Evitando la Exploración de Puertos a través de *DEP*: un Sistema Dedicado

G. Aguirre, C. Alaniz, R. Apolloni, F. Piccoli *
Departamento de Informática
Universidad Nacional de San Luis
Ejército de los Andes 950
5700 - San Luis
Argentina
e-mail: {mpiccoli}@unsl.edu.ar

Resumen

La seguridad en redes es un tópico que ha captado la atención en la mayoría de las investigaciones y desarrollos relacionados a redes. Tener redes seguras significa definir políticas de seguridad y tener herramientas capaces de detectar y prevenir distintos ataques.

Existen numerosos puntos a considerar a la hora de hacer una red segura. Uno de ellos son las vulnerabilidades provistas por el software. Las vulnerabilidades son la puerta de acceso para los distintos ataques. El sistema operativo LINUX, la familia de protocolos de comunicación TCP/IP y el software de aplicación poseen varias.

Los puntos de ataque de TCP/IP son producto de sus características jerárquicas y la amplia familia de protocolos. Si bien existen muchos ataques, la exploración de puertos es uno de los más comunes. La presente propuesta tiene como objetivo mostrar una herramienta: Sistema para la **D**etección de **E**xploración de **P**uertos(*DEP*), la cual permite detectar actividades de exploración de puertos en la computadora local y evitar el acceso de extraños, no sólo en el momento sino también en el futuro.

Palabras Claves: Redes de Computadoras, Seguridad de Redes, Modelo Cliente/Servidor, Exploración de Puertos, Denegación de Servicios.

1. Introducción

Desde los inicios de las redes de computadoras, la seguridad fue un tópico que concentró la atención de los investigadores y desarrolladores de tecnologías de redes. Con el correr del tiempo y el crecimiento de las redes de computadoras, más y más personas se dedican a investigar y desarrollar herramientas para proveer redes seguras.

*Grupo subvencionado por la UNSL y ANPCYT (Agencia Nacional para la Promoción de la Ciencia y Tecnología)

De las investigaciones realizadas se determinó la dificultad o imposibilidad de establecer una teoría general para hacer seguras todas las redes, en cambio sí se logró establecer los lineamientos a seguir: definir una política de seguridad.

La seguridad de las redes de computadoras depende de la vulnerabilidad del software disponible y de los ataques que sufren, tanto internos como externos. Las vulnerabilidades del software constituyen los caminos a través de los cuales se pueden realizar los ataques.

Así como todas las vulnerabilidades no son conocidas, tampoco lo son todos los posibles ataques. En estos últimos años se han desarrollado productos para detectar tanto las posibles vulnerabilidades de los sistemas y de los servicios de red, como los posibles ataques que se pueden perpetrar.

Si bien existen muchas formas de vulnerar una red de computadoras o una computadora específica de la red, los dos ataque más comunes son: la exploración de puertos y la intromisión de intrusos, ambos con el objetivo de hacer un mal uso de los sistemas de información a acceder.

Toda las redes pueden sufrir numerosos ataques, algunos de los más conocidos o empleados son: *Footprinting*, *Fingerprinting*, *Exploración de puertos-vulnerabilidades*, *Exploración basada en el protocolo ICMP*, *Sniffing*, *Eavesdropping*, *Snnoping*, *IP Spoofing*, *SMTP Spoofing* y *Spamming*, *DoS: Denial of Service*, *Net Flood*, *Smurf*, *TCP Syn Flood*, *Connection Flood*, *SMTP Flood*, *DDos*, *Trinoo*, *Tribe Flood Network y TFN2K*, *Stacheldraht*, *Ping of Death*, *Loki*, *Land*, *Routing Protocols*, *Session Hijaking*, *Source Routing*, *ICMP Redirects*, *Directed Broadcast*, *SNMP*, *TCP Initial Sequence Numbers*, *Tiny Fragment Attack*, *Winnuke*, *Teardrop*, *DNS*, *NTP*, C*aballos de Troya* o *Troyanos*, *IPSec*, *Finger Bomb*, *RPC*, *Buffer-overflows*, *Format Strings*, etc. [13][15].

Para cada uno de los diferentes tipos de ataque se han desarrollado numerosas técnicas y herramientas para su detección, en algunos casos para repeler y/o contrarrestarlos.

En este trabajo proponemos un sistema para detectar la exploración de puertos y evitar el ingreso de intrusos a través de ellos: *DEP*(**D**etección de **E**xploración de **P**uertos). Contar con un sistema que detecte y prevenga la exploración de los puertos, proveerá a los administradores de redes de computadoras de una herramienta capaz de implementar un protocolo, el cual, en base a la información obtenida en los intentos de acceso no autorizado, tomará decisiones en consecuencia. *DEP* es una herramienta que permite: detectar la exploración de puertos de computadoras conectadas en redes (con sistema operativo LINUX) y evitar el ingreso a través de alguno de ellos. En las próximas secciones se detalla la arquitectura del sistema, su modo de funcionamiento y las distintas exploraciones detectadas. Finalmente se muestra la interfaz de usuario de *DEP*.

2. Exploración de Puertos

Generalmente, un aspecto común considerado en la seguridad de la mayoría de las redes de computadoras es la restricción de los accesos no autorizados a la red por parte de personas ajenas a la organización y cuyo único objetivo es dañar, husmear o sustraer información[3]. Estos accesos se producen, principalmente, a través de los puertos de acceso de las aplicaciones de red. Impedirlos o controlarlos es tarea del administrador de la red.

Si bien en computación podemos asociar dos definiciones al concepto de puerto, en este trabajo y dada la temática involucrada, decimos que un puerto es una puerta virtual propia de los protocolos de redes como es el caso de los protocolos TCP/IP [2][4][5]. Los puertos son los canales de comunicación de las aplicaciones de red brindadas por la computadora, controlar su flujo de información permite detectar los intentos de accesos por parte de personas no deseadas.

La exploración de puertos es un método utilizado por personas ajenas a una organización, quienes, sin ninguna autorización, intentan determinar qué puertos se encuentran abiertos, o en uso, en la red para acceder a las aplicaciones correspondientes y efectivizar un ataque. En una exploración se determinan las características de la red o sistema remoto, teniendo como objetivo la identificación de los recursos disponibles y accesibles a través de la red.

Determinar los puertos abiertos de un sistema significa establecer los puntos de acceso al mismo y la explotación de potenciales vulnerabilidades de los servicios detrás de dichos puertos. Existen distintas herramientas a través de las cuales el intruso puede explorar, al mismo tiempos, los distintos puertos de una computadora en la red. Dependiendo de la respuesta recibida, los puertos abiertos accesibles, determina el ataque a realizar. Una de las más populares es la $NMAP$ [11][19].

Cuando se habla de modelo *cliente/servidor* en ambientes de redes, se refiere a la manera en que las aplicaciones involucradas en una comunicación están organizadas. La aplicación que inicializa el contacto es llamada *"aplicación cliente"*, mientras que la aplicación que aguarda la llegada de contactos es denominada *''aplicación servidora"*.

En la exploración de puertos, el atacante es considerado un *cliente*(solicita recursos o información), quien solicita un servicio a la computadora víctima, el *servidor*(comparte sus recursos o gestiona la información solicitada). El término servidor no hace referencia sólo a aquellas computadoras dedicadas a un fin determinado, sino a cualquier host conectado a la red, teniendo algo para compartir. De esta manera pueden existir tanto Servidores DNS, de Correo, de Web, de FTP, de Noticias, de Archivos, de Impresoras, de Terminales, etc.; como servidores de información.

Contar con un sistema que detecte y prevenga la exploración de los puertos, proveerá a los administradores de redes de computadoras de una herramienta capaz de implementar un protocolo, el cual, en base a la información obtenida en los intentos de acceso no autorizado, tome decisiones en consecuencia.

3. Vulnerabilidades del Protocolo TCP/IP

La familia de protocolos TCP/IP(***T****ransport* ***C****ontrol* ***P****rotocol / **I**nternet **P**rotocol*) es considerada el estándar de los protocolos de comunicación entre sistemas informáticos. Siguiendo la filosofía propuesta por la organización *ISO* en la formulación del protocolo *OSI*[3][14][17], el protocolo TCP/IP está estructurado en capas. Cada capa es responsable de llevar a cabo una tarea específica para la comunicación, además de tener múltiples protocolos definidos para cada una. La figura 1 muestra las cuatro capas del protocolo y los protocolos definidos para cada una de ellas.

Desde el punto de vista de seguridad, la familia de protocolos TCP/IP puede ser vulnerada en base a dos conceptos derivados de su diseño [15]:

- Formato de los paquetes: Para lograr la comunicación, cada protocolo incluye

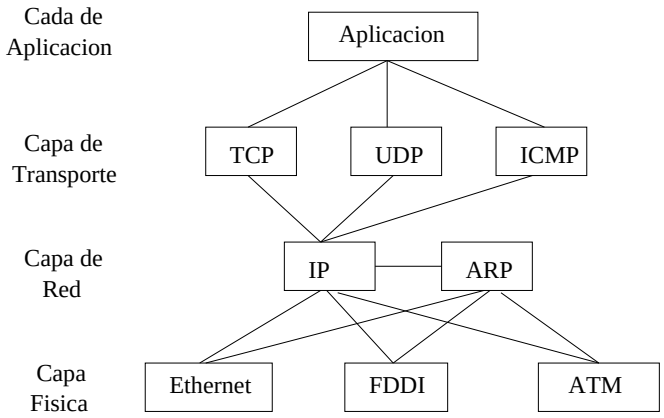

Figura 1: Familia de protocolo TCP/IP

en el paquete sus propios datos, los cuales ofrecen valioso conocimientos al atacante.

- Funcionamiento de los protocolos: Cada paso asociado a los procesos involucrados en el protocolo es capaz de brindar importante información.

Aprovechando estas características ampliamente conocidas, se puede proceder a la exploración de puertos. Para ello, el atacante (denominado también cliente) envía un paquete según un protocolo específico de la capa de transporte, TCP, UDP o ICMP, solicitando un servicio a la computadora víctima, la cual se transforma en servidor. Para cada uno de estos protocolos existen distintas alternativas o técnicas de ataque, cada una con su propio conjunto de datos y acciones.

4. *DEP*: Sistema para la Detección de Exploración de Puertos

El sistema para la *Detección de Exploración de Puertos DEP* trabaja en paralelo al proceso propio del protocolo para la atención de paquetes. *DEP* pasa a formar parte del módulo de atención de servicios, es por ello que está residente en cada computadora perteneciente a la red que se desea monitorear y proteger.

DEP fue desarrollado para detectar exploraciones de puertos en sistemas operativos LINUX. *DEP* es el encargado de determinar posibles intentos de exploración de puertos de servicios brindados por el sistema local. Cada paquete que arriba a la computadora es capturado y analizado por *DEP*. Si el análisis determina un intento de exploración, *DEP* registra al emisor de la solicitud como posible atacante en *"Servicios Denegados"*. La figura 2 muestra la interrelación de *DEP* con la atención de servicios propia del sistema operativo.

Para evitar futuras exploraciones, *DEP* registra la historia de los accesos a la computadora, cada acceso o intento de acceso proveniente de un origen remoto es registrado en *BD* (figura 2). Cada registro almacena la dirección de origen, número de puerto accedido, servicio asociado al puerto, fecha y hora del acceso.

¿Cómo trabaja *DEP*? Al recibir un paquete, dependiendo de las características de éste y del historial de accesos del emisor, *DEP* decide que acciones tomar. Si el paquete contiene información que coincide con las características de alguna de

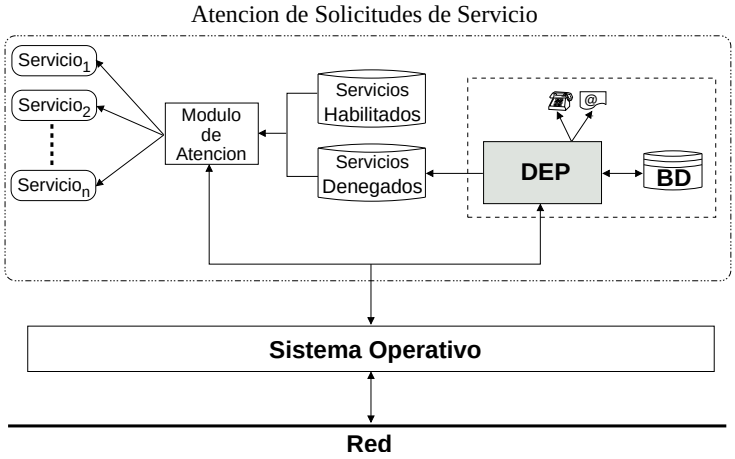

Figura 2: Entorno de DEP

las técnicas de exploración de puertos o ha realizado un número determinado de accesos a puertos diferentes en un umbral de tiempo determinado, se procede con la correspondiente denegación de acceso a cualquier servicio que brinde el sistema local al emisor, identificado a través de su dirección IP. De esta manera el sistema previene cualquier intento de acceso a alguno de los servicios que mantiene activos, evitando así los accesos indebidos o dañinos al sistema.

Si en cambio no se detecta ninguna característica peculiar en la composición del paquete, DEP no toma ninguna acción.

El *número* de puertos diferentes a los cuales puede acceder un usuario remoto en un *tiempo* determinado son ambos parámetros configurables. El proceso de negación de acceso a la computadora es llevado a cabo por las herramientas de *"control de acceso"* provista por el sistema operativo LINUX.

En el momento que DEP detecta un posible intento de exploración de puertos, además de cerrarlos para la dirección remota, alerta al administrador del sistema del evento y de las acciones realizadas.

En las próximas secciones detallamos la estructura interna de DEP como algunos detalles de implementación.

4.1. Arquitectura

DEP es un sistema formado por seis módulos, cada uno cumple una función específica y bien definida. Los seis módulos son: *Capturador de paquetes (CP)*, *Analizador de Confiabilidad(AC)*, *Analizador de Tipo de Mensaje(ATM)*, *Analizador y Detector de Ataques(ADA)*, *Negador Servicios(NS)* y *Alertas(AL)*. En la figura 3 se muestra la estructura interna del sistema y la relación existente entre sus módulos.

La función de cada módulo es:

- *Capturador de paquetes (CP)*: Es el modulo encargado de capturar cada uno de los paquetes que arriban al sistema. Interactúa directamente con el sistema operativo, encargado de recibir los paquetes, y con el módulo AC.

  Para llevar a cabo la tarea de captura y análisis de los paquetes emplea la interfase de socket, específicamente sockets de tipo RAW [9][12][16].

  Los sockets de tipo RAW permite un control absoluto sobre los datos que serán enviados o recibidos a través de la red. Además permite *"espiar"* todo el

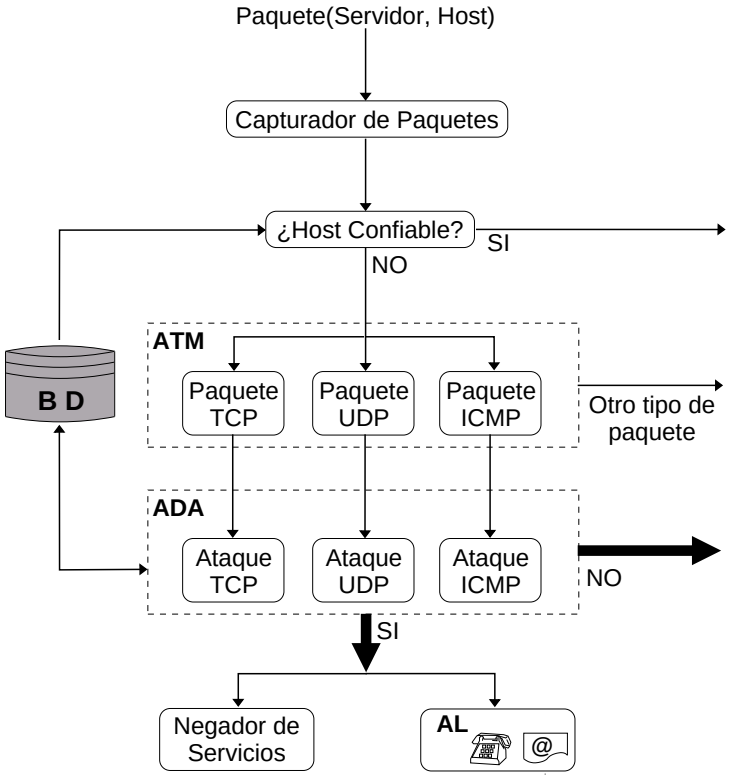

Figura 3: Arquitectura DEP

tráfico propio de la computadora y el que viaja por la red con destino a otras computadoras.

- *Analizador de Confiabilidad (AC)*: Recibe los paquetes desde CP y si determina que puede ser sospechoso, con ADA. Su tarea es determinar si el paquete que arriba a la computadora pertenece al conjunto de *"clientes confiables"*.

  Debido al costo involucrado en el análisis de cada paquete y a su posible influencia negativa en los tiempos de atención de servicios, definir un conjunto de clientes de confianza para la computadora servidor permite evitar el análisis de todos los paquetes que arriban. Todo paquete cuyo emisor está incluido en el conjunto de clientes confiables no será tenido en cuenta por DEP y en consecuencia no es analizado.

  El conjunto de clientes confiables es definido por el usuario administrador del sistema, no necesariamente todas las computadoras de la red tendrán el mismo conjunto de clientes confiables, éste dependerá de las funciones a las que está dedicada la computadora particular y principalmente a la política de seguridad definida para la red.

- *Analizador de Tipo de Mensaje (ATM)*: este módulo determina que tipos de paquetes serán analizados, dependiendo de la configuración realizada por usuario administrador. Interactúa con los módulos AC y ADA. Cuando se desea controlar la exploración de puertos TCP, UDP, ICMP y derivadas, ATM crea un socket de tipo IPROTO_TCP para capturar paquetes TCP, un socket de tipo IPROTO_UDP para capturar paquetes UDP y/o un socket de tipo IPROTO_ICMP para capturar paquetes ICMP. Dependiendo del tipo de paquete

capturado es el tipo de análisis a realizar por el módulo *ADA*.

- *Analizador y Detector de Ataques (ADA)*: Recibe información desde el módulo *ATM* y, si es necesario, envía información a *NS* y *AL*.

  Dependiendo del tipo de paquete y la información de los accesos realizados por el cliente y registrados en el historial, *ADA* determina si se trata de un posible ataque. En caso de obtener una respuesta positiva, es decir existe la sospecha de estar en presencia de un ataque, envía la información a los módulos *NS* y *AL*. Caso contrario descarta el paquete y no realiza ninguna acción.

- *Negador Servicios (NS)*: en el caso que el modulo *ADA* determine que el nodo cliente este realizando un posible ataque, el módulo aprovecha las características brindadas por las herramientas de "control de acceso" para deshabilitar todos los servicios al cliente.

- *Alertas (AL)*: además si el módulo *ADA* detecta un posible ataque, solicita emitir una alerta al usuario administrador, informándole por medio de un mensaje electrónico los datos del hecho: identificación del atacante, hora, puerto, entre otros.

El análisis de cada paquete recibido por la computadora, toma un tiempo considerable. Además, si la computadora brinda un elevado número de servicios, *ssh, ftp, http, smtp*, etc., el flujo de paquetes a analizar por *DEP* puede ser excesivo, degradando el desempeño del servidor. Por esta razón es necesario evaluar los costos en función de los beneficios provistos por la herramienta. Sin embargo *DEP* incluye mecanismos que tienden a reducir su costo, ejemplo de ello es el conjunto de clientes confiables para cada una de las computadoras de la red.

4.2. Detalles de Implementación

El sistema *DEP* tiene dos ambientes de ejecución: un ambiente de consola de texto y un ambiente gráfico. El sistema se implementó empleando el lenguaje de programación *C* [1]. La interfase gráfica fue desarrollada utilizando la biblioteca gráfica *QT* versión 3.3.3 [6].

El historial de accesos remotos producidos por computadoras "no confiables", administrado por el módulo *ADA*, es realizado mediante una base de datos relacional administrada por MySQL[8] [18]. La base de datos contiene las siguientes relaciones:

- HOST_ALLOW: Conjunto de host "confiables" para el sistema.

- IP_TABLE: Conjunto de posibles exploradores de puertos, es decir todo cliente que realizó una actividad sospechosa.

- PORT_TABLE: Conjunto de puertos accedidos en una exploración.

- TIME_TABLE: Historia de accesos sospechosos a los puertos.

La exploración de puertos se realiza mediante la herramienta *NMAP*, la cual permite explorar varios puertos al mismo tiempo.

La negación de servicio se realiza a través de *TCP-Wrapper*. En la siguiente sección se explica brevemente su funcionamiento.

4.2.1. TCP Wrapper

TCP Wrapper [7][10] es una herramienta capaz de monitorear y filtrar todo los requerimientos de servicios de red realizados al sistema desde computadoras remotas, tales como: *systat, finger, ftp, telnet, rlogin, rsh, exec, talk*, etc. El programa TCP Wrapper es invocado por el "Super Servidor de Internet".

El "Super Servidor de Internet", *inetd*, aguarda por el arribo de cada solicitud de servicios de red, evitando mantener en ejecución un programa por cada servicio. Cada vez que un cliente inicia una conexión para solicitar un servicio, es atendido por *inetd*, el cual invoca al programa TCP Wrapper, también llamado "Control de Accesos a los Servicios de Internet". Un vez que *inetd* deriva a TCP Wrapper una conexión, queda a la espera de nuevas solicitudes.

TCP Wrapper registra el requerimiento y realiza algunos chequeo. Si todo está correcto, pasa el control al servidor apropiado, encargado de atender el servicio solicitado por el cliente. Luego de la delegación de las funciones, continua aguardando conexiones derivadas de *inetd.*

TCP Wrapper ofrece las siguientes características:

- Loggin: monitorea las conexiones, registrándolas en los archivos de *logs* del sistema.
- Control de accesos: soporta una forma simple de control de accesos basada en la unificación de patrones.
- Verificación del nombre del host: para cada conexión, verifica si coincide el nombre con la dirección IP del host cliente. Para ello realiza consultas al servidor de DNS a partir del nombre (nombre_host→dirección_host) y de la dirección (dirección_host→nombre_host).
- Falseado de Nombre o dirección del host: ofrece protección contra aquellos hosts que ocultan su verdadera dirección o nombre y se hacen pasar por otro. Si existe alguna discrepancia en la dirección o nombre del cliente, niega el acceso al host cliente.

TCP Wrapper consulta los archivos de control de acceso **/etc/hosts.allow** y **/etc/hosts.deny**. Estos archivos de control de accesos son usados para realizar la comparación de la dirección IP del cliente y determinar en cada nuevo requerimiento de servicio las acciones a seguir: negar o permitir el acceso al cliente.

Cada archivo de control de acceso consiste de un conjunto de reglas de control para cada uno de los diferentes servicios. Una regla de control de acceso tiene la siguiente forma:

<lista_servicios>:<lista_clientes>:<opción>:<opción>:...

Donde <lista_servicios> contiene la lista de servicios de red y <lista_clientes> es la lista de clientes para los cuales esta regla es aplicable.

El módulo de control de accesos lee el archivo **/etc/hosts.allow** y el archivo **/etc/hosts.deny** antes de permitir o denegar el acceso a cualquier servicio. Los archivos son consultados en el siguiente orden:

1° Si el par *(servicio, cliente)* coincide con una entrada en el archivo **/etc/hosts.allow**, TCP Wrappper permitirá el acceso al servicio.

2° Si el par *(servicio, cliente)* coincide con una entrada en el archivo **/etc/hosts.deny**, TCP Wrapper negará el acceso al servicio.

3° Si el par *(servicio, cliente)* no coincide con ninguna entrada en alguno de los archivos de control, TCP Wrapper permitirá el acceso al servicio.

Para prevenir posibles ataques a uno o más servicios de red disponibles, *DEP* emplea las utilidades brindadas por TCP Wrapper: Cada vez que una posible exploración de puertos desde *nombre.host-cliente* es detectada, una regla de control de acceso es insertada en el archivo **/etc/hosts.deny** con la siguientes características:

ALL : nombre.host-cliente

Con ella se niega el acceso de la computadora *nombre.host-cliente* a todos los servicios de red, evitando tomar ventaja de las vulnerabilidades propias de alguno de ellos.

5. Facilidades de *DEP*

Si bien existen diferentes tipos de exploración de puertos, *DEP* detecta eficientemente las exploraciones: *TCP CONNECT, TCP SYN, TCP FIN, TCP XMAS, TCP NULL, UDP* e *ICMP*. Actualmente se están incorporando otras.

Respecto a la interfaz del sistema *DEP*, está compuesta de una única pantalla, desde la cual es posible acceder a todos los recursos y posibilidades de configuración que brinda, mediante el empleo de las siguientes pestañas:

DetectPortScan: Muestra, en tiempo real, el reporte de los intentos de conexión realizado desde hosts remotos. Cada línea del reporte informa del tipo de exploración que se está empleando, el nombre (si es posible determinarlo) e IP del hosts que realiza la conexión. Finalmente muestra el puerto al cual se conecta. Ver figura 4.

Modo: Corresponde a la configuración del sistema *DEP*. En ella se pueden configurar las técnicas de ataque que se desean detecta, tales como exploraciones *TCP, UDP* y *ICMP*. Además permite configurar el conjunto de "clientes confiables" para el sistema, agregar un nuevo cliente, eliminar uno existente o ver la lista completa. Ver figura 5.

Reporte: Muestra la información registrada por *DEP* durante la corriente ejecución en la base de datos. Permite ver el listado de los hosts atacantes, de los puertos explorados y de aquellos explorados por un determinado origen. Los listados son mostrados en orden cronológico.

Mail: Permite configurar el modulo de alerta del sistema *DEP*, actualmente sólo se encuentra implementado el sistema de alerta por mensajes de correo electrónico. Para ello desde esta pantalla es posible configurar la dirección del destinatario, el asunto y el texto del mensaje a ser enviado cada vez que se detecte una posible exploración de puertos.

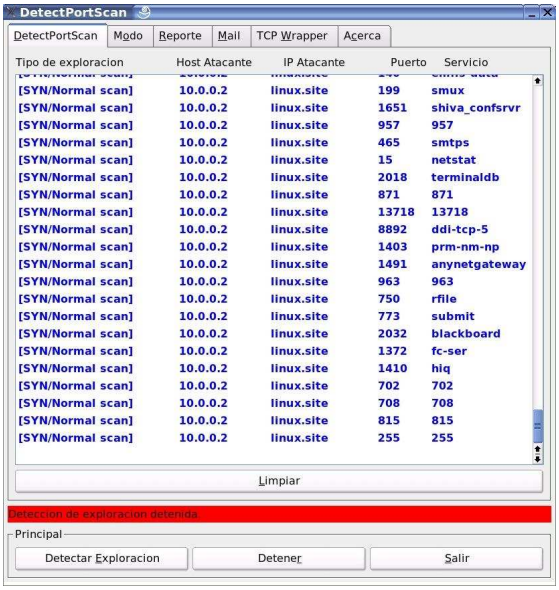

Figura 4: Pantalla de reporte del sistema *DEP*. Pantalla de detección de una posible exploración de puertos.

TCP Wrapper: Permite configurar las acciones de negación de servicios a llevar a cabo cada vez que una posible exploración de puertos es detectada. Permite: realizar una copia de seguridad del actual archivo **/etc/hosts.deny** y restaurar desde una copia de seguridad anterior. Además es posible configurar el número de intentos de conexión realizado por una computadora para se considerada como una posible exploración de puertos y establecer las acciones para evitar posibles ataques.

Todas las configuraciones y reportes descriptos anteriormente se encuentran disponibles también en el modo consola de texto.

6. Conclusiones

La exploración de puertos es una de las técnicas más comunes para acceder y obtener información desde una computadora conectada a una red. Si bien existen varios trabajos realizados en este ámbito, el desarrollo del sistema *DEP* permite no sólo proveer a un administrador de redes Linux de una herramienta capaz de detectar y reaccionar ante un intento de acceso no autorizado, sino también de analizar las características y vulnerabilidades propias del sistema operativo y de la familia de protocolos de comunicación utilizada, protocolo TCP/IP.

DEP es un sistema simple, fue desarrollado combinando diversas herramientas: *QT*, *MySQL*, *TCP Wrapper*. Brinda la posibilidad de trabajar en un ambiente de texto como en uno gráfico. Permite al administrador de redes definir y/o modificar una política de seguridad respecto a la exploración de puertos de manera sencilla, a través de la configuración de parámetros bien definidos.

Si bien *DEP* funciona correctamente en las exhaustivas pruebas a las que fue sometido, tiene varios puntos a optimizar, entre ellos están la disminución del overhead implicado en el análisis y la administración del historial. Un punto que merece

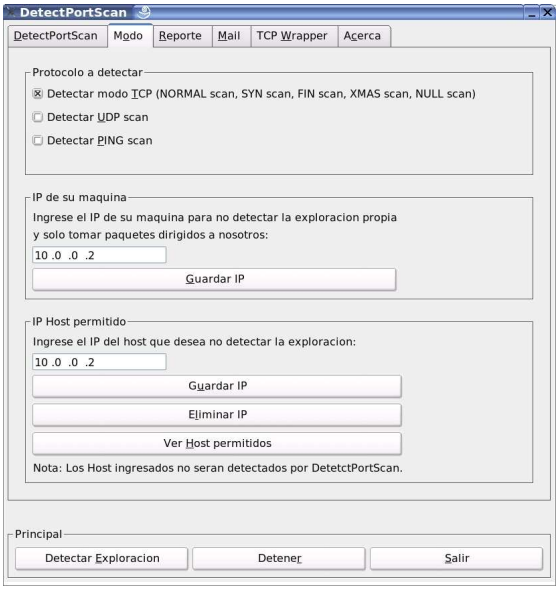

Figura 5: Pantalla de configuración del sistema *DEP*. Configurar las exploraciones de puertos a ser detectadas.

especial atención es la actitud a seguir ante una exploración de puertos, pasiva o activa.

Finalmente, una tarea a realizar es el análisis de factibilidad y portabilidad de *DEP* a otros sistemas operativos y protocolos de comunicación.

Referencias

[1] American National Standard Institute Inc New York, ANSI. *American National Standards for Information Systems Programming Language C*. ANSI X3. Pp 159-198. 1990.

[2] Beck, M., Bohme, H., Dziadzka, M., Kunitz, U., Magnums, R., Verworner, D.. *Linux Kernel Internal* - Second Edition. Addison-Wesley - 1998 - ISBN: 0-201-33143.8

[3] Comer, D. E.. *Computer Networks and Internet* - Second Edition - Prentice Hall - 1999 - ISBN: 0-13-083617-6

[4] Comer, D. E.. *Internetworking with TCP/IP, Protocols, and Architecture*. Second Edition. Prentice Hall. ISBN: 0-13-474222-2. 1991.

[5] Comer, D. E.. *Internetworking with TCP/IP Principles, Protocols and Architecture*. Prentice Hall - ISBN: 0-13-468505-9.

[6] Dalheimer, M. K.. *Programming with Qt (2nd Edition)*. O'Reilly. ISBN: 0596000642. 2002.

[7] Drake, J.. Linux Networking HOWTO. Commandprompt, Inc - 2000.

[8] Elmasri, R. A., Navathe, S. B.. *Fundamentos de sistemas de Bases de Datos*. Adison Wesley, 2000.

[9] Glass, G. *UNIX For Programmers and Users A Complete Guide*. Prentice Hall - ISBN: 0-13-480880-0

[10] Hewlett-Packard Company. *TCP Wrappers Release Notes First Edition*. 2001.

[11] Insecure. *Nmap Security Scanner*. http://www.insecure.org/nmap.

[12] Matthew, N., Stones, R.. *Beginning Linux programming*. Primera edición. Wrox.

[13] Scambray, J. , McClure, S., Kurtz, G.. *HACKER: Secretos y soluciones para la Seguridad de Redes*. McGraw Hill. 2001.

[14] Stallings, W.. *Data and Computer Communications*. Fourth Edition. ISBN: 0-02-415441-5. 2000.

[15] Siles Peláez, R., *Análisis de seguridad de la familia de protocolos TCP/IP y sus servicios asociados, primera edición*. O'Reilly & Associates Inc. 2002.

[16] Stevens, W.R.. *UNIX Network Programming, Volume 1, Second Edition: Networking APIs: Sockets and XTI*. Prentice Hall. ISBN 0-13-490012-X. 1998

[17] Tanenbaum, A. S.. *Computer Networks*. Fourth Edition, Prentice Hall - 2002 - ISBN: 0-13-066102-3.

[18] Welling, L., Thomson, L.. *MySQL Tutorial, First Edition*. MySQL Press. ISBN: 0-672-32584-5. 2004.

[19] Wolfgang, M.. *Host Discovery with nmap*. DOE Computer Security 2003. Department of Energy. U.S. Baltimore. November 2003.