parte habitual de la primera línea de búsqueda del atacante experto (y única parte de la búsqueda del atacante oportunista). Del otro lado, quien tiene a su cargo la seguridad de los sistemas puede utilizar dichas herramientas para verificar el comportamiento de sus medidas e implementaciones de fortificación ante indagaciones de ese tipo. Además de proveer certeza sobre la resistencia a ataques genéricos, los programas aquí listados son capaces de mostrar aspectos más complejos si se configuran para ello (llegando así al tipo de fallas que podría intentar explotar un usuario versado), haciéndolos parte elemental de los recursos que debe tener a la mano cualquier persona preocupada por la seguridad.

### 4.1. Aircrack-ng <sup>1</sup>

**A**ircrack-ng es un conjunto de herramientas que se utiliza para probar la fortaleza del cifrado en redes inalámbricas. Es capaz de obtener llaves (en la mayoría de los casos) para redes que funcionan bajo WEP, WPA y WPA2. Para comprender su funcionamiento, recuérdese el esquema de cifrado WEP (sección 3.2), que hace uso de vectores de inicialización prefijados a cada paquete. Ahí reside el recurso para la explotación que hace aircrack-ng, como se verá a continuación, a través de cada componente:

1. **Airmon-ng**. Este programa habilita o deshabilita el modo monitor o *promiscuo* de la interfaz de red inalámbrica con la cual se realizará el ataque. Es

---

<sup>1</sup>Aircrack-ng - main documentation. <http://www.aircrack-ng.org/documentation.html>.

indispensable para el resto del procedimiento, ya que permite la lectura de todos los paquetes que circulan en la red, y no sólo los que están destinados a la misma.

2. **Aireplay-ng**. Si se desea acelerar el proceso de recolección de tráfico (teniendo en mente que el objetivo es adquirir la mayor cantidad de vectores de inicialización en el menor tiempo posible), es necesario *inyectar* paquetes al flujo. Si la red inalámbrica que se está intentando explotar tiene poco tráfico, la recolección de vectores puede ser extremadamente tardada; **Aireplay-ng** contribuye a reducir el espacio de búsqueda de la llave generando tráfico “repetido”, obligando al punto de acceso a revelar más vectores de inicialización. Nótese que no todas las tarjetas inalámbricas son capaces de inyectar paquetes - sólo aquellas con controladores físicos que permiten dicha tarea.
3. **Airodump-ng**. Similar a la herramienta `tcpdump`, es el programa que captura los paquetes generados por **Aireplay-ng** (en particular, los vectores de inicialización para WEP). Guarda el volcado en archivos para su posterior decodificación, incluyendo detalles sobre el punto de acceso y otros clientes asociados a él, si es que los hay.
4. **Aircrack-ng**. El último paso en la obtención de la llave, toma como entrada los archivos generados por **Airodump-ng** en la etapa anterior. Determina la llave WEP utilizando dos métodos. El primero es `PTW` (por los apellidos de sus autores, Pyshkin, Tews y Weinmann), cuya principal ventaja es el reducido número de paquetes que necesita para arrojar la llave. El segundo método es `FMS/KoreK` (también llamado así por sus autores); incorpora varios ataques estadísticos para descubrir la llave, utilizándolos en conjunto durante el ataque de fuerza bruta.

## 4.2. Ettercap-ng <sup>2</sup>

Ettercap es una herramienta muy versátil que permite la manipulación del tráfico de una red. Sus habilidades le permiten efectuar ataques de espionaje (*Man in the Middle*) muy fácilmente, incluso dentro de entornos *switcheados*. Su espectro de acción es tal que ambos bandos de la seguridad suelen utilizarlo: los atacantes pueden obtener información sustancial mediante su empleo y los defensores lo utilizan para descubrir tales ataques y fortalecer sus políticas de seguridad. Una vez habilitado, `ettercap-ng` puede capturar y examinar todo

---

<sup>2</sup>Norton, D. An Ettercap Primer. SANS Institute, 2004.



el tráfico que fluye entre dos víctimas cualesquiera y actuar con alguna de las siguientes funcionalidades:

- **Modificación del flujo:** se puede insertar arbitrariamente una serie de caracteres dentro de una conexión activa (en cualquiera de ambas direcciones), simulando peticiones del cliente y/o respuestas del servidor.
- **Filtrado de paquetes:** es una disección de la carga útil de los paquetes TCP ó UDP, en busca de secuencias ASCII o hexadecimales dentro de la carga útil de éstos. Se puede proceder a reemplazar o desechar los paquetes filtrados.
- **Recolección de contraseñas:** ettercap-ng es capaz de reconocer y extraer información pertinente de una multitud de protocolos, incluyendo TELNET, FTP, POP3, RLOGIN, SSH1, ICQ, SMB, MYSQL, HTTP, NNTP, X11, NAPSTER, IRC, RIP, BGP, SOCKS 5, IMAP 4, VNC, LDAP, NFS, y SNMP. La utilidad de esto es indiscutible; la facilidad, apabullante.
- **Espionaje:** siempre y cuando el cliente acepte un certificado falsificado por el atacante, ettercap-ng se puede colocar en medio de una transmisión HTTP sobre SSL.
- **Eliminación de conexiones:** se puede examinar y terminar cualquier conexión activa.

Además de las anteriores herramientas de ataque, ettercap-ng es una excelente opción para exploración y descubrimiento de topologías de red, teniendo dentro de sus posibilidades lo siguiente:

- **Identificación del sistema operativo** que está ejecutando cualquier cliente asociado a la red.
- **Análisis pasivo** del tráfico circulante, obteniendo información sobre sistemas operativos, puertos disponibles, servicios activos, direcciones IP y MAC.
- **Análisis en modo *promiscuo*:** permite almacenar tráfico que no está dirigido específicamente al atacante, de manera similar a tcpdump.
- **Búsqueda de envenenadores:** detecta la presencia de otros sistemas que estén analizando la red o llevando a cabo envenenamientos de la tabla ARP.
- **Construcción de paquetes:** se pueden elaborar paquetes Ethernet e IP a la medida, probando la respuesta de los dispositivos asociados a la red. Puede utilizarse para fijar encabezados de paquete manualmente, además de falsificar direcciones IP y MAC.

### 4.3. Nmap <sup>3</sup>.

Es una herramienta de código abierto diseñada para explorar redes y hacerles auditorías de seguridad. Es capaz de ejecutarse velozmente aún sobre segmentos poblados considerablemente. Se utiliza muy frecuentemente en una variedad de contextos (no todos ellos necesariamente relacionados con seguridad), dadas las capacidades de reconocimiento que tiene:

- **Disponibilidad.** Cualquier inmersión en la topología de una red implica querer obtener una lista concisa de dispositivos activos y/o interesantes. Tal inmersión puede quedar a cargo del administrador de la red o de un auditor de seguridad (por mencionar un par de ejemplos); cada uno de ellos tendrá una versión distinta de lo que es *interesante*. Con esto queda claro que se requiere una gran diversidad de métodos para descubrir dispositivos activos en una red - Nmap los posee. Si no se especifican opciones avanzadas, Nmap envía una solicitud de eco ICMP, un paquete TCP SYN al puerto 443, un paquete TCP ACK al puerto 80 y una solicitud de señal de tiempo (*timestamp*) para probar la existencia de los dispositivos. Entre sus opciones avanzadas de investigación están el ping UDP y ARP.
- **Servicios activos.** Una vez reconocido como activo un dispositivo en la red, es deseable saber qué servicios está ofreciendo. Para el administrador es importante saber que cada servidor está haciendo bien su trabajo y para el auditor es uno de los primeros pasos en la búsqueda de vulnerabilidades. Nmap interroga a los puertos que encuentra disponibles en el dispositivo para averiguar más sobre su funcionamiento. Lo hace mediante una base de datos con expresiones y patrones de servicios comunes. Intenta determinar el protocolo del servicio, la aplicación que lo utiliza, el número de versión, el tipo de dispositivo y la familia de sistema operativo.
- **Sistemas operativos.** Nmap es capaz de interpretar información sobre múltiples sistemas operativos con el fin de identificarlos remotamente. Envía una serie de paquetes TCP y UDP, examinando minuciosamente la respuesta. Al igual que en el descubrimiento de servicios, existe una base de datos poblada de respuestas que emparejan con cada sistema operativo disponible.
- **Evasión de firewalls.** El mapeo de una red se puede ver obstruido por medidas de seguridad como *firewalls* y sistemas de detección de intrusos (*IDS*, por sus siglas en inglés). Nmap, junto con la paciencia y experiencia

---

<sup>3</sup>Chapter 15. Nmap Reference Guide. <http://nmap.org/book/man.html>

del usuario, ofrece distintas posibilidades para entender más a fondo estas medidas, saber si están funcionando correctamente o evadirlas del todo:

- **Fragmentación** de los paquetes IP utilizados para la búsqueda. Básicamente, se distribuye el encabezado TCP a lo largo de pequeños fragmentos de paquete, dificultando la tarea que hacen los filtros, IDS y demás impedimentos.
- Otra opción es el escaneo con **señuelos**, donde la investigación parece provenir de *varios* dispositivos en la red concurrentemente (y no solamente el dispositivo que está ejecutando Nmap en realidad).
- Envío de paquetes TCP/UDP con sumas de verificación **inválidas**. Aprovechando el hecho que casi cualquier *stack* IP (bien implementado) descartará paquetes con *checksum* incorrecto, obtener una respuesta así inducida significa muy probablemente que un *firewall* o IDS no se molestó en hacer la verificación, indicando su presencia.



## **Parte II**

# **Análisis y discusión puntual**



## 5. Alcance del evento

---

**S**n la parte anterior se estudió brevemente el contenido teórico-técnico necesario para proceder al estudio de las ponencias del evento. Introducir debidamente a las exposiciones subsecuentes requiere explicar que todas ellas (conferencias, talleres y el reto, por supuesto) conforman lo que se creyó más adecuado y completo para dotar a los participantes de fundamentos teóricos, herramientas clave y motivación al abordar la seguridad de los sistemas que tienen a su mando. En este respecto, cabe destacar que el enfoque abarcó casos de uso personal y profesional, pretendiendo siempre hacer ver la cercanía e inmediatez de los riesgos. Las conferencias, por ejemplo, tuvieron un nivel elevado de conocimiento teórico que muchas veces dependía recíprocamente con lo visto en los talleres y el reto. El objeto de los talleres fue ofrecer contenido directamente aplicable por los asistentes, dado el contacto cercano con cada instructor. En el respectivo capítulo se critica el papel de cada taller con respecto al curso actual de la seguridad, justificando la importancia de lo impartido. El reto fue la oportunidad de poner en práctica algunas de las técnicas que los participantes ya conocían previamente o habían adquirido en el transcurso del evento.

### 5.1. Omisiones

Para la organización, gran parte del aprendizaje sobre este evento es a través de lo que no incluyó. Aunque es injusto con los expositores hablar sobre los aspectos que sus ponencias *pudieron haber incluido*, es necesario hacerlo a manera de ejercicio para el futuro y no como una falta de agradecimiento para con ellos. En el afán de refinar la estructura de INTROMISIÓN para hacerla acercarse a la idea que motivó su organización, aquí se plantean algunos puntos concretos que hubieran

ensalzado al evento y se tomarán como directrices para futuras ediciones.

- En el tema de seguridad perimetral, una excelente demostración para taller es obtener la contraseña de `root` utilizando un *Live CD* de GNU/LINUX y algún programa capaz de ataques por fuerza bruta (como `John the Ripper` [10]). Muestra de una forma maravillosamente efectiva la necesidad de restringir el acceso físico al equipo de cómputo mediante una variedad de disposiciones (contraseña en el BIOS, cerrar con seguro la puerta, etc.).
- Montar un servidor DNS *pirata* que dirija el tráfico de los participantes a réplicas de páginas web, almacenando su información personal de banca en línea, correo electrónico, etc.
- Involucrar en el reto a equipos de forenses y no sólo a penetradores. Su inclusión requeriría, obviamente, mayor y mejor diseño de antemano.


## 5.2. Intromisiones

Habiendo reconocido las deficiencias, aquí comienza la historia de éxito de INTROMISIÓN 2008, con todo lo que *sí* formó parte de ella.



## 6. Estructura administrativa

---

 equirió una gran labor de planeación y organización el incorporar en el evento los elementos que se consideraron necesarios. Primeramente, reuní una lista de potenciales expositores (locales e internacionales) que ofrecieran contenido interesante y útil para los asistentes. Se dividieron conceptualmente en conferencistas y talleristas, aportando información y técnicas utilizables desde el momento de su exposición. El principal objetivo del evento, si no es que su esencia, fue brindar la oportunidad de ejercitar **en vivo** las habilidades en cuestión, con la ventaja de poder consultar a los expertos sobre las interrogantes que resultaren.

Asimismo, decidí dar imagen formalmente al evento mediante un logotipo. Se eligió una tipografía sólida que representara el espíritu del evento y a Minerva (diosa griega de los guerreros y el conocimiento, entre otros tantos) por aparecer en el escudo de la Facultad de Ciencias, agregándole una expresión de sorpresa en vista del bochorno que le causaría tanto entrometimiento. Los diseños anteriores estuvieron a cargo de Santiago Russek (el diseñador gráfico que consulté) y se incluyeron en playeras y estampas, a manera de recuerdo.

En el ámbito de la difusión, se hicieron dos pósters: uno previo, muy incitador, con la leyenda “¿muy metiche?”, y otro con la información detallada sobre el evento. Ambos están incluidos en los anexos. Hice, junto con el diseñador, una página web <sup>1</sup> con toda la información sobre los conferencistas y los temas que impartieron, conformando además el portal para las solicitudes de participación en los talleres y el reto. El avance del reto y las pistas que se dieron a lo largo de éste también se incluyeron. Se videograbaron las conferencias para disponibilidad inmediata en línea de quien así lo deseara. Se enviaron correos a las listas de la

---

<sup>1</sup><http://intromision.fciencias.unam.mx>

Facultad de Ciencias, que incluso llegaron a otras instancias, como el Instituto de Astronomía.

INTROMISIÓN se llevó a cabo en instalaciones de la Facultad de Ciencias en la semana del 24 al 28 de noviembre de 2008. Los talleres se impartieron en el laboratorio de Ciencias de la Computación II del edificio Tlahuizcalpan, y las conferencias tuvieron por sede al auditorio Alberto Barajas Celis.


Además de las labores de gestión y organización en las que estuve involucrado, impartí un taller sobre ETTERCAP-NG descrito en el capítulo de talleres y diseñé el reto de penetración (cuyo capítulo tampoco se olvidó).

## 7. Conferencias

---

### 7.1. Raúl Monroy: *Corrección de protocolos de seguridad*

*Raúl Monroy es Doctor en Inteligencia Artificial por la Universidad de Edimburgo (1998). Es Profesor Asociado en el ITESM Campus Estado de México y miembro de CONACYT-SNI. Su investigación está enfocada en la automatización de la demostración de teoremas y métodos formales de desarrollo de sistemas. Le interesa la seguridad informática, y en particular, la aplicación de estrategias generales de control de búsqueda para descubrir y corregir errores en un sistema o su especificación, y hallar métodos nuevos para detectar anomalías en la seguridad.*

 n un giro distinto dentro del enfoque ofensivo que se deseó para INTRO-MISIÓN 2008, esta conferencia ofreció contenido de valor teórico útil para la búsqueda de vulnerabilidades en protocolos de todo tipo. Aunque no son *directamente* relativas a una aplicación o implementación (i.e., programa), las técnicas descritas atañen a los protocolos más usuales (incluidos, por supuesto, los ilustrados en este trabajo), mostrando su atractivo para auditorías teóricas. Cabe resaltar los siguientes puntos por su conexión con otros aspectos discutidos tanto en el evento como en este escrito:

- **Importancia del formato.** Se estableció que los protocolos poseen formatos de mensaje asociados a ellos (forman parte de su definición o diseño). Uno de los elementos que hay que tomar en cuenta durante el análisis de un protocolo en busca de sus fallas es justamente el formato. Tan exacerbado es, que la explotación del protocolo WEP discutida en la sección 5.1 (más específicamente, el programa Aireplay-ng) echa mano de un ataque tipificado como *replay attack*. Consiste en repetir o retrasar fraudulentamente

el contenido de una transmisión, aprovechando la falta de revisión de formato por parte del protocolo. Una manera de evitarlo es utilizar control de sesiones con *timestamps* o cualquier valor de uso único (i.e., desechable). [13]

- Dos **enfoques** sobre la robustez. La visión predominante en la seguridad está relacionada con la *complejidad* y pretende responder a la pregunta “¿Qué tan fácil es romper un sistema?”. Constituye el respaldo de la confianza que se tiene en las funciones *puerta de trampa*, por ejemplo. En el estudio de protocolos, no obstante, la pregunta cambia a “¿Qué se puede aprender presenciando la ejecución de un protocolo?”. Creo muy importante esta pregunta porque independientemente de la complejidad computacional de un sistema, una ejecución particular (*patológica*) del protocolo con el que funciona puede divulgar secretos, arruinándolo todo.
- **Herramientas** disponibles. La corrección de protocolos está relacionada con la rama de métodos formales, de manera que existen proyectos como AVISPA (*Automated Validation of Internet Security Protocols and Applications* [15]) que contribuyen a la depuración de protocolos complejos, en todas las fases de su formulación.
- **Principios de diseño**. La labor de diseño de protocolos no es ciega; existen principios de diseño que asisten en la obtención de propiedades deseables como simplicidad, modularidad, consistencia y robustez. También son útiles para la búsqueda de errores en los protocolos. [18]
- **Compromisos** en la reparación de protocolos. El ubicuo concepto del sacrificio computacional se hace patente aquí también. Interesante en demasía resulta el que la reparación de un protocolo pueda convertirlo en otro completamente, si ésta va en contra de las intenciones originales con que se diseñó.

## 7.2. José García Sabbagh: Construcción de un plan estratégico de seguridad de la información

*José García Sabbagh es Director de Consultoría para Smart Security Services. Dentro de sus funciones está la coordinación y dirección de un grupo de personas que se desenvuelven como Oficiales de Seguridad de la Información dentro de empresas importantes en México, en las cuales Smart Security es responsable de la seguridad de la información. Ha ocupado algunos puestos dentro de empresas financieras y de manufactura, dentro de las que destacan*

*el de Gerente de Administración de Riesgos de Información para ING México, y la posición de Arquitecto de Seguridad para Grupo Modelo. José posee algunos certificados internacionales de Seguridad de la Información y TI (CISSP, CISM, FC-ITSM, BSI-LA). Estudia una maestría en Administración de Tecnologías de Información en el Tecnológico de Monterrey. Posee una licenciatura en Administración de Sistemas de Cómputo, tiene un posgrado en Seguridad de la Información, y ha sido ponente en distintas conferencias sobre temas relacionados.*

Antes de proceder con una discusión más detallada sobre esta conferencia, me parece oportuno resaltar las dos características que creo la hicieron especial:

1. La demostración en vivo de una penetración compleja y vistosa. Su ejecución ilustró una plétora de técnicas, debilidades y programas para encontrarlas y explotarlas.
2. Trató en conjunto a los diversos aspectos que comprende la seguridad informática, sin abocarse a sólo uno de ellos.

Habiendo comenzado por lo más destacado, se procede a precisar otras características favorables de la ponencia:

- Más allá de un tratamiento aislado de las medidas de seguridad, se expuso la importancia de entretrejerlas con las operaciones habituales de los sistemas que resultarán fortalecidos. Creo que este punto es importantísimo por su conexión con el *mundo real* de la seguridad informática, incluyendo sus limitantes (como la disponibilidad de los datos, por mencionar una de ellas).
- Como se indica en el título, esta conferencia trató con el desarrollo de una **estrategia** para asegurar la información. Incluyó varios puntos significativos por ir más allá de las implementaciones computacionales de la seguridad y estar, más bien, en el campo de lo humano. Por ejemplo, se habló de la importancia de tener como aliados y/o patrocinadores a los altos mandos del entorno que se desea asegurar; las medidas de protección adquieren mayor eficacia cuando no encuentran obstáculos administrativos.
- Se habló también de normativas y estándares internacionales de gestión de seguridad, como ISO/IEC 27000 y los de *British Standards Institution*. Para quien desea desenvolverse profesionalmente en la seguridad de la información, es clave conocer los estándares del momento.
- Ya que una solución completa de mejoría en la seguridad requiere una gran cantidad de puntos a resolver, la implementación de un plan toma en cuenta múltiples aspectos:

- Evaluación de riesgos. Es necesario poner en la balanza el costo de la protección y el valor de lo protegido para adaptar las políticas de la manera más adecuada. Para dicho efecto existen métodos **cuantitativos** y métodos **cuantitativos** (siendo más complicados los últimos, por la cantidad de información y seguimiento que ocupan).
  - Después de la evaluación se elabora un **plan general de mitigación**, el cual describe los procedimientos a mediano y largo plazo para incorporar las mejoras de seguridad al ambiente objetivo.
- Finalmente, me gustaría elogiar profusamente la demostración de *hackeo* que se hizo en esta conferencia. Consistió en la penetración de un servidor *web* por vía de su página de autenticación, de una forma completamente apegada a lo que podría suceder en realidad. Ejemplificó conceptos como *inyección SQL*, escalamiento de privilegios, uso de Nmap y Netcat. En cuanto a la enseñanza, creo más interesante el *cómo* que el *qué*, siendo que este tipo de exposiciones son exactamente lo que se quería desde el principio para INTROMISIÓN.

### 7.3. José Galaviz Casas: *La criptografía nuestra de cada día*

*José Galaviz es egresado de la carrera de Matemático de la Facultad de Ciencias de la UNAM, institución en la que realizó también sus estudios de posgrado en Ciencias de la Computación. Es autor de libros de texto y de divulgación y en 2007 le fue otorgada la Distinción Universidad Nacional para Jóvenes Académicos en Docencia en Ciencias Exactas. Es profesor de tiempo completo de la Facultad de Ciencias desde 1998.*

En el marco de este evento, enfocado a la demostración de fallas de seguridad y su explotación, también es de suma importancia mostrar la teoría detrás de los mecanismos vulnerados; esta conferencia cumple muy bien el papel de la divulgación del funcionamiento de los sistemas criptográficos. Lo logra de una manera comprensible para todo público con mínimos requisitos de conocimiento técnico - si acaso, la teoría matemática (aunque el panorama global de los conceptos no lo exige). En el transcurso de la ponencia se aportan diversas nociones importantes para la formación de una comprensión completa de la seguridad y su implementación. En lo sucesivo, algunas de ellas:

- La diferencia entre **criptografía** y **esteganografía**. Si bien se trató principalmente con la primera en este evento, es importante tener en cuenta a

la esteganografía como elemento para la formulación de mensajes secretos, intentando evitar que se conozca su misma existencia. Esta conferencia contiene la única aparición del concepto en todo el evento; considero importante su inclusión como tema separado en ediciones subsecuentes a ésta.

- **Criptoanálisis.** En el deseo de intervenir comunicaciones ajenas está forzosamente inmiscuído este conjunto de técnicas; los encargados de la seguridad deben estar conscientes de sus implicaciones y alcances para poder fortificar la protección de las transacciones que les ocupan.
- Cifrado **simétrico** y **asimétrico**. Se abordó muy profundamente este campo, indicando diferencias clave entre estos dos esquemas (usos más frecuentes de cada uno, implementaciones computacionales, ventajas y desventajas). La importancia de esta distinción reside en las herramientas utilizables para atacar al uno o al otro; este conocimiento aporta elementos a la “caja de herramientas” del personal de seguridad para fortalecer sus prácticas cotidianas.
- Funciones de **un solo sentido**. El mencionar esta particularidad matemática y computacional es sumamente atinado, dado que la criptografía actual se basa completamente en ella para justificar su robustez. Se discute, incluso, en la sesión de preguntas y respuestas, que de llegar a cierto punto la computación cuántica, los algoritmos del presente quedarían invalidados.
- **Implementaciones criptográficas.** La teoría expuesta no tiene sentido sin la mención de los protocolos y sistemas que las utilizan. Se complementó en esta conferencia la base matemática con ejemplos concretos que la utilizan (mencionados, también, en el primer capítulo de este trabajo).
- Ataque *Man in the Middle*. De gran relevancia por el riesgo que implica y la frecuencia con que ocurre. Es el ataque más severo a los sistemas criptográficos mencionados y debe ser considerado en todo momento para paliar sus efectos.

#### 7.4. **Sergio J. Rojas Hernández: *Linux Firewall en 10 líneas***

*Pasante de Licenciatura en Ciencias de la Computación de la Facultad de Ciencias UNAM. Ha trabajado como administrador de sistemas en el Instituto de Fisiología Celular, UNAM, y como consultor de tecnología y desarrollo de sistemas en el Centro de Investigaciones en*

*Enfermedades Infecciosas en el INER de la Secretaría de Salud. Es socio fundador de Penwin-tux.com, una empresa de consultoría de servicios de Internet y desarrollo de sistemas. Desde 1997 ha brindado consultoría para la instalación de servidores con Linux en el Instituto de Biología, Instituto de Investigaciones Filosóficas, Dirección General de Bibliotecas, Dirección General de Administración Escolar, Facultad de Medicina y en el Posgrado en Ciencias Biomédicas de la UNAM. Como docente, ha sido ayudante de profesor en la Facultad de Ciencias en materias optativas relacionadas con sistemas operativos, redes de computadoras y seguridad en cómputo. En el sector privado, ha trabajado como consultor en varias agencias de publicidad y como director de sistemas en ProezaSLAI en el 2001. Experto en servidores web con Apache y de correo electrónico con sendmail-MailScanner-Spamassassin, routers y firewalls en Linux, desarrollo sistemas en Perl, PHP y bases de datos en MySQL sobre plataformas Linux, MacOS y Windows.*

Considerando que es el primer punto de entrada a una red, la implementación de un *firewall* como método de protección eleva considerablemente la dificultad de su penetración, aún cuando sea muy sencillo. En esta conferencia se habló de cómo poner en funcionamiento un *firewall* así descrito, exponiendo sus principales características:

- **Segmentación de redes.** Cuando existe la necesidad de restringir el paso de paquetes a distintas áreas de una red interconectada, se puede desplegar un *firewall* para ejercer las políticas de acceso y exclusión debidas. Esto es una práctica muy importante ya que contribuye a la estratificación de la seguridad de la información, interponiendo obstáculos en la obtención de la información más sensible.
- **Dualidad** en el funcionamiento. Un *firewall* puede fungir como restricción de acceso, pero también como enrutador. Una buena práctica en soluciones *web*, por ejemplo, es redirigir el tráfico de acuerdo con su destino específico (servidor HTTP, base de datos, etc.), para limitar los daños si es que alguno de los elementos es comprometido. Así, un *firewall* correctamente instalado ayuda a formar las trayectorias de los datos.

Además de la teoría asociada al funcionamiento, se planteó un ejemplo concreto (y, desafortunadamente, muy factible en la vida real) sobre una empresa que desea hacer “invisible” a su departamento de finanzas, dado el involucramiento del dueño en prácticas de dudosa rectitud. El caso expuesto paso a paso me parece muy apropiado para demostrar las propiedades fundamentales del *firewall* mencionado, haciendo hincapié en los siguientes puntos:

- Se estableció la función específica de `netfilter` e `iptables`, el conjunto de herramientas para definir y modificar el comportamiento del tráfico que



circulará por el *firewall* en cuestión. Extremadamente útiles por su pertenencia al *kernel* de Linux, y, por tanto, su gratuidad y diversidad de uso.

- La mención de la manera abstracta de organizar este tipo de *firewall* es muy importante: primero se establecen las políticas generales, a las cuales se agregarán excepciones después. Constituye el pensamiento detrás de gran cantidad de aspectos de la seguridad, tomando en cuenta primero todo lo que se debe restringir terminantemente y permitir después el acceso a procesos y/o personas con funciones singulares.
- **Rastreo de conexiones.** El querer ocultar al segmento de finanzas implica la negación de conexiones provenientes de fuera de éste. Cuando un dispositivo dentro del segmento desea abrir una conexión legítima (por ejemplo, con un servidor *web*), la respuesta del servidor se perderá por esta misma política, a menos que se habilite el rastreo de conexiones. Así, es sumamente significativo aprender sobre esta función de *netfilter* con *iptables*, ya que permite continuar con actividades “normales” dentro del segmento de finanzas, sin divulgar de otra forma su presencia en la red.
- Uso de **bitácoras.** No se puede enfatizar suficientemente el valor de las bitácoras en la seguridad. Representan una poderosísima herramienta forense en caso de haber intrusiones y su utilidad es innegable en la búsqueda de problemas de desempeño.



## 8. Talleres

---

### 8.1. Fernanda Sánchez Puig: *Administración segura de sistemas Linux*

*Fernanda Sánchez es egresada de la carrera de Ciencias de la Computación de la Facultad de Ciencias de la UNAM. Participó en el proyecto PUEMAC del Instituto de Matemáticas de la UNAM. Ha impartido diversos cursos como ayudante y profesor titular dentro de la misma institución. Actualmente es Técnico Académico de la Facultad de Ciencias realizando labores de administración de sistemas y estudia el posgrado en Ciencias de la Computación en el Instituto Investigaciones en Matemáticas Aplicadas y Sistemas.*



El ser el primer taller de la semana que abarcó INTROMISIÓN, lo aportado aquí fue de gran importancia para sentar las bases de conferencias y talleres subsecuentes. Los puntos, a mi parecer, más importantes:

- La seguridad **no es perfecta**. Considero muy adecuada la indicación que la seguridad es una medida que tiene limitaciones. Se debe ejercer activamente, teniendo en cuenta que el objetivo es reducir las pérdidas en caso de una intrusión.
- Políticas de seguridad **adaptables**. Cada configuración y conjunto de prestaciones informáticas es distinto; las políticas que gobiernan su seguridad también lo deben ser. Aunque hay pautas generales para ciertos aspectos de la seguridad, una solución completa exige tratamiento específico.

- Seguridad **física**. Una omisión fácil de cometer al centrar la atención en la seguridad de red, por ejemplo. Teniendo en cuenta que muchos ataques a la integridad surgen dentro del perímetro que se desea proteger, es de suma importancia incorporar medidas que impidan dicho acceso traicionero.
- **Reducción** de potenciales vulnerabilidades. La discusión ofrecida en este taller con respecto a las medidas habituales para disminuir la probabilidad de intrusiones corresponde a un nivel básico de conocimientos en seguridad, contribuyendo nociones útiles para el manejo seguro de un entorno casero, inclusive. Como ejemplos concretos de lo anterior están evitar el uso de cuentas de *invitado*, cambiar frecuentemente las contraseñas, usar un *firewall*, desactivar servicios y protocolos innecesarios y revisar las bitácoras.
- Herramientas **básicas** de administración segura. Creo que la manera más certera de causar inquietud en los asistentes y mostrar realmente de qué se trata la seguridad es mediante las herramientas que se utilizan cotidianamente para dicho fin. Aquí se dio una introducción a algunos programas empleados para tareas habituales como detección de servicios en la red, conexiones remotas seguras, detección de intrusos y búsqueda de código malicioso. Con las anteriores creo que se comienza a formar una idea completa del tipo de mecanismos que deben ocupar los administradores para fortalecer los sistemas de los que se ocupan.
- Buena preparación para el peor de los casos. Las conclusiones de este taller hicieron ver claramente que la seguridad “se trata de un ciclo constante de mejorar y evaluar medidas de protección”. Siempre es necesario aclarar que la seguridad no es una práctica pasiva, que resuelve los problemas en una sola sesión. Más importantemente, se debe tener en cuenta el enfoque inherentemente pesimista de este ramo: actuar siempre esperando que cualquier ataque será el peor posible.

## 8.2. Max de Mendizábal: Configuración básica de un servidor de nombres (DNS)

*Max de Mendizábal es Físico por la Facultad de ciencias de la UNAM. Obtuvo un Máster en Software Libre en la Universitat Oberta de Catalunya. Sus áreas de estudio son la administración de sistemas, la seguridad informática y el desarrollo de programas con bases de datos utilizando principalmente herramientas de software libre.*

Si bien este taller no abordó *directamente* los aspectos de seguridad relacionados con DNS, su principal atributo fue la profundidad con la que discutió el funcionamiento, despliegue y optimización de DNS (a pesar del título, que haría esperar un tratamiento básico). Los puntos más notorios, a mi criterio:

- Los múltiples significados del término DNS. Siempre es necesario aclarar este punto, dado que DNS se utiliza de varias formas. Citando directamente a la presentación del taller, “DNS es un espacio de nombres jerárquico, una tabla de hosts implementada como una base de datos distribuida, un resolovedor y una serie de bibliotecas, enrutamiento mejorado para el e-mail [...]”.
- **Seguridad.** Dado que DNS se pensó como un sistema abierto, cualquiera puede hacer cierto grado de investigación sobre un dominio al consultar su base de datos. BIND, en sus versiones más recientes, permite un control más fino de la información a la que se puede acceder, ayudando a establecer políticas adecuadas para las necesidades de cada zona.
- Ejemplo de uso **casero** de BIND, paso a paso. Creo que aquí está lo más valioso del taller, ofreciendo la oportunidad a todos los asistentes de probar en casa lo aprendido. Ilustra, a menor escala, lo que requiere la administración de zonas DNS, y las políticas que la pueden hacer más segura.
- Depuración de una configuración. Se dieron ejemplos extensivos sobre cómo probar y asegurar el buen funcionamiento de la instalación detallada a lo largo del taller. Los sistemas de gestión de la seguridad de la información (SGSI) utilizan el círculo de Deming como base estratégica de mejora continua de la calidad [9]; la depuración del sistema corresponde al tercer paso en el círculo (Planear-Hacer-**Verificar**-Actuar ó Plan-Do-**Check**-Act).

### 8.3. Mario Arturo Pérez Rangel: *Uso de Wireshark*

*Egresado de la carrera de Matemáticas de la Facultad de Ciencias de la UNAM. Con experiencia de 15 años en el manejo y administración de redes LAN, así como programación. Actualmente es Coordinador del Centro de Operación de la Red en la Facultad de Ciencias de la UNAM.*

Wireshark es de valor indiscutible por sus capacidades de disección con gran detalle del tráfico en una red. Este taller ofreció una visión muy completa de las tareas más comunes y útiles que se dan a esta herramienta. El taller parte de elementos de muy bajo nivel (como la composición de *frames* y paquetes en

Ethernet), haciendo más clara su interacción en niveles más altos. Entre las facetas de Wireshark que se presentaron están:

- Uso de **filtros**. La cantidad de paquetes en una red es muy grande, aún cuando la red misma no lo sea. La búsqueda de problemas e intrusiones se volvería un problema intratable de no ser por herramientas para filtrado como ésta. Aquí se expusieron algunos ejemplos valiosos de esta práctica, enfocados a la averiguación de los usos que están haciendo los usuarios:
  - Usos indebidos como transferencia de archivos por edonkey. Se filtra el tráfico de acuerdo con el puerto empleado, encontrando fácilmente a los dispositivos que comparten archivos ilegalmente.
  - Filtrado por *host*. Útil cuando se desea hacer una investigación a fondo sobre las actividades de un dispositivo en particular, ayudando a encontrar y descartar actividades sospechosas.
- **Conexiones**. Cuando se quiere seguir completamente la comunicación entre dos puntos (a diferencia del filtrado, donde puede ocurrir que sólo se muestre una dirección de ella), se puede habilitar a Wireshark para seguir la *conversación*. De esta forma se detecta completamente un intercambio, sin pérdida de información. Muy provechoso para examinar aún más de cerca las actividades de un dispositivo cuando se tienen más elementos para sospechar de él.
- Estadísticas de **uso de ancho de banda**. Encontrar cuellos de botella y puntos de uso excesivo son tareas cotidianas en el mantenimiento de cualquier red. Además, no sólo es una tarea de depuración y mejora del desempeño, pues existen muchos ataques fácilmente detectables por su uso intensivo del ancho de banda: cuando un dispositivo es anfitrión para la propagación de *spam* y *gusanos*, suele intentar enviar mensajes de manera masiva.

## 8.4. Patricio López-Serrano Erickson: *Uso de ettercap*

*El autor.*

Cuando supe que impartiría un taller, el objetivo primordial que tenía en mente era mostrar de manera sensacionalista algún ataque de seguridad, dado el espíritu del evento. No tardé mucho en concluir que ettercap-ng cumpliría muy bien el requisito. Para lograr el efecto deseado, primero expuse los fundamentos teóricos de la herramienta, para después demostrar tres aplicaciones en vivo:

- **Interceptar una contraseña transmitida en texto claro (plano).** Escogí un servicio de correo electrónico web cuya autenticación se realizara sin cifrar los datos para mostrar a la concurrencia la facilidad del espionaje en un entorno público. El asombro fue amplio al ver flotando por la pantalla del atacante (uno de los participantes) los datos que yo acababa de ingresar desde mi explorador de Internet. Creo importante este ejercicio por su capacidad de elevar la conciencia sobre los riesgos de espionaje al navegar irresponsablemente en puntos de acceso públicos y/o desconocidos.
- **Uso de filtros para sustitución arbitraria.** Aunque la aplicación de este ejemplo no es útil directamente en la fortificación de la seguridad, aclara firmemente el concepto de filtrado de paquetes e inyección del contenido al mostrar cómo todas las imágenes de cualquier página HTML pueden susituírse por otra, a la sazón del atacante. En términos estrictos, podría utilizarse para desalentar la navegación inapropiada, aunque en la presentación se mostró con un tono bromista.
- **Visor de circulación de imágenes.** Para esta demostración aumenté al repertorio una herramienta llamada *driftnet* [19], diseñada para extraer imágenes de flujos TCP. Utilizadas en conjunto, ambas herramientas son capaces de formar una estación de monitoreo de uso de la red; si ésta se coloca en un lugar público (el pasillo de los sanitarios o la cafetera, en un ambiente laboral) puede servir como amable persuasión para dar buen uso al Internet. Insté a los participantes a buscar “muchachas” en un buscador web (con el filtro de contenido activado, por supuesto), lo cual resultó en gran algarabía general.

## 8.5. Thomas d’Otreppe: *aircrack-ng*

*Creador de aircrack-ng, egresado de la Haute Ecole de Bruxelles. Diseñó, junto con Mati Aharoni, Offensive Security WiFu, un curso proactivo de seguridad inalámbrica.*

Ya se vio en la sección teórica destinada a *aircrack-ng* la utilidad que tiene para conducir auditorías de seguridad sobre un elemento que en muchas ocasiones constituye el punto de entrada a la red de escuelas, negocios y hogares: las redes inalámbricas. Por eso mismo, es elemental conocer las fortalezas y debilidades de los protocolos que las protegen. Este taller, ocupado de ello, fue sumamente valioso por razones como las siguientes:

- El contenido impartido fue el más actual de todas las exposiciones, al tratar con un ataque a WPA de muy reciente implementación. Las distintas vías de

vulneración de WEP eran muy conocidas (y utilizadas) al momento de este evento; no así con WPA. Otro aspecto muy interesante que se trató en este taller fue la tendencia emergente de utilizar los procesadores de gráficos para ayudar al CPU en tareas muy intensivas: se ilustró el gran impacto positivo que tiene el delegar operaciones de fuerza bruta a arquitecturas de gráficos que lo permitan (como CUDA, por ejemplo). Así, se habló no sólo de hallazgos recientes en seguridad, sino también de su relación y aprovechamiento del *hardware* (en este caso, de las capacidades masivamente paralelas de los procesadores de gráficos modernos).


- El mismo autor expuso el taller. Es incomparable la experiencia de discutir y aprender sobre una pieza de *software* con quien la conoce mejor: su desarrollador.
- Conocer mejor a WPA es esencial, ya que se utiliza en un número creciente de redes inalámbricas (esto porque se ha corrido la voz de la *gran* inutilidad de WEP para proteger cualquier cosa).
- Es debido señalar que el tallerista comenzó su explicación de WPA-PSK desde un nivel muy bajo (i.e., la descripción del protocolo de *handshake*), incluyendo ilustraciones sobre la forma de los paquetes utilizados para este tipo de intercambio. Lo anterior aseguró que los asistentes tuvieran una noción completa del protocolo y por qué es débil, sin limitarse a su mera explotación.
- Se mencionó a Airbase-ng, una herramienta utilizada para simular un punto de acceso inalámbrico. Airbase-ng intenta que los clientes se asocien al dispositivo del atacante, capturando así el *handshake* WPA y cifrando/descifrando los paquetes en circulación. Me parece muy interesante esta metodología por ser distinta a la necesaria para atacar un punto de acceso con WEP: engañar a los clientes.
- Se montó un ambiente de prueba para que todos los participantes pusieran en práctica exactamente aquello que habían visto minutos antes en las diapositivas de la presentación. Considero un muy buen recurso para las demostraciones de seguridad el discutir las teóricamente primero y pasar a un ejercicio en vivo después. Además, el expositor dio a la concurrencia la oportunidad de resolver el reto que había preparado (penetrar una red WPA) sin su asistencia y después llevó a cabo una ejecución acompañada, para quien así lo requiriera.



## 9. El reto de penetración

---

*Un buen día vas por ahí y tropiezas con una red inalámbrica llamada METETE SI PUEDES. El hallazgo captura tu atención. Obviamente, el administrador es un engreído. Como buen usuario avanzado de computadoras, y en perfecta armonía con tu inquebrantable ética personal, decides darle un golpe a su ego, informándole sobre todas las fallas que encuentras en su mediocre sistema. Tras recolectar algo de información, te enteras que existe un archivo en la computadora del administrador que contiene su correo electrónico personal y número telefónico. Tu misión es obtener el contenido del archivo e informarle acerca del surtido rico de fallas que encontraste en el camino.*

o sólo fue un reto para los concursantes; su implementación fue el aspecto más complicado de todo el evento. La gran dificultad residió precisamente en la calibración de la dificultad: ¿cómo diseñar un concurso que atrape la atención de los participantes? Había, en un principio, dos salidas fáciles. La primera, instalar un sistema Linux actualizado al día, volviendo prácticamente imposible su rompimiento y resultando en la rifa de los premios (en vez de su obtención por vías de mérito). La segunda, ofrecer un sistema Windows, quitando todo dejo de emoción que podría tener el concurso. Optando por el camino de la diversión y la adrenalina, se montó un sistema cuyas características son más claras en términos de las etapas que lo compusieron. Cabe destacar que la ruta de penetración aquí descrita no es necesariamente la única, sino la planeada durante la formulación del reto.

1. A cada equipo inscrito en el concurso se asignó una computadora objetivo, identificable por su dirección IP, además de un nombre de usuario. Estaba terminantemente prohibido atacar cualquier otra computadora o interferir con el progreso de los demás equipos.

2. Se utilizó un punto de acceso inalámbrico funcionando bajo el protocolo de cifrado WEP. Se asociaron a ella las tres computadoras objetivo. Dentro de la red no fluía gran cantidad de paquetes, haciendo necesario un ataque con inyección de paquetes mediante `aircrack-ng`, por ejemplo.
3. Una vez obtenida la llave de la red inalámbrica y asociadas las computadoras atacantes a ella, se debía realizar una tarea de reconocimiento de la computadora asignada. Una opción para lo anterior sería ejecutar `nmap` con la dirección IP objetivo, descubriendo así los puertos abiertos y servicios activos.
4. La exploración anterior revelaría que cada computadora objetivo fungía como servidor `telnet` y `ssh`, brindando la primera información sobre potenciales puntos de entrada para los atacantes. En esta fase se planeó que los participantes realizaran un ataque de fuerza bruta o diccionario sobre `telnet` para averiguar la contraseña asociada a su nombre de usuario. Dicho ataque podría llevarse a cabo utilizando `THC-HYDRA`, aunque hubo un equipo que escribió su propio *script* en ruby para probar distintas contraseñas. Dado que el avance de esta etapa se estancó, la organización decidió dar como pista el diccionario que contenía las contraseñas de todos los equipos, permitiendo un paso más rápido a la siguiente fase.
5. Para este momento, los equipos atacantes tendrían acceso de usuario *mortal* a su computadora objetivo (i.e., con un nivel reducido de permisos). El atacante perspicaz y ambicioso no perdería la oportunidad de ascender su nivel de permisos a `root`, lo cual podría hacer notando que el sistema operativo huésped poseía una versión insegura del *kernel*, susceptible a un *exploit* de `vmsplICE`, programado por milworm[20].
6. Contando ya con privilegios de `root`, en la carpeta de éste (`/root`), se encontraba un archivo de texto incitando a los participantes a seguir intentando llegar al final del reto. Más técnicamente, se especificaba la ruta del verdadero archivo mediante un *hash* md5. El equipo ganador, por ejemplo, sorteó este obstáculo obteniendo el md5 de cada una de las posibles rutas del sistema huésped y comparándolo con la hallada en el archivo.
7. Así, la última fase consistía en navegar hasta el directorio hallado en el paso anterior y leer el archivo especificado. ¡No! - todavía faltaría algo más. El archivo en cuestión era un zip cifrado con otra contraseña del diccionario utilizado en el paso de entrada por fuerza bruta, al cual además se había modificado la extensión. Sólo un análisis del encabezado revelaría el tipo verdadero del archivo, haciendo necesario después su descifrado. Ahora sí,

podrían ver en texto plano las instrucciones para enviar una frase secreta a la organización, indicando que habían ganado.

Para completar el relato del reto, he aquí el reglamento oficial:

1. A tu equipo le será asignada una de tres posibles computadoras (*intromision2*, *intromision3*, *intromision4*) mediante una dirección IP. Queda estrictamente prohibido que intentes acceder a cualquier equipo que no te corresponda y/o que interfieras con los procesos de comunicación/intercambio de información inherentes a los demás equipos. En otras palabras, está prohibido que interfieras con el progreso de cualquier otro equipo.
2. Queda estrictamente prohibido que lleves a cabo cualquier actividad ilegal dentro del marco del evento. Más específicamente, está prohibido que realices cualquier acción indebida sobre sistemas computacionales de los que no eres un usuario autorizado.
3. El comité organizador sólo proveerá el *hardware* que habrá de ser penetrado. Es tu responsabilidad llevar cualquier implemento necesario para concursar (laptop, tarjeta inalámbrica, periféricos, etc.).
4. El reto comienza el lunes 24 de noviembre a las 12:00 horas, y termina el viernes 28 de noviembre a las 12:00 horas. Cada día dispondrás de 24 horas para resolver la etapa. La manera de resolver la etapa del día anterior se publicará a las 12:00 horas.
5. Deberás documentar de manera reproducible por el comité organizador tu progreso a lo largo de todo el concurso (entregando un archivo en formato `.txt`, `.doc` ó `.pdf`, lo que mejor te acomode). Deberás enviar el documento con tu progreso del día a `intromision@ciencias.unam.mx` cada día, antes de las 12:00 horas (excepto el lunes, obviamente). Queda estrictamente prohibido que borres los rastros de tus acciones en el sistema penetrado (historial de shell, bitácoras, etc.).
6. El esquema de puntuación es el siguiente:
  - Por resolver correctamente la etapa, en el tiempo estipulado y sin solicitar sugerencias, se asignarán diez (10) puntos.
  - En cada etapa se podrán solicitar dos (2) sugerencias. Cada solicitud de sugerencia restará a tu puntaje máximo de la etapa la cantidad de dos (2) puntos. Así, si solicitas las dos sugerencias, podrás obtener un máximo de  $(10-2-2) = 6$  puntos.

- Si termina el tiempo de la etapa y no logras resolverla, podrás pasar a la siguiente con un puntaje de cero (0) para seguir concursando.
7. El intercambio de información entre equipos no es algo muy inteligente, así que queda prohibido.
  8. Puedes recurrir a cualquier fuente para obtener información sobre cómo superar el reto.
  9. Está estrictamente prohibido que intentes acceder físicamente a los sistemas que deberás penetrar. Todo el reto está planeado para llevarse a cabo de manera remota.
  10. El comité organizador solucionará cualquier imprevisto en el momento.
  11. Disfruta el escarmiento que le darás al administrador.

**Parte III**

**Reflexiones**



## 10. Notas finales

---

### 10.1. Logros



Como resultados de su esfuerzo la organización puede contar:

- **Participación de personalidades importantes en el campo de la seguridad.** En este respecto, la presencia de Thomas d'Otreppe (¡una celebridad en su medio!) fue valiosísima por sus aportes de punta a nuestro evento. Ningún otro participante mostró contenido tan novedoso y ofensivo. La conferencia de José García Sabbagh podría tomarse como modelo para futuras conferencias, dada la efectividad de su exposición en vivo de un ataque de inyección SQL, además de su pertenencia a la comunidad de seguridad, siendo *pen-tester* profesional.
- **Amplio poder de convocatoria.** El número de asistentes a las conferencias y talleres fue sumamente satisfactorio, mostrando la efectividad de los medios de difusión empleados. El reto también tuvo una afluencia respetable, añadiendo una sensación de apremio a los participantes. Como ya se había mencionado, fue muy agradable enterarse de la llegada de las invitaciones a dependencias ajenas a la Facultad de Ciencias.
- **Motivación adicional.** Gracias al generoso involucramiento de los patrocinadores fue posible hacer un montaje elegante y completo: parafernalia publicitaria de alta calidad (ejemplo de ello, las playeras, estampas y pósters), una página web atractiva e interesante y excelentes premios para los

ganadores del reto. Todo lo anterior contribuyó a la difusión como evento “serio” (i.e., bien formado, porque nos gustaría que la seriedad quedara fuera) de INTROMISIÓN 2008.

- **Fomento de la inquietud.** Presenciando las preguntas de los asistentes, fue claro en el transcurso de los talleres y conferencias que los temas tratados habían causado curiosidad e incomodidad, provocando la necesidad de profundización por propia cuenta.
- **Sentido del humor.** Creemos que es un excelente recurso didáctico, y quisimos que fuera parte del espíritu del evento. Me ocupé de que se manifestara en el nombre mismo, el logotipo, el reto y el taller que impartí. Agrega una actitud distinta al tratamiento de la seguridad ofensiva, que generalmente se aborda con gran seriedad por sus implicaciones.

## 10.2. Metas

- Ver a INTROMISIÓN convertido en un evento anual. Aunque no se pudo hacer en 2009 (por falta de recursos humanos y monetarios), creo que tiene todo el potencial de volverse un distintivo de la Facultad de Ciencias, haciéndola única por acercarse a la seguridad informática de esta manera.
- En futuras ediciones, quisiera un reto de penetración más elaborado, competitivo y dinámico, que haga confrontarse entre ellos a los equipos (al estilo del concurso de “capturar la bandera” que se juega en DEF CON).
- Globalmente, creo necesario progresar hacia un tratamiento ofensivo en todo lo que implique el evento, al punto que cada conferencia y cada taller sea demostrativo de las fallas que se pueden encontrar en las implementaciones de seguridad.

En general, considero que INTROMISIÓN 2008 cumplió con las expectativas de la organización (el tutor y el autor), al ver lo satisfechos que habían estado tanto los ponentes como los asistentes. Además, transcurrió en el espíritu con el que fue planeado, dejando, con un estilo particular, enseñanzas de gran utilidad (i.e., cómo vulnerar redes cifradas con WPA o cambiar el escudo de tu escuela en su página *web*, por citar un par de ellas).



# Bibliografía

- [1] Buffer overflow. [http://en.wikipedia.org/wiki/Buffer\\_overflow](http://en.wikipedia.org/wiki/Buffer_overflow).
- [2] Cifrado extremo a extremo. [https://www.ccn-cert.cni.es/publico/2008/401/es/c/end\\_end\\_encryption.htm](https://www.ccn-cert.cni.es/publico/2008/401/es/c/end_end_encryption.htm).
- [3] CWE-476: NULL Pointer Dereference. <http://cwe.mitre.org/data/definitions/476.html>.
- [4] Dangling pointer. [http://en.wikipedia.org/wiki/Dangling\\_pointer](http://en.wikipedia.org/wiki/Dangling_pointer).
- [5] End-to-End Encryption. <http://polyarista.tripod.com/>.
- [6] Ethernet LAN Security. <http://www.javvin.com/networksecurity/EthernetLANSecurity.html>.
- [7] Executable space protection. [http://en.wikipedia.org/wiki/Executable\\_space\\_protection](http://en.wikipedia.org/wiki/Executable_space_protection).
- [8] How RC4 Works. <http://www.wireless-center.net/Wi-Fi-Security/2209.html>.
- [9] Information security management system. [http://en.wikipedia.org/wiki/Information\\_security\\_management\\_system](http://en.wikipedia.org/wiki/Information_security_management_system).
- [10] John the ripper password cracker. <http://www.openwall.com/john/>.
- [11] Network Security Glossary. <http://www.watchguard.com/glossary/a.asp#ARP>.
- [12] Null-pointer dereference. [http://www.owasp.org/index.php/Null-pointer\\_dereference](http://www.owasp.org/index.php/Null-pointer_dereference).
- [13] Replay attack. [http://en.wikipedia.org/wiki/Replay\\_attack](http://en.wikipedia.org/wiki/Replay_attack).

- [14] Stack buffer overflow. [http://en.wikipedia.org/wiki/Stack\\_buffer\\_overflow](http://en.wikipedia.org/wiki/Stack_buffer_overflow).
- [15] The AVISPA Project. <http://avispa-project.org/>.
- [16] Transport layer security. [http://en.wikipedia.org/wiki/Transport\\_Layer\\_Security](http://en.wikipedia.org/wiki/Transport_Layer_Security).
- [17] DORAISWAMY, A. Wireless Security - How WEP works. <http://palisade.plynt.com/issues/2006Dec/wep-encryption/>.
- [18] HOLZMANN, G. J. *Design and Validation of Computer Protocols*. Prentice Hall, 1991.
- [19] LIGHTFOOT, C. Driftnet. <http://www.ex-parrot.com/~chris/driftnet/>.
- [20] MILWORM. Linux Kernel 2.6.17 - 2.6.24.1 vmsplice Local Root Exploit. <http://milw0rm.org/exploits/5092>.
- [21] NACHREINER, C. Anatomy of an ARP Poisoning Attack. <http://www.watchguard.com/infocenter/editorial/135324.asp>.
- [22] TANENBAUM, A. S. *Redes de computadoras*. Prentice Hall, 2003.
- [23] TEWS, E., WEINMANN, R.-P., AND PYSHKIN, A. Breaking 104 bit WEP in less than 60 seconds. <http://eprint.iacr.org/2007/120.pdf>.