UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

"ANÁLISIS FORENSE PARA SISTEMAS OPERATIVOS LINUX"

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO EN COMPUTACIÓN

PRESENTAN:

OSWALDO GÓMEZ GALLARDO
VÍCTOR HUGO SÁNCHEZ QUIJADA

DIRECTOR DE TESIS:

M. EN C. MA. JAQUELINA LÓPEZ BARRIENTOS

CIUDAD UNIVERSITARIA MÉXICO, D.F. 2004

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

"ANÁLISIS FORENSE PARA SISTEMAS OPERATIVOS LINUX"

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO EN COMPUTACIÓN

P R E S E N T A N:

OSWALDO GÒMEZ GALLARDO
VÍCTOR HUGO SÁNCHEZ QUIJADA

DIRECTOR DE TESIS:

M. EN C. MA. JAQUELINA LÓPEZ BARRIENTOS

Ciudad Universitaria México, D.F. 2004

A la Universidad Nacional Autónoma de México, UNAM.

Por haberme forjado como profesionista, por ser una fuente inagotable de conocimiento, de superación, y por darme la oportunidad de crecer.

A mis Padres.

Por su gran apoyo incondicional, su infinita paciencia, por todas sus lecciones que me acompañarán toda la vida, este logro es suyo y en mi trasciende su amor. Gracias.

A mis Hermanos Amelia, Eduardo, Wulfrano, Perla.

Ustedes representan para mi un gran apoyo y una amistad desinteresada que supera las circunstancias y el tiempo, todos son para mí un ejemplo a seguir y eso me hace sentir orgulloso. Gracias por su amor y su amistad.

A mis Sobrinitas Perlita y Carlita.

Siempre me han hecho pasar buenos momento, me dan la oportunidad de escaparme de la tensiones del día y hacen que en mi florezca otra vez mi niñez, por compartir juegos y pleitos conmigo, este trabajo es de su "Tío" para ustedes. Gracias.

A Liliana.

Dedico este trabajo a ti porque creíste en mí, tú me has apoyado y acompañado en este camino, siempre estas en mi corazón porque me enseñaste el significado de la palabra amor, eres muy especial e importante en mi vida. Gracias por todo "Lili", te amo.

A Víctor Hugo.

Gracias por tu apoyo para este trabajo, por tu tiempo, tu dedicación por tu amistad y tu paciencia, por no dejar de luchar para conseguir esta meta, seguimos siendo un buen equipo que cuando se propone algo lo logra. Gracias Vic.

Quiero hacer una agradecimiento especial a Margarita, por todo su apoyo para este trabajo, por su amistad y compañerismo. Sin tu ayuda este logro hubiera sido difícil de conseguir. Gracias.

A la Maestra Ma. Jaquelina López Barrientos.

Gracias por su gran e invaluable apoyo, por sus conocimientos y sus buenos consejos, por creer y confiar en nosotros para conseguir esta meta, reciba este logro que es suyo.

A mis amigos imborrables.

El tiempo son como las olas que se llevan todo y solo dejan lo autentico, lo que perdura, les agradezco a ustedes porque a pesar de todos estos años siguen siendo mis amigos y cuando los vuelvo a ver parece que no ha cambiado nada y el tiempo no hubiera pasado, gracias por estar siempre conmigo incondicionalmente, Gilberto, Víctor, Ernesto, Héctor y Gabriel.

A mis compañeros de Batalla.

Ustedes son una inspiración para superarme todos los días, gracias por esas horas de trabajo, esfuerzo y conocimiento, por compartir sus inquietudes y vivir un mismo sueño: el ser mejores. Su ejemplo siempre será mi guía, su amistad mi apoyo, este trabajo también es de ustedes: José Armando, Víctor Durán (el "Master"), Silvia Ramírez, Oscar Iván y Juan Gabriel ("el Mai").

Oswaldo Gómez Gallardo.

Quiero dar las gracias a mis queridos papás. A mi mamá por sus sabios consejos, apoyo incondicional y porque siempre a estado a mi lado en las buenas y en las malas. A mi papá por orientarme por el buen camino, apoyándome siempre que lo necesito y sobretodo por comprenderme. A mis papás les dedico este trabajo porque gracias a ellos soy lo que soy, porque su esfuerzo y dedicación han logrado que su hijo termine una etapa más en su vida. Padres míos, los amo con todo mi corazón.

A mi tía "la china" por brindarme siempre su cariño, por consentirme y estar al lado de mis hermanos y mío. Te dedico este breve trabajo como parte del gran amor y agradecimiento que tengo tía linda. Siempre te llevaré en mi corazón.

A mis queridos hermanos, porque en todo momento me han apoyado y aconsejado dando un mayor apoyo a mi formación. Les dedico la tesis por ser los mejores hermanos del mundo, porque siempre me han demostrado su amor y cariño hacia mí, aunque estemos en distintos lugares, juntos permaneceremos. Gracias Fer y Enrique.

A Maguito, por ser tan linda conmigo. Por demostrarme su amor y comprensión. Apoyarme en el desarrollo de mi tesis y darme valiosos consejos en el transcurso de la misma. Quiero dedicarte mi tesis por ser una gran mujer, ser mi novia y la persona a la que amo. Te agradezco cariño mío.

A la maestra Jaquelina por todas sus enseñanzas que me ha dado. Porque es una persona que apoya incondicionalmente y siempre busca la formar personas con un alto grado de profesionalismo. Gracias querida Maestra.

A Oswald, mi amigo Oswald, porque confió en mi al poder desarrollar juntos este trabajo. Gracias Oswald, no voy a olvidar todos los momentos que pasamos en clases y en UNICA. Todos los sufrimientos que tuvimos que pasar para sacar las cosas y que juntos logramos.

Y a todos los demás, Salud!!!

Víctor Hugo Sánchez Quijada.

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

Organización del documento

En este trabajo de tesis iniciamos con la definición de seguridad informática así como su importancia; también mencionamos los tipos de vulnerabilidades, amenazas, ataques y un análisis de los problemas de seguridad que establecen el escenario en el cual el análisis forense en Linux va a encontrarse. Para ello definimos qué es el análisis forense y su importancia en el ámbito de la seguridad informática. Por otra parte, exponemos lo imprescindible de la aplicación del análisis forense en la Facultad de Ingeniería, de esta manera, finalmente establecimos los alcances y objetivos de la tesis.

Posteriormente tratamos a fondo el marco teórico en el cual nuestro trabajo se basa, comenzamos a hacer una remembranza de la historia de los incidentes de seguridad en el mundo, del FIRST (Forum of Incident Response and Security Teams) y los CERT's (Computer Emergency Response Team) nacionales e internacionales así como su definición, todo esto con el fin de terminar de mencionar el entorno y problemática en el que se desempeña la ciencia forense. Definimos los conceptos más fundamentales así como críticos del análisis forense, es decir, los pasos para recolectar la evidencia, la cadena de custodia y las dificultades del investigador forense, por último mencionamos la aportación más importante de nuestro trabajo, la metodología para realizar el análisis.

A continuación explicamos a fondo las tecnologías que utilizamos en nuestros casos prácticos, como son los IDS (Sistemas de Detectores de Intrusos), las HoneyPots y las HoneyNets; también explicamos de manera exhaustiva cada uno de los puntos de la metodología forense: captura de evidencia, análisis de la imagen capturada, análisis del sistema cliente, estructura de archivos binarios y la teoría de la evasión forense.

Ya definida perfectamente la metodología así como las tecnologías a utilizar, nos dimos a la tarea de realizar un análisis del entorno de seguridad del laboratorio de Redes y Seguridad de la Facultad de Ingeniería, que fue donde se desarrolló esta investigación.

Posteriormente construimos bajo un medio perfectamente controlado tres casos prácticos en los cuales probamos nuestra metodología. Básicamente estas prácticas, de acuerdo a los recursos del laboratorio y a la seguridad implícita en ellas, consisten en tres diferentes configuraciones de servidor lo más representativas posibles a los de la Facultad, posteriormente fueron comprometidos y analizados con la metodología forense propuesta.

Finalmente, realizamos una evaluación y un estudio profundo de los resultados obtenidos por la metodología, definimos la contribución del análisis forense a la seguridad en cómputo, redefinimos las fronteras de la ciencia forense e hicimos expectativas a futuro, esto último conforman las conclusiones de nuestro trabajo.

CAPÍTULO I

INTRODUCCIÓN

CAPÍTULO I INTRODUCCIÓN

La información[1] es uno de los activos más importantes de las entidades, y de modo especial en algunos sectores de la vida pública o privada de cada nación. Es indudable que cada día las entidades dependen en mayor medida de la información y de la tecnología frente a la realidad de hace pocas décadas. Por otra parte, hace unos años la protección era más fácil, con arquitecturas centralizadas y terminales no inteligentes, pero hoy día los entornos son realmente complejos, con diversidad de plataformas y proliferación de redes, no sólo internas sino también externas, e incluso con enlaces internacionales.

Todo indica que este crecimiento continuará durante muchos años más. Al aumentar el interés en Internet, también ha aumentado el interés en el sistema operativo UNIX, que es el favorito para las compuertas de Internet, para las computadoras de investigación de alto rendimiento y para las plataformas educativas avanzadas.

Linux, un tipo de UNIX, es ahora utilizado en muchas de las PC's de la Facultad de Ingeniería, no sólo se usa en servidores sino como sistema operativo para escritorio. El usuario comúnmente no le da mucha importancia y el interés suficiente para darle un mínimo de seguridad a sus computadoras personales. Por ende, la seguridad informática juega un papel importante dentro de la Facultad de Ingeniería.

La seguridad[2] se hace más indispensable en los servidores, de tal forma que se analicen los ataques, los hechos y los medios en que se perpetúan los mismos, es por eso que el objetivo de un análisis forense informático es realizar un proceso de búsqueda detallada para poder reconstruir el log de acontecimientos que tuvieron lugar desde el momento en el que el sistema estuvo en su estado integro hasta el momento de detección de un acceso no autorizado.

Al contar con un Laboratorio de Redes y Seguridad en la Facultad de Ingeniería, podemos poner en práctica los conocimientos del análisis forense en un ambiente controlado para que de esta forma se tenga una referencia teórico-práctica que se pueda aplicar en la Facultad en general cuando ocurra algún incidente de seguridad.

[1] Ver Glosario
[2] Ver Glosario

Últimamente no es extraño el "asalto" a un sistema informático, y en determinadas circunstancias puede que el afectado necesite conocer quién, cómo, desde dónde, etc., se ha accedido a dicho sistema, para ponerlo en manos del "Comité de Seguridad Informática de la Facultad de Ingeniería".

1.1 Seguridad Informática

En realidad es un concepto cuya definición exacta es difícil de proporcionar, debido a la gran cantidad de factores que intervienen en ella. Sin embargo, nosotros podemos definir en términos generales que la seguridad informática es el conjunto de recursos (metodologías, planes, políticas documentos, programas y dispositivos físicos) encaminados a lograr que los recursos de cómputo estén protegidos, de tal manera que siempre se mantengan íntegros, de forma confidencial y disponible. Por lo que sólo la persona indicada puede acceder a la información correcta en cualquier momento.

Para proteger nuestra información es necesario conocer los servicios de seguridad, los cuales también se pueden tomar como requerimientos al plantear políticas de seguridad dentro de la institución. Valorar los servicios de seguridad de acuerdo a la función que cumple y la importancia de la información se le dará mayor peso a uno u a otro.

Confidencialidad

Entendemos por confidencialidad el servicio de seguridad que asegura que la información no pueda estar disponible o ser descubierta por personas, entidades o procesos no autorizados[3]. Por ejemplo, en áreas de seguridad gubernamentales el secreto asegura que los usuarios pueden acceder a la información que les está permitida en base a su grado de autoridad. En entornos de negocios, la confidencialidad asegura la protección en base a disposiciones legales o criterios estratégicos de información privada, tal como datos de las nóminas de los empleados, documentos internos sobre estrategias, nuevos productos o campañas, etc. En cuestiones de una institución educativa

[3] Cinthia Reyes Quezada y Sergio Rodríguez Gutiérrez, "Tesis: FUNDAMENTOS DE SEGURIDAD DE LA INFORMACIÓN"(Licenciatura en Ingeniería en Computación). México, D.F., Universidad Nacional Autónoma de México, Facultad de Ingeniería, 2001.

aseguramos que proyectos, investigaciones, estadísticas y todas las cuestiones académicas de dicha institución sean accedidas por las personas autorizadas ha hacerlo.

La confidencialidad se divide en dos partes:

- Confidencialidad de contenido

  Nada ni nadie pueda leer, copiar o descubrir información sin la autorización correspondiente.

- Confidencialidad de flujo

  Nada ni nadie puede interceptar los canales de comunicación ni realizar análisis de tráfico.

Autenticación

Este servicio permite asegurar el origen de la información, la identidad del emisor puede ser validada, de modo que podemos demostrar que es quien dice ser[4]. De este modo se evita que un usuario envíe información haciéndose pasar por otro. La forma más popular de autenticación es una firma la cual se usa para autenticar al titular de una cuenta, de un tarjeta, etc. De manera que la firma regularmente se emplea tanto para autenticar como para autorizar.

No repudio

Este servicio nos permite asegurar que cualquier entidad que envía o recibe información, no puede alegar ante terceros que no la envió o la recibió[5]. Esta propiedad y la anterior son especialmente importantes en el entorno bancario y en el uso del comercio digital.

[4] Véase la nota 3.
[5] Véase la nota 3.

Integridad

Entendemos por integridad el servicio de seguridad que garantiza que la información es modificada, incluyendo su creación y borrado, sólo por el personal autorizado, es decir, que el sistema no debe modificar o corromper la información que almacene, o permitir que alguien no autorizado lo haga[6]. Esta propiedad permite asegurar que no se ha falseado la información. Por ejemplo, los datos recibidos o recuperados son exactamente los que fueron enviados o almacenados, sin que se haya producido ninguna modificación, adición o borrado.

Con esto tratamos de verificar que el origen de los datos es el correcto, quién los envió y cuándo fueron enviados y recibidos. En el entorno financiero o bancario, este aspecto de la seguridad es el más importante. En los bancos, cuando se realizan transferencias de fondos u otros tipos de transacciones, normalmente es más importante mantener la integridad y precisión de los datos que evitar que sean interceptados o conocidos. En el campo de la criptografía hay diversos métodos para mantener la autenticidad de los mensajes y la precisión de los datos recibidos. Se usan para ello firmas añadidos a los mensajes en origen y comprobadas en el destino. Este método puede asegurar no sólo la integridad de los datos sino la autenticidad de la misma.

Los servicios de integridad se refieren al control que se tenga sobre los datos a fin de asegurar que el contenido de la información no haya sido modificado y que la secuencia de éste se mantenga durante la transacción.

Disponibilidad

Entendemos por disponibilidad el grado en que un dato está en el lugar, momento y forma en que es requerido por el usuario autorizado. Por lo tanto, es la situación que se produce cuando se puede acceder a un sistema en un periodo de tiempo considerado aceptable[7]. Un sistema seguro debe mantener la información disponible para los usuarios. Este servicio de seguridad significa que el sistema, tanto hardware como software, se mantienen funcionando eficientemente y que es capaz de recuperarse rápidamente en caso de fallo. Lo opuesto a disponibilidad, y uno de los posibles métodos de ataque a un

[6] Véase la nota 3.
[7] Véase la nota 3

sistema informático, se denomina denegación de servicio, que significa que los usuarios no pueden obtener del sistema los recursos deseados.

Control de acceso

Es la habilidad para limitar y controlar el acceso a los sistemas anfitriones y las aplicaciones mediante los puentes de comunicación[8].

1.2 La importancia de la seguridad informática

Se debe tener conciencia tanto de la importancia como de lo peligroso y catastrófico que puede resultar el que algo le sucede a la información, por lo que es vital que se conserve verídica, completa, no duplicada e inalterada. Es muy importante conocer su significado dentro la función informática, de forma esencial cuando su manejo está basado en tecnología moderna, para esto se debe conocer que la información:

- Está almacenada y procesada en computadoras
- Puede ser confidencial para algunas personas o a escala institucional
- Puede ser mal utilizada o divulgada
- Puede estar sujeta a robos, sabotaje o fraudes

Los primeros puntos nos muestran que la información esta centralizada y que puede tener un alto valor y los últimos puntos nos muestran que se puede provocar la destrucción total o parcial de la información, incurriendo directamente en su disponibilidad causando retrasos de alto costo.

El disponer de la información después del momento necesario puede equivaler a la no disponibilidad. Otro tema es disponer de la información a tiempo pero que ésta no sea correcta, e incluso que no se sepa, lo que puede originar la toma de decisiones erróneas.

[8] Véase la nota 3

Otro caso grave es la no disponibilidad absoluta, por haberse producido algún desastre. En ese caso a medida que pasa el tiempo el impacto será mayor, hasta llegar a suponer la no continuidad de la entidad, como ha pasado en muchos de los casos producidos.

La facilidad del manejo, la preocupación de mantener el sistema actualizado y en constante monitoreo por parte del usuario es una situación deseable, pero que no se da en la realidad, perjudicando no sólo a la misma persona y a su información sino también pone en riesgo a toda la red interna de la Facultad de Ingeniería, porque con una sola máquina vulnerada, probablemente puede desencadenar un ataque en serie paralizando a la mayor parte de los servidores de la Facultad, provocando daños a la información que reflejan pérdidas económicas, administrativas, académicas, de tiempo, fraudes al proceso de inscripción, etc.; todo esto ocasiona un daño a la Universidad en general.

1.3 Clasificación general de vulnerabilidades y amenazas

Vulnerabilidades

Vulnerabilidad es un punto del sistema que es susceptible de ser atacado o de dañar la seguridad del mismo, es decir, representan las debilidades o aspectos atacables en el sistema informático[9].

Con base en esta definición consideramos seis tipos de vulnerabilidades:

1. Vulnerabilidad física.
2. Vulnerabilidad natural.
3. Vulnerabilidad del hardware.
4. Vulnerabilidad del software.
5. Vulnerabilidad de red.
6. Vulnerabilidad humana.

[9] Véase la nota 3

1. Vulnerabilidad física

La podemos encontrar en el edificio o entorno físico del sistema. La relacionamos con la posibilidad de entrar o acceder físicamente al lugar donde se encuentra el sistema para robar, modificar o destruir el mismo. Esta vulnerabilidad se refiere al control de acceso físico al sistema[10].

Para un servidor se pueden plantear mecanismos estrictos de seguridad en este punto, pero para una computadora personal donde pueden acceder al lugar más de una persona autorizada es difícil llevar un control, para este caso, se podrían implementar medidas vía software para impedir el robo de información, el acceso a periféricos, pero aún así quedan vulnerables los medio físicos que almacenan datos de forma externa a la computadora personal, como son CD-ROM's, disquetes, cintas, hojas impresas, etc.

2. Vulnerabilidad natural

Se refiere al grado en que el sistema puede verse afectado por desastres naturales o ambientales[11]. Las vulnerabilidades pueden ser: el no contar con un espejo del sistema en otro lugar geográfico en caso de inundaciones o terremotos, no disponer de reguladores, no-breaks, plantas de energía eléctrica alterna, tener una mala instalación eléctrica de los equipos, en caso de rayos, fallos eléctricos o picos altos de potencia. Además que las instalaciones se encuentren en mal estado, no contar con un adecuado sistema de ventilación y calefacción para que los equipos trabajen en temperaturas de 18 y 21 ° C y se tenga la humedad entre el 48% y 65%. En caso de inundación, el no contar con paredes, techos impermeables y puertas que no permitan el paso del agua.

Otra vulnerabilidad lo es el no estar informado de las condiciones climatológicas locales al construir un centro de cómputo, o para tomar medidas en determinado tiempo.

Las vulnerabilidades que se pueden tener en caso de incendio son las siguientes: el área donde se encuentran las computadoras está en un local combustible o flamante; el centro de cómputo está situado encima, debajo o adyacente a áreas donde se procesen,

[10] Cristian F. Borhelio, Tesis "Seguridad Informática: Sus Implicaciones y Implementación", 2001.

[11] Véase la nota 10

fabriquen o almacenen, materiales inflamables, explosivos o sustancias radiactivas; las paredes no están hechas de material contra incendio y no están extendidas desde el suelo al techo; que no se cuente con un piso falso instalado sobre el piso real con materiales incombustibles y resistentes al fuego; que se pueda fumar en los centros de cómputo; el no contar con muebles incombustibles y cestos metálicos para papeles o se tengan materiales de plástico inflamables; el no contar con los equipos para la extinción de incendios en relación del grado de riesgo y la clase de fuego que sea posible en ese ámbito, además de no contar con extintores manuales (portátiles) y/o automáticos (rociadores); el no tener los medios para proteger el sistema de daños causados por el humo; el no contar con procedimientos planeados para recibir y almacenar abastecimientos de papel, ya que este material es por lo general el que empeora el fuego.

3. Vulnerabilidad hardware

El no verificar las características técnicas de los dispositivos junto con sus respectivas especificaciones, la falta de mantenimiento del equipo[12]. Desde el punto de vista del hardware, ciertos tipos de dispositivos pueden ser más vulnerables que otros. Así, pueden existir algunos sistemas que no cuenten con la herramienta o tarjeta para poder acceder a los mismos; adquirir un equipo de mala calidad o hacer un mal uso del mismo, tener el equipo de cómputo expuesto a cargas estáticas, etc.

4. Vulnerabilidad del software

Ciertos fallos o debilidades de los programas del sistema hacen más fácil acceder al mismo y lo hacen menos confiable[13]. Este tipo de vulnerabilidad incluye todos los bugs en el sistema operativo Linux, u otros tipos de aplicaciones que permiten atacar al sistema operativo desde la red explotando la vulnerabilidad en el sistema. Hay que tomar en cuenta que no solamente los bugs que trae por default el sistema operativo son vulnerables, también los programas hechos por el usuario son puntos débiles que se deben de cuidar. Un ejemplo claro de esto, son programas en PHP o cgi's puestos en la web; los protocolos de comunicación carecen de seguridad, existen errores en la configuración, diseño, implementación y operación de los sistemas, además que se debe

[12] Véase la nota 10
[13] Véase la nota 10

recordar que todo sistema es inseguro. Y podemos seguir nombrando amenazas a medida que se haga un análisis más a fondo.

5. Vulnerabilidad de red

La conexión de las computadoras a redes supone sin duda un enorme incremento de la vulnerabilidad del sistema[14]. Aumenta considerablemente la escala del riesgo al que está sometido, al aumentar la cantidad de gente que puede tener acceso al mismo o intentar tenerlo. También se añade el riesgo de intercepción de las comunicaciones:

- Se puede penetrar al sistema a través de la red.
- Interceptar información que es transmitida desde o hacia el sistema.

Eso en cuanto a la conexión de red, por otra parte, se tienen fallos debido a una mala estructura y diseño del cableado estructurado por no seguir ningún estándar para el diseño e implementación del mismo. Además de no contar con plantas emergentes para la red; no disponer de pisos de placas extraíbles para el cableado estructurado, entre otras.

6. Vulnerabilidad humana

Ser vulnerable a la Ingeniería Social y la Ingeniería Social Inversa, contratar personal sin perfil psicológico y ético, no tener personal suficiente para todas las tareas, el descuido, el cansancio, maltrato al personal, mala comunicación con el personal, malos entendidos, que todo el personal no cuente con sus respectivas llaves de acceso a las instalaciones, no tener servicio técnico propio de confianza, no instruir a los usuarios para no responder a ninguna pregunta sobre cualquier característica del sistema y deriven la inquietud a los responsables que tenga competencia para dar esa información, no asegurarse que las personas que llaman por teléfono son quien dicen ser, el no tener un control de acceso o acceso basado en restricciones de tiempo, no contar con guardias de seguridad, no tener un control de registros de entrada y salida de las personas que visitan el centro de cómputo, no contar con credenciales que identifiquen al personal, no tener detectores de metales o no contar con algún tipo de sistema biométrico, como: emisión de

[14] Véase la nota 10

calor, huella digital, verificación de voz ó verificación de huellas oculares; no disponer de algún sistema de protección eléctrica como: barreras infrarrojas o de microondas, detector de ultrasonido, detectores pasivos sin alimentación, sonorización o dispositivos luminosos, edificios inteligentes, etc[15].

Amenazas

Primero antes de mencionar la clasificación de las amenazas es importante definir qué es una amenaza; una amenaza es todo aquello que intenta o pretende destruir, en el aspecto informático se pueden clasificar en los siguientes tipos:

- Amenazas Humanas.
- Amenazas en el Hardware.
- Amenazas Lógicas (Software).
- Amenazas en la Red.
- Desastres.

A continuación se definirá con más detalle cada una de ellas:

Amenazas Humanas

Para este caso podemos considerar los Hackers, Crackers, Phreakers, Carding, Trashing, Gurús, Lamers o Script kidders, CopyHackers, Bucaneros, Newbie, Wannaber, Samurai, Creadores de Virus, etc[16].

Además de las anteriores podemos encontrar también la ingeniería social, la ingeniería inversa, el robo, el fraude y el sabotaje.

Ingeniería Social: Es la manipulación de las personas para convencerlas de que ejecuten acciones o actos que normalmente no realizan para que revele todo lo necesario y así superar la barreras de seguridad. Si el atacante tiene la experiencia suficiente puede

[15] Véase la nota 10
[16] Ver Glosario

engañar fácilmente a un usuario en beneficio propio. Esta técnica es una de las más usadas y efectivas a la hora de averiguar nombres de usuarios y passwords[17].

Ingeniería Social Inversa: Consiste en la generación, por parte de los intrusos, de una situación inversa a la originada en Ingeniería Social[18].

En este caso el intruso publicita de alguna manera que es capaz de brindar ayuda a los usuarios y estos lo llaman ante algún imprevisto. El intruso aprovechará esta oportunidad para pedir información necesaria para solucionar el problema del usuario y el suyo propio.

La Ingeniería Social Inversa es más difícil de llevar a cabo y por lo general se aplica cuando los usuarios están alertados acerca de ingeniería social.

Robo: Es frecuente que los operadores utilicen las computadoras de las empresas para realizar trabajos privados o para otras organizaciones y, de esta manera, robar tiempo de máquina. La información importante y confidencial puede ser fácilmente copiada. El software es una propiedad fácilmente sustraible y las cintas y discos magnéticos pueden ser fácilmente copiados sin dejar ningún rastro.

Fraude: Cada año millones de dólares son sustraídos de las empresas y, en muchas ocasiones, las computadoras han sido utilizadas para dichos fines.

Sin embargo, debido a que ninguna de las partes implicadas tienen algo que ganar, si no más bien pierden imagen, no se da ninguna publicidad a este tipo de situaciones.

Sabotaje: El peligro más temido en los centros de procesamiento de datos, es el sabotaje, la protección contra el sabotaje es uno de los retos más duros. Éste se puede realizar por un empleado o por un sujeto ajeno a la empresa.

Físicamente los imanes son las herramientas a las que se recurre aunque el dispositivo de almacenamiento este dentro de su funda de protección, el imán afecta la información, por lo que todo un historial de cintas o discos magnéticos pueden ser

[17] Véase la nota 10
[18] Véase la nota 10

destruidos sin la necesidad de entrar en ellos. También partículas de metal o gasolina pueden ser introducidos por los conductos de aire acondicionados, líneas de instalaciones eléctricas cortadas, etc.

Personal: De los robos, fraudes, sabotajes o accidentes relacionados con los sistemas, el 73% son causados por el propio personal de la organización propietaria de dichos sistemas.

El siguiente gráfico (ver figura 1.1) detalla los porcentajes clasificando los atacantes como internos y externos.

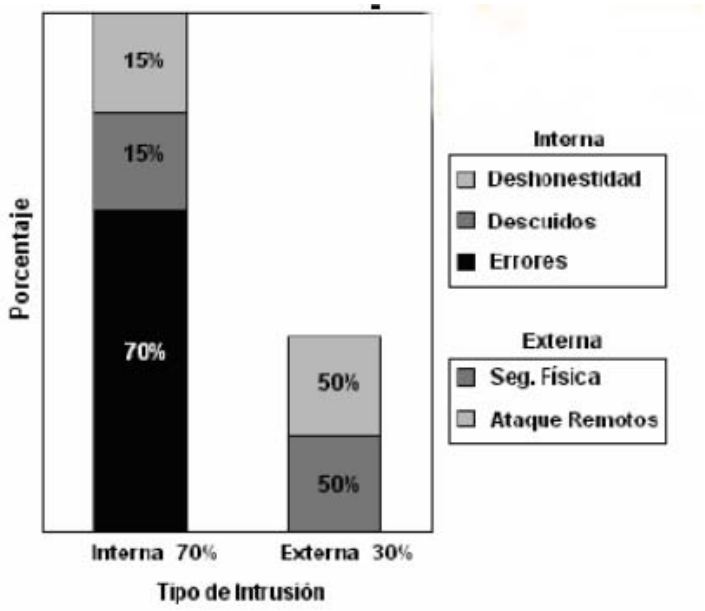


Fig. 1.1 Atacantes internos y externos

Esto es realmente preocupante, ya que una persona que trabaje con el administrador o el encargado de una máquina conoce perfectamente sus puntos débiles o fuertes, de manera que un ataque realizado por esa persona podría ser más directo, difícil de detectar y más efectivo que el que un atacante externo pueda realizar.

Existen diversos estudios que tratan sobre los motivos que llevan a una persona a cometer delitos informáticos, contra su organización, pero sean cuales sean sus motivos existen y deben prevenirse y evitarse, se suele decir que todos tenemos un precio por lo

que nos suelen arrastrar a robar o a vender información o simplemente proporcionar acceso a terceros.

Personal Interno: Son las amenazas de un sistema provenientes del personal del propio sistema informático, rara vez es tomada en cuenta porque se supone un ámbito de confianza muchas veces inexistente. Generalmente estos ataques son accidentes por desconocimiento o inexistencia de normas básicas de seguridad, pero también pueden ser de tipo intencional.

Ex-empleado: Este grupo puede estar especialmente interesado en violar la seguridad de nuestra empresa, sobre todo aquellos que han sido despedidos y no han quedado conformes o bien, aquellos que hallan renunciado para pasar a trabajar con la competencia. Generalmente son personas descontentas con la organización que conocen a la perfección la estructura del sistema y tienen los conocimientos necesario para causar cualquier tipo de daño.

Curiosos: Pueden ser los atacantes más comunes del sistema. Son personas que tienen un alto interés en las nuevas tecnologías, pero aún no tienen la experiencia ni conocimientos básicos para considerarlos hacker o crackers, generalmente no se trata de ataques de daño pero afectan el entorno de fiabilidad y confiabilidad generado en un sistema.

Terroristas: Bajo esta definición se engloba cualquier persona que ataca al sistema causando un daño de cualquier índole en él, tienen fines proselitistas o religiosos.

Intrusos Remunerados: Es el grupo de atacantes más peligroso pero también es el menos habitual. Se trata de cracker o piratas con grandes conocimientos y experiencia, pagados por una tercera parte para robar secretos o simplemente para dañar de alguna manera la imagen de la entidad atacada. Suele darse en grandes multinacionales donde la competencia puede darse el lujo de un gran gasto para realizar este tipo de contratos.

Amenazas en el Hardware

Desperfecto de los equipos, bajo rendimiento, la pérdida del mismo dispositivo físico por deterioro o incorrecto funcionamiento, pérdida total o parcial del equipo por sobrecalentamiento, problemas con cargas estáticas entre otros.

Amenazas Lógicas (Software)

Virus de Java Applets, JavaScript y VBScript, ActiveX, Backdoors, Exploits, virus ejecutables, Virus del sector de arranque, Virus Residentes, Macro virus, virus de sabotaje, Hoax o virus fantasma, Reproductores Gusanos, Caballos de Troya, Bombas Lógicas.

Amenazas en la Red

Amenazas de monitorización, Scanning, Ataques de Autenticación, obtención de passwords, Denial of Service (DoS), vulnerabilidades en los navegadores, virus del Mail, Hoax (virus fantasmas), decoy o señuelos, TCP connect Scannig, TCP SYN Scanning, Eavesdropping, Snooping Downloading, Spoofing, DNS Spoofing, WEB Spoofing, Splicing Hijacking, virus de e-mail, Caballos de Troya, uso de diccionarios electrónicos para obtener passwords, todos estos son algunos peligros potenciales en la red. Lo anterior es en el aspecto lógico pero en el aspecto físico podemos mencionar las siguientes: interferencia, cables cortados o dañados que pueden alterar la integridad de los datos, entre otros.

Desastres

Los desastres suelen ser de muchos tipos, tales como rayos o quizás más comúnmente, fallos eléctricos o picos de potencia. También podemos incluir el polvo, la humedad o la temperatura excesiva y que son aspectos importantes a tener en cuenta.

Los incendios son causados por el uso inadecuado de combustibles, fallas de instalaciones eléctricas defectuosas y el inadecuado almacenamiento o traslado de sustancias peligrosas.

El fuego es una de las principales amenazas contra la seguridad. Es considerado el enemigo número uno de las computadoras ya que puede destruir fácilmente los archivos de información y programas.

Las inundaciones son definidas como la invasión del agua por exceso de escurrimientos superficiales o por acumulación en terrenos planos, ocasionada por la falta de drenaje ya sea natural o artificial[19]. Esta también es una de las causas mayores de desastres en los centros de cómputo. También existe la posibilidad de una inundación provocada por la necesidad de apagar un incendio en un piso superior .

Los terremotos son fenómenos sísmicos que pueden ser tan intensos que causan la destrucción de edificios, de equipo y hasta pérdidas humanas[20]. El problema es que en la actualidad, estos fenómenos están ocurriendo en lugares donde no se les asociaba.

Las señales de radar sobre el funcionamiento de una computadora han sido exhaustivamente estudiados, las investigaciones más recientes demostraron que señales muy fuertes de radar pueden inferir en el procesamiento electrónico de la información pero solamente si se alcanzan los 5 Volts/metro.

1.4 Formas de ataques a los sistemas

Podríamos definir como ataques todas aquellas acciones que supongan una violación de la seguridad de nuestro sistema (confidencialidad, integridad, disponibilidad y la autenticidad). Es decir, un ataque es llevar a la acción una amenaza[21].

Dichas acciones las podemos clasificar de la siguiente manera:

Interrupción

Un recurso del sistema es destruido o se vuelve no disponible. Éste es un ataque contra la disponibilidad[22]. La destrucción o sabotaje de un elemento del hardware, como cortar una línea de comunicación, lo podemos considerar un ataque de interrupción (figura 1.2).

[19] Véase la nota 10
[20] Véase la nota 10
[21] Véase la nota 3
[22] Véase la nota 3

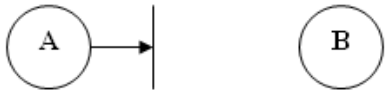


Fig. 1.2 Interrupción

Ejemplos:

Denial of Services(DoS) como: Jamming o Flooding, envío de mail para saturar las cuentas de correo de los usuarios, Connection Flood (Monopolio), Net Flood (Llamadas continuas), Land Attack, Smurf o Broadcast Storm, OOB Supernuke o Winnuke, E-mail Bombing Spamming[23].

Intercepción

Una entidad no autorizada consigue acceso a un recurso. Éste es un ataque contra la confidencialidad[24]. Este es uno de los ataques más peligrosos porque no se notan (figura 1.3).

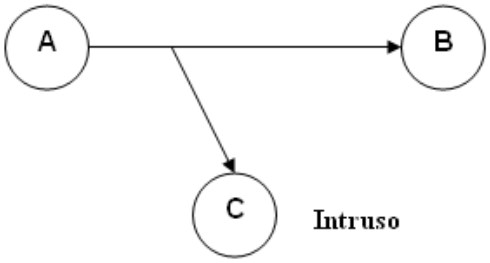


Fig. 1.3 Intercepción

Ejemplos:

SYN Flood (Saludo incompleto), EavesDroping-Packet Sniffing, TCP SYN Scanning, Scanning, TCP FIN Scanning –Sealth Port Scanning, Snooping Downloading[25].

[23] Véase Glosario
[24] Véase la nota 3
[25] Véase Glosario

Modificación

Una entidad no autorizada no sólo consigue acceder a un recurso, si no que es capaz de manipularlo. Éste es un ataque contra la integridad[26] (figura 1.4).

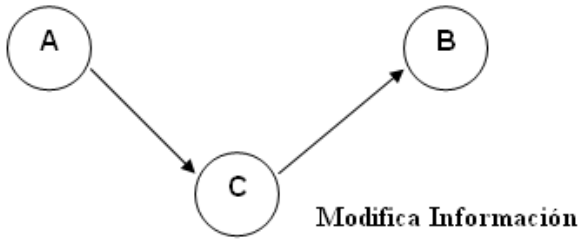


Fig. 1.4 Modificación

Ejemplos:

Teardrop I y II-Newtear-Bonk-Boink(Modificación de paquetes de red), Tampering o Data Diddling, borrado de huellas, ataques mediante Java Applets, ataques con JavaScript y VBScript, ataques mediante Backdoors, exploits, virus ejecutables, virus del sector de arranque, virus residentes, macrovirus, virus de sabotaje, Hoax o virus fantasma, reproductores gusanos, caballos de Troya, bombas lógicas, virus e-mail[27].

Suplantación.

Una entidad no autorizada inserta objetos falsificados en el sistema. Éste es un ataque contra la autenticidad[28].

[26] Véase la nota 3
[27] Véase Glosario
[28] Véase la nota 3

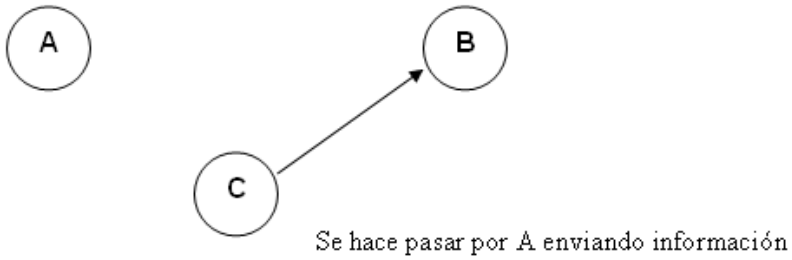


Fig. 1.5 Suplantación

Ejemplos:

Trashing(Cartoneo), Shoulder Surfing, decoy, Spoofing Looping, Spoofing, IP Spoofing, DNS Spoofing, WEB Spoofing, obtención de passwords, uso de diccionarios, IP Splicing Hijacking, Backdoors, exploits, ingeniería social inversa e ingeniería social[29].

Asimismo estos ataques se pueden clasificar en términos de ataques pasivos y ataques activos.

Ataques activos

Estos ataques implican algún tipo de modificación de los datos o la creación de falsos datos. En este caso tenemos a la suplantación de identidad, y la modificación de información.

Ataques pasivos

En los ataques pasivos el atacante no altera la comunicación, si no que únicamente la escucha o monitoriza, para obtener de esta manera la información que está siendo transmitida. Los ataques pasivos son muy difíciles de detectar, ya que no provocan ninguna alteración de los datos. Sin embargo, es posible evitar su éxito mediante el cifrado de la información y otros mecanismos. En este caso tenemos a la intercepción.

[29] Véase Glosario

Las consecuencias de los ataques se podrían clasificar en:

- Corrupción de dato: La información que no contenía defectos para tenerlos.
- Servicios denegados en el Servidor: (DoS Denial of Service): Servicios que deberían estar disponibles y no lo están.
- Leakage: Los datos llegan a destinos a los que no deberían llegar.

1.5 El problema de la seguridad informática

El auge de Internet durante los últimos 13 años ha llevado a las instituciones a utilizar esta tecnología dentro de sus procesos y actividades cotidianas tomando la comunicación vía red como algo indispensable y necesario para el desarrollo de la investigación y el traslado de la información. Sin embargo, simultáneamente, estadísticas internacionales revelan que el número de incidentes de seguridad en la red se ha incrementado de manera dramática durante los últimos años. Un estudio realizado sobre un número significativo de sitios de Internet en México pertenecientes a entidades de diferentes sectores de la vida pública nacional, nos revela que más de la mitad de ellos presentan vulnerabilidades de alto riesgo, y por lo tanto, están expuestos a incidentes de seguridad de consecuencias severas.

Desafortunadamente, a lo largo de los años hemos conocido casos en que importantes empresas y organizaciones han visto comprometidas sus operaciones debido a que han sufrido ataques perpetrados a través de redes globales, como la Internet. Estudios y análisis muestran una tendencia creciente en cuanto a incidentes de seguridad informática. Por ejemplo, CERT (*Computer Emergency Response Team*) dio a conocer un incremento impresionante en el número de incidentes de seguridad que le fueron reportados durante los años 2001 y 2002, respecto a los años anteriores. De igual forma, los resultados parciales del primer y segundo cuarto (Q1-Q2) del 2003 presagian un incremento nada alentador al término del año, como lo ilustra la figura 1.6:

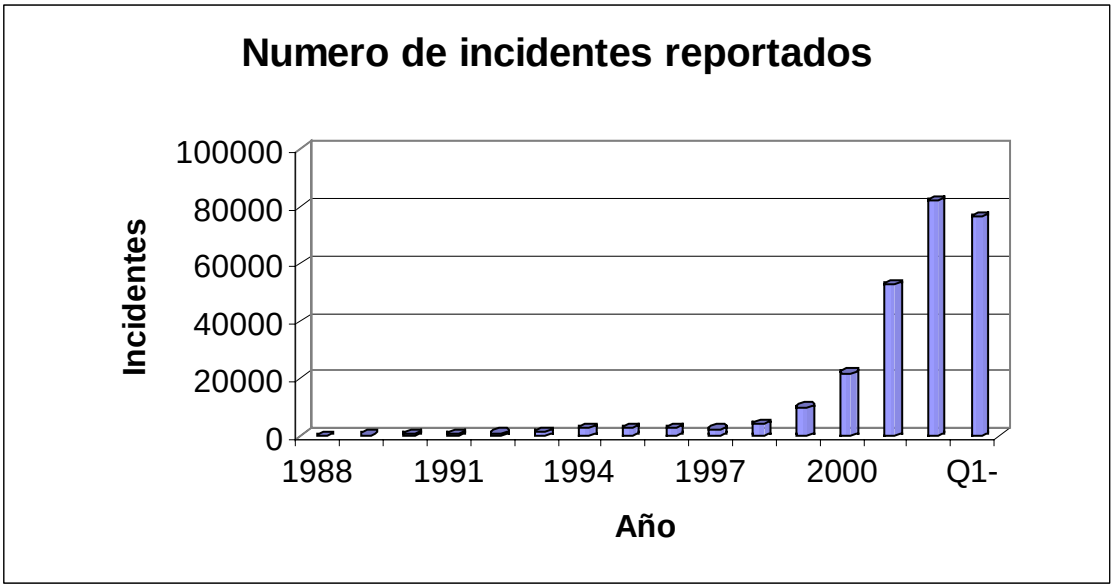


Fig. 1.6 Número de incidentes reportados

Número de incidentes reportados

1988-1989

Año	1988	1989
Incidentes	6	132

1990-1999

Año	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999
Incidentes	252	406	773	1,334	2,340	2,412	2,573	2,134	3,734	9,859

2000-2003

Año	2000	2001	2002	1Q-2Q 2003
Incidentes	21,756	52,658	82,094	76,404

Total de incidentes reportados (1988-2Q 2003): **258,867**

Una vez identificados los incidentes, la figura 1.7 detalla los tipos de ataques, las herramientas utilizadas, en qué fase se realiza el ataque, los procesos atacados, los resultados esperados y/u obtenidos y los objetivos perseguidos por los intrusos; estos ataques pueden ser realizados sobre cualquier tipo de red, sistema operativo, usando diferentes protocolos, etc.

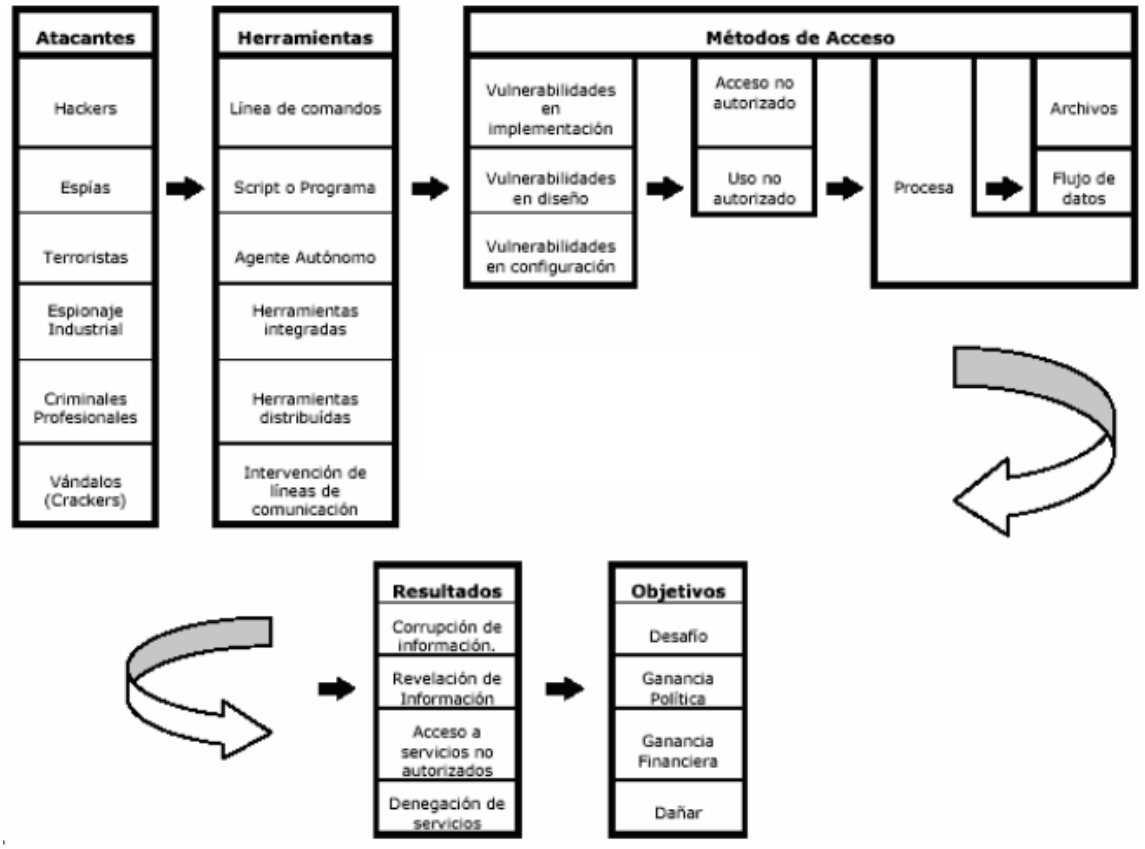


Fig. 1.7 Descripción de un ataque

En los primeros tiempos, los ataques involucran poca sofisticación técnica, los Insider (operadores, programadores, data entrys) utilizaban sus permisos para alterar archivos o registros. Los Outsiders ingresaban a la red simplemente averiguando un password válido. A través de los años se han desarrollado formas cada vez más sofisticadas de ataques para explorar agujeros en el diseño, configuración y operación de los sistemas.

Los siguientes (ver tabla 1.1 Ataques) son solo algunos términos que sirven para describir unos cuantos tipos de ataque.

Trojan horses	Fraud networks	Invalid values on calls	Combined atacks
Time Bombs	Get a job	Wiretapping	Denial of service
Brides	Dumpster diving	Eavesdropping	Degradation of service
Data diddling	Computer viruses	Data copying	Session hijacking
Login spoofing	Software piracy	Viruses and worms	Logic bombs
Scanning	Covert channels	IP spoofing	e-mail overflow
Masquerading	Excess privileges	Infrastructure observation	Human engineering
Trap doors	Fictitious people	Infrastructure interference	Packet insertion
Tunneling	Protection limit poke	Password guessing	Packet watching
Password sniffing	Sympathetic vibration	Van Eck Bugging	e-mail spoofing

Tabla 1.1 Ataques

Al describirlos no se pretende dar una guía exacta ni las especificaciones técnicas necesarias para su uso. Solo se pretende dar una idea de la cantidad y la variedad de los ataques, así como que su adaptación (y aparición de nuevos) continúa paralela a la creación de nuevas tecnologías.

Dado que el interés sobre seguridad informática es relativamente nuevo en México, pocos datos se conocen sobre el número de incidentes que se presentan en el

país. Organizaciones como el MxCERT (*Mexican Computer Emergency Response Team*) establecido en el Instituto Tecnológico de Estudios Superiores de Monterrey (ITESM) y el UNAM-CERT, de la Universidad Nacional Autónoma de México, hacen esfuerzos por promover y concienciar a la población respecto al problema de la seguridad informática en el país, pero no cuentan con estadísticas que reflejen la magnitud del mismo en México. Una de las barreras que hace difícil la obtención de estadísticas confiables, no sólo en México sino también internacionalmente, es el hecho de que las organizaciones no están dispuestas a reconocer públicamente que sus sistemas han sido comprometidos, debido a los daños que en su imagen, prestigio y consecuentemente en su economía, representaría hacerlo.

1.6 La importancia del análisis forense dentro de la seguridad informática

La seguridad en equipos de redes de computadoras sigue siendo un tema de gran relevancia en la actualidad y dentro de los diversos enfoques que se le puede dar, el análisis forense es uno de los campos que más importancia tendrá en el futuro porque uno de los principales problemas de seguridad que se plantean es la detección de las intrusiones, ya que recientes estudios demuestran que estas pasan desapercibidas en gran medida y como punto añadido es de vital importancia el saber analizar correctamente los sistemas que han sido víctimas de un ataque. La mayoría de las investigaciones están realizadas por parte de las empresas especializadas o por parte de agencias gubernamentales donde precisan un estudio forense previo para recoger todas las pruebas encontradas en los equipos y determinar los factores claves y reconstruir los hechos transcurridos, antes, durante y a posteriori del posible compromiso.

Todo ese trabajo puede ser complicado por múltiples factores, siendo una analogía directa la ciencia forense tradicional en los casos criminales, dónde la escena del crimen es el servidor comprometido y cualquier equivocación o descuido pueden causar la pérdida de información vital que podría develar algún hecho importante sobre la "víctima", el "criminal", el "objetivo" o el "móvil"; en algunos casos es imposible reconstruir el 100% de los eventos ocurridos, por ello, es importante planificar de manera correcta la detección de intrusiones, como analizar un sistema que ha sufrido un ataque y organizar correctamente un plan de respuesta a incidentes y las medidas preventivas necesarias para evitarlo.

Además, los forenses de hace varios años tienen dificultades adaptándose a las nuevas técnicas ya que no sólo son necesarios los conocimientos de la materia sino experiencia en campos que tienen muy poco que ver con la ciencia forense, ingeniería inversa, criptografía, programación en lenguajes de bajo nivel. Por otra parte, la aplicación del análisis forense nos beneficia al proteger los datos y evidencia de los equipos afectados, llevando la información ante un comité que determine los aspectos legales que se aplicarán. Además, al conocer los ataques, el equipo de análisis forense dará las medidas preventivas necesarias para evitar futuras intrusiones en nuestros equipos.

Son por estos motivos que se está haciendo más imprescindible profundizar en la ciencia del análisis forense para que de esta forma se pueda tener al día, con recursos humanos y con las herramientas más avanzadas en todas aquellas entidades donde se maneje información extremadamente importante.

1.7 La importancia de implementar el análisis forense en el Laboratorio de Redes y Seguridad de la Facultad de Ingeniería

Como ya se mencionó anteriormente, las vulnerabilidades, las amenazas y los tipos de ataque son muchos y variados, por lo que representan un gran problema para cualquier institución como lo es la Facultad de Ingeniería.

A simple vista, en esta institución, sin necesidad de hacer un análisis minucioso, se pueden identificar muchas vulnerabilidades que hacen susceptible a la Facultad de ser victima de varios tipos de amazas; históricamente, nos hemos enterado que se han registrado varios tipos de ataques informáticos, tanto físicos como lógicos, ya sea por el personal interno o externo, como robo, perdida total o parcial de la información, ataques generalmente por virus informáticos, entre otros, que afectan a la Facultad como a sus centros de cómputo. Desconocemos el número de incidentes surgidos dentro de la Facultad de Ingeniería, pero con las estadísticas observadas en la figura 1.6, con la falta de conocimiento de los usuarios acerca de seguridad informática, la ausencia de docentes con la especialización suficiente para impartir cursos sobre este tema, podemos suponer que han sido muchos los incidentes.

Aparentemente han sido muy pocas las veces que se han encontrado o castigado a los culpables, esto debido a que por lo general, no se sabe con certeza quién fue el atacante, con que herramientas o móviles perpetuó el ataque, que daño hizo.

Por otra parte, el problema de la seguridad informática en México y, por ende en la Facultad de Ingeniería, cobra especial relevancia en vista del destacado papel que se planea jugará Internet en el ámbito educativo durante los próximos años.

Son por estas razones que es importante aplicar el análisis forense en el laboratorio de Redes y Seguridad del Departamento de Cómputo de la Facultad de Ingeniería, para experimentar y probar la metodología forense en un ambiente controlado, sin comprometer ningún Servidor con información valiosa para la Facultad y, de esta forma, se pueda aplicar el análisis forense en la Facultad en general cuando exista un incidente de seguridad.

CAPÍTULO II

MARCO TEÓRICO

2.1 Historia de los incidentes de seguridad en cómputo

Estos son sin duda algunos de los incidentes de seguridad en cómputo más destacados que se han tenido a lo largo de la historia, nuestro propósito no es hacer una lista exhaustiva de todos los incidentes, ya que han sido muchos y muy variados, sino mas bien resaltar la importancia del análisis forense.

Murphy Ian en 1981 entra a los sistemas de la Casa Blanca, el Pentágono, BellSouth Corp. TRW y deliberadamente deja su currículum.

En 1987 dos hackers alemanes habían ingresado sin autorización al sistema de la central de investigaciones aeroespaciales de la NASA.

Aproximadamente el mayor de los incidentes de seguridad fue en 1989 con el gusano WANK/OILZ, un ataque automatizado en los sistemas VMS atacados por la Internet, explotando las vulnerabilidades en los programas de software libre como lo es el sendmail, un programa complicado comúnmente encontrado en los sistemas basados en UNIX para enviar y recibir correo electrónico.

En diciembre de 1992 Kevin Poulse fue acusado de robar órdenes de tarea relacionados con el ejercicio de la fuerza aérea militar americana.

En 1994 David La Macchia distribuyó en Internet multitud de programas informáticos obtenidos sin licencia y por un valor de un millón de dólares.

En ese mismo año, las herramientas para intrusos fueron creadas para "husmear" paquetes de las redes fácilmente, dando como resultado la libre distribución de los nombres de usuarios y passwords. En 1995, el método que las computadoras de Internet usan para nombrarse y autenticarse unas con otras, fue explotado por un nuevo tipo de herramientas de ataque que permiten el libre esparcimiento de ataques en Internet en computadoras que tienen una comunicación confiable con alguna otra computadora. Hoy en día utilizar los lenguajes de programación bajo WWW, crean nuevas oportunidades para los ataques en la red.

En 1995 Levin Vladimir penetró vía Internet desde San Petersburgo en los sistemas informáticos centrales del Banco Citybank en Wall Street, logrando transferir a

diferentes cuentas de EE.UU., Rusia, Finlandia, Alemania, Israel, Holanda y Suiza fondos por valor de 10 millones de dólares.

En 1997 los hackers Paint y Hagis accedieron al popular navegador de Yahoo y dejaron un mensaje amenazante a los visitantes.

En 1998 Ronald y Kevin asaltaron las computadoras del pentágono, se introdujeron a cuatro sistemas de la marina y siete de las fuerzas aéreas, relacionados con Estados Unidos y Okinawa.

El 30 de abril de 1999 Ing-Hou Chen escribe el virus Chernobyl en Taipe Taiwán, este inusual virus destructivo estaba programado para funcionar el 26 de Abril (en el 3º aniversario del desastre nuclear Chernobyl) y tratar de borrar el disco duro además de escribir basura en algunos otros componentes.

De esta manera podríamos seguir mencionando los incidentes en seguridad en cómputo.

2.2 CERT (Computer Emergency Response Team)

2.2.1 Historia de CERT

El Internet comenzó en 1969 como la ARPANET, un proyecto fundado por la Agencia de Investigaciones y Proyectos Avanzados (ARPA) en el departamento de defensa de los Estados Unidos. Uno de los primeros objetivos del proyecto fue crear una red que pudiera seguir funcionado aún si la mayoría de las secciones cayeran o fueran atacadas. La ARPANET fue designada para reorientar el tráfico en la red automáticamente cuando existan problemas en la conectividad de los sistemas o pasar a lo largo de ella, la información necesaria que se guarda en la funcionalidad de la red.

El protocolo de ARPANET (las reglas de sintaxis que permiten a las computadoras comunicarse con la red) fue originalmente diseñado para ser abierto y flexible "pero inseguro".

Durante estos años, los investigadores también jugaron bromas con otros usuarios de la ARPANET. Estas bromas usualmente eran mensajes anónimos, y otras violaciones menores de seguridad. Algunos de estos se describen en el libro de Steven Levy's, "Hackers: Héroes de la Revolución de las Computadoras". Fue raro que una conexión desde un sistema remoto se considerara un ataque, por lo tanto, los usuarios de ARPANET formaron un pequeño grupo de personas que generalmente se conocían y confiaban el uno del otro.

En 1986, la primera publicación internacional de incidentes de seguridad fue cubierta por Cliff Stoll, en el noreste de California en el Laboratorio Nacional de Lawrence Berkeley. Un simple error de conteo en los registros de cómputo del sistema de conexión a la ARPANET permitió robar un irrecuperable patrimonio internacional, usando la red para conectar a computadoras del gobierno de los Estado Unidos y copiar toda la información de ellos. No solo fueron las computadoras de los Estado Unidos, si no también, las de universidades y sitios militares por todo ese país. Cuando se publicaron estas experiencias en 1989, comenzó la advertencia de que la ARPANET podría ser usada para propósitos destructivos.

En 1988, la ARPANET tuvo su primer incidente de seguridad automatizado, povocado por el gusano Morris. Un estudiante de la Universidad de Cornell (Ithaca,NY), Robert T. Morris, escribió un programa que podía conectar a otra computadora, encontrar y usar una de sus muchas vulnerabilidades para copiarse a sí mismo hacia otra computadora y comenzar a correr copias de sí mismo en una nueva locación. Ambos, tanto el código original y las copias, pudieron entonces repetirse en esas secciones en un ciclo infinito hacia otras computadoras en la ARPANET. Esta herramienta de ataque de auto replicación automatizada, causó una explosión geométrica de copias de sí mismo que empezaban en todas las computadoras de ARPANET. El gusano usó muchos recursos del sistema ocasionando que las computadoras atacadas no pudieran funcionar más. Como resultado el 10% de las computadoras de los Estado Unidos conectadas a la ARPANET afectivamente dejaron de funcionar al mismo tiempo.

El gusano obligó a la Agencia de Investigaciones y Proyectos Avanzados de la Defensa (DARPA el nuevo nombre de ARPA) a crear un equipo de respuestas a las emergencias en cómputo conocido como el CERT Centro de Coordinaciones, para darles a los expertos un punto central parar coordinar respuestas a las emergencias en la Red.

Otros equipos rápidamente registraron sus direcciones para incidentes de cómputo en organizaciones o regiones geográficas específicas. Dentro del año de su formación, estos equipos de respuesta a incidentes crearon una organización informal ahora conocida como el Foro de Respuesta a Incidentes y los Equipos de Seguridad (FIRST). Estos equipos y las organizaciones FIRST existen para coordinar respuestas a los incidentes de seguridad en cómputo, asistencia a los ataques más destacados y educar a los usuarios de la red acerca de temas de seguridad en cómputo, dándoles prácticas preventivas.

En 1989, la ARPANET oficialmente se convirtió en la Internet y se movió de un proyecto de investigación gubernamental a una red operacional, desde entonces ha crecido a más de 100,000 computadoras. Los problemas de seguridad continúan, tanto las tecnologías agresivas y defensivas se han vuelto más sofisticadas.

La organización y operación de CERT se ha desarrollado a través de tres periodos: 1) un temprano e informal periodo de Noviembre de 1988 a junio de 1992 aproximadamente, 2) un periodo de transición por el siguiente año y medio, y 3) un periodo más formal comenzando en el verano de 1993 y que se extiende a la fecha.

1) El periodo informal noviembre de 1988 a junio de 1992: Después del incidente del gusano en Internet en noviembre de 1988, DARPA rápidamente se oriento a reestablecer al CERT para institucionalizar la capacidad de responder a los incidentes, ya que fue espontáneamente formado durante el incidente de 1988. Dentro de semanas CERT estuvo funcionando en el Instituto de Ingeniería de Software (SEI) de la Universidad de Carnigie Mellon en Pittsburg, el CERT respondía a los incidentes de una manera informal, la comunicación fue primeramente por correo electrónico y complementada por el teléfono.

Los Registros durante esta temprana etapa reflejan la resistencia del personal del CERT para esforzarse en formalizar las respuestas a incidentes, a pesar de eso hubieron continuos esfuerzos para formalizar el proceso de formulación de respuestas. El raciocinio y la gran flexibilidad para el personal del CERT, quienes entonces podían usar su propio juicio para determinar el correcto curso de la acción durante cualquier incidente, al final de esta etapa, lograron establecer por medio de la Internet, puntos de contacto con organizaciones, así como esparcir información sobre vulnerabilidades, esta información era investigada por los equipos o enviada vía correo electrónico por sus clientes o cualquier

otra persona, dicha información se almacena en bases de datos. Los equipos de respuesta estaban a la espera en línea por turno y solo uno atendía el incidente que le tocaba, con todo lo anterior lograron clasificar tres tipos de respuestas: a)respuestas para asistencia sobre algún tipo de incidente, b)información en Internet sobre las vulnerabilidades a instituciones y comercio, y c)respuesta desde Internet a los usuarios acerca de cómo reducir las vulnerabilidades e incrementar la seguridad.

2) Periodo de Transición, de enero de 1992 a septiembre de 1993. A principios de 1992 el número de incidentes creció hasta que el proceso de respuesta a incidentes ya no respondía satisfactoriamente. El personal de CERT tuvo que reorganizarse, ya que el método para guardar el registro de los incidentes era informal, como: escritos a mano, notas y correo electrónico. Los incidentes no se organizaban por números ni por sitios. Como el número de incidentes fue creciendo, también creció la dificultad para mantener ordenado el registro de los mismos y dar respuestas efectivas. Por otra parte, pasar la responsabilidad de todos los incidentes que los equipos registraban, a un equipo encargado de su registro, fue haciéndose más difícil, consumía mucho tiempo y era confuso.

A finales de 1993 pudieron ordenar los registros asignándoles un número aleatorio que lo identificaba tanto como un incidente o como una vulnerabilidad, así como mensajes aislados.

El segundo ajuste fue hacer que los incidentes no fueran atendidos por equipo si no por una persona respaldada por un equipo especializado.

3) Periodo de formación de septiembre de 1993 a la fecha. A finales de este periodo CERT ya había formalizado completamente su proceso de respuesta a incidentes. El personal responde en línea y vía correo electrónico, conocidos como coordinadores técnicos. Varias de las tareas de registro, procesamiento y organización de los incidentes fueron completamente automatizadas por sistemas de cómputo.

2.2.2 Funciones de un CERT

Las funciones del CERT son proporcionar soporte, realizar seguimiento y organizar la coordinación entre organizaciones en la resolución de incidentes de seguridad en

Internet, además, difundir información sobre problemas de seguridad, dar formación y fomentar una cultura de la seguridad

CERT/CC es ahora parte del programa de sistemas de red resistentes del Instituto de ingeniería de software (SEI Networked Systems Survivability Program), cuyo objetivo principal es el facilitar la tecnología adecuada y las prácticas de administración que se utilizan para resistir ataques a sistemas de red y limitar las pérdidas y asegurar los servicios críticos.

El Centro de Coordinación Seguro (CERT CC) trabaja en cuatro grandes áreas:

Área 1: Análisis de vulnerabilidades y gestión de incidencias: Desde 1988 han recibido más de 9.160 informes de vulnerabilidades. Interactúan con más de 450 vendedores de Hardware y software, y divulgan esas vulnerabilidades con sus recomendaciones públicamente.

Con respecto a la gestión de incidencias, han reportado más de 182.000 incidentes de seguridad de ordenadores. En la actualidad y debido al crecimiento de Internet colaboran con equipos de trabajo llamados CSIRTs, de los cuales hay ya más de 200. Al mismo tiempo han generado un sistema automatizado de confección de informes de incidencias llamado AirCERT.

Área 2: Administración de resistencia a ataques en empresas: El objetivo de este apartado es el de ayudar a que las empresas y sus responsables a nivel informático se puedan auto defender frente a ataques. Han desarrollado unas valoraciones de riesgos que ayudan a las empresas a identificar y caracterizar amenazas e información crítica. Para ello han creado un método que se llama OCTAVE (Operationally Critical Threat, Asset, and Vulnerability Evaluation).

Área 3: Formación y enseñaza: Uno de los objetivos es el de poder formar a personal dentro de las empresas que permitan dar la seguridad y resistencia a ataques de los sistemas informáticos. CERT ofrece cursos de formación para administradores de sistemas y técnicos que están interesados en aprender acerca de la seguridad de la red. Muchas de estas clases son parte de un programa de certificación para llegar a ser un CERT-Certified Computer Security Incident Handler (Encargado de seguridad de incidentes informáticos CERT).

Muchos de los miembros del equipo CERT dan formación en la especialización de Gestión de la seguridad de la información en el Master de Gestión de Sistemas de la Información en la universidad de Carnegie Mellon.

Disponiendo como base el trabajo que se realiza en CERT/CC, ellos han desarrollado unas prácticas en seguridad CERT basadas en una tecnología neutral las cuales facilitan en sus cursos. Estas prácticas las cuales sirven de ayuda a administradores de sistemas se han recopilado en un libro publicado por Addison-Wesley llamado la Guía CERT para prácticas de seguridad en redes y sistemas.

Área 4: Tecnología de red resistente a ataques: Uno de los aspectos de las investigaciones del CERT/CC están en la ingeniería de sistemas resistentes a ataques. Este trabajo incluye análisis que permiten determinar cómo de susceptibles son los sistemas hacia ataques sofisticados y formas de mejorar el diseño de los sistemas. Ellos han desarrollado una herramienta llamada Easel la cual estudia las respuestas de la red hacia ataques.

El objetivo fundamental que tiene CERT/CC es el de la divulgación de la información, la cual distribuye por diversas vías: teléfono, fax, mail, la web, grupos de noticias, bases de datos de conocimiento, etc.

Es la primera fuente de alertas en la red Internet sobre aspectos de ataques y vulnerabilidades, de aquí que tenga relación con los principales medios de comunicación a nivel nacional como internacional (Computerworld, Washington Post, Forbes Magazine, the Dow Jones News Services, USA Today, CNN, ABC News, CBS News, BBC, CNET News, etc.). ServicioHelpDesk recopila diariamente información de CERT/CC y la publica desde la web a todos sus visitantes.

Entre los objetivos del CERT se encuentran:

a) Trabajar junto a la comunidad Internet para facilitar su respuesta a problemas de seguridad informática que afecten la operación de Internet.

b) Dar soporte a los administradores y usuarios para elevar la conciencia colectiva sobre temas de seguridad informática.

c) Realizar tareas de investigación que tengan como finalidad mejorar la seguridad de las redes existentes.

d) Brindar asistencia 24 horas al día para responder a incidencias sobre seguridad informática, asistencia sobre vulnerabilidad de productos, documentos técnicos y cursos de formación.

e) Adicionalmente, el CERT mantiene numerosas listas de correo y ofrece un servidor de FTP anónimo, en ftp://cert.org donde se archivan documentos y herramientas sobre temas de seguridad informática.

CERT participa en las siguientes organizaciones:

- Forum of Incident Response and Security Teams (FIRST)
- Internet Engineering Task Force (IETF)
- Ational Security Telecomunications Advisory Committee's Network Security information Exchange (NSTAC NSIE)
- Infragard

2.2.3 Surgimiento de CERTS en el Mundo

Europa cuenta con 28 equipos de seguridad europeos, 17 de ellos miembros de FIRST, con un alto porcentaje de incidentes registrados (con respecto al resto del mundo), con un marco organizativo común como es la Unión Europea, y con ventajas como la proximidad geográfica y la afinidad cultural.

Sigue un breve resumen de este largo camino:

1992: RARE organiza el task-force CERT-TF para especificar cómo debería ser un CERT europeo. Sus trabajos durarán hasta 1994.

1993: Primera reunión de equipos de seguridad europeos (Amsterdam), para evaluar los progresos de CERT-CF y aportar sus contribuciones.

1994: Conclusión final CERT-TF.

- Se detalla cómo debe ser CERT-EU.

- Segunda reunión (Hamburgo).

  - Se presentan propuestas concretas para realizar esta tarea (RIPE-NCC, DANTE).

- Llamada a nueva reflexión. TERENA inicia proyecto ConCERT-in-E.

1995: TERENA inicia un nuevo task-force (CERTS in Europe o CERIE), en el contexto de su Grupo de Trabajo de Seguridad.

- Creación de un equipo de seguridad en RedIRIS.

- Tercera reunión (Karlsruhe), primera en que participa RedIRIS.

1996: Informe CERIE, detallando la creación de EuroCERT (aún sin nombre) en tres fases.

- Primera propuesta para gestionar este centro: no hay candidatos.

- Creación en TERENA de CERT-TAG (Technical Advisory Group).

- Segunda propuesta. Varios candidatos, finalmente es seleccionada la coalición Dante/Ukerna. Comienza oficialmente el piloto, con nombre interno SIRCE (Security Incident Response Coordination for Europe).

1997: Presentación de EuroCERT.

- Mayo: EuroCERT contrata su primer empleado y comienza su actividad.

- Julio: Presentado en FIRST (Bristol).

- Noviembre: Presentado en RedIRIS (Zaragoza)

Se preveía un EuroCERT en tres fases:

- Fase de apoyo y creación de infraestructuras: personal básico, servidores de información, listas de correo, comunicaciones seguras.

- Fase de atención de incidentes: medidas de actuación a posteriori, coordinación multilateral, atención 24 horas.
- Fase de coordinación de incidentes: incorporar también medidas preventivas, formación, análisis, investigación aplicada.

El piloto EuroCERT está operativo desde mayo de 1997. Los objetivos de la primera fase se han cumplido, y están en una fase de transición que facilite el acceso a la segunda.

El objetivo principal de este organismo es la coordinación de los IRTs (Incident Response Teams) europeos.

Los IRTs en España que forman parte del EuroCert y que actúan en coordinación con ellos son:

- El esCERT-UPC (Equipo de Seguridad para la Coordinación de Emergencias en Redes Telemáticas)
- El IRISCERT

AUSCERT (Australian Computer Emergency Response Team)

El AusCERT es similar al CERT, pero en Australia. AusCERT es miembro del FIRST (Forum of Incident Response and Security Teams). Está estrechamente relacionado con el CERT (Coordination Center), con otros "Incident Response Teams" (equipos de respuesta de incidentes) internacionales y con la Policía Federal Australiana.

Los anteriores fueron de los primeros en surgir pero también existen: CERT Español, CERT Brasileño, CERT en Polonia, CERT Holandés, CERT Danés, CERT IT (italiano), DFN CERT (alemán), SWITCH CERT (escocés), CCERT (chino), id-CERT (indonés), JP CERT (japonés), CERTCC (Coreano) entre otros.

2.2.3.1 CERT en México

Durante ocho años de trabajo e investigación, y luego de cubrir los exigentes requisitos técnicos, académicos y administrativos, establecidos por el Forum Incident Response Security Team (FIRST), www.first.org, organismo rector a nivel mundial con

sede en Chicago, Estados Unidos, la DGSCA (Dirección General de Servicios de Cómputo Académico) funda en el año 2000 UNAM-CERT y con ello se crea, en nuestro país, el primer equipo de respuesta a incidentes con reconocimiento internacional, y con respaldo del System Administration Networking Security, entidad que trabaja de cerca con el FBI.

UNAM-CERT es un equipo de especialistas en seguridad en cómputo que atiende a instituciones de cualquier tipo, que han sido víctimas de algún ataque tanto en sus sistemas de cómputo como en sus sitios de Internet; además, publica periódicamente información actualizada sobre alertas y vulnerabilidades, implantación de políticas, elabora análisis de riesgos y realiza investigación dentro de esta área para contribuir a hacer, cada día, más seguros los sistemas y las redes.

Una tarea particular de este organismo académico, consiste en la realización de programas de divulgación y capacitación en seguridad en cómputo, por lo que realiza los seminarios Grupo de Administración y Seguridad en Unix (GASU) y Admin-UNAM. Participa desde 1993 en la celebración mundial del Día Internacional de Seguridad en Cómputo (DISC) y anualmente realiza un congreso internacional altamente especializado.

Actualmente, UNAM-CERT atiende un promedio de 400 incidentes al año en coordinación con los mejores equipos de respuesta a incidentes de todo el mundo, y entre sus principales funciones también se encuentran informar, catalogar, clasificar y analizar oportunamente los problemas relacionados con la seguridad en cómputo; así como regular, controlar estándares y facilitar la difusión de normas para que los laboratorios, centros de investigación, instituciones bancarias y financieras, empresas y organizaciones las adopten en su beneficio; también trabaja en línea y ofrece un foro abierto en donde concurren diferentes instituciones, escuelas y universidades para obtener información, asesorías y herramientas especializadas.

UNAM-CERT es un organismo universitario y sin fines de lucro, único en su tipo en América Latina y el Caribe, pues sólo hay dos más, uno en Brasil y el otro en Perú, el primero pertenece a la policía brasileña y el segundo opera con un enfoque comercial en tanto pertenece a la compañía transnacional ATT.

En México, en lo que va del año, los ataques se han incrementado y nadie es inmune a la amplia gama de actos maliciosos, por tanto, la seguridad en cómputo debe

apreciarse como un problema de personas y de procesos que puede solucionarse con la conciencia del problema y con la tecnología.

Aquí, es importante destacar que cerca del 70% de los crímenes informáticos los realiza personal especializado que trabaja dentro de la misma compañía o institución víctima. Si bien es cierto que en México no se han cuantificado, en términos económicos, las pérdidas que causan los ataques informáticos, el FBI estima que en Estados Unidos las pérdidas ascienden a cerca de los cinco billones de dólares.

2.3 FIRST (Forum of Incident Response and Security Teams)

2.3.1 Historia del FIRST

En noviembre de 1988, un incidente de seguridad en cómputo conocido como el gusano Morris de Internet puso a la mayor parte de las computadoras conectadas a Internet de rodillas. La reacción ante este incidente fue desorganizada y sin coordinación, resultando una mayor duplicación de los daños y soluciones conflictivas. Semanas después el centro de operaciones CERT fue formado.

En los siguientes dos años el número de equipos de respuesta a incidentes ha crecido, cada uno con su propio propósito. La interacción entre estos equipos experimentó dificultades debido a los diferentes lenguajes, tiempos regionales y las convenciones de estándares internacionales. En Octubre de 1989, un gran incidente llamado el gusano Wank sobresaltó la necesidad de una mejor comunicación y coordinación entre los equipos.

El FIRST fue formado en 1990 en respuesta a esos problemas. Desde ese tiempo, ha continuado creciendo y adaptándose a los cambios necesarios de las respuestas a los incidentes, de los equipos de seguridad y sus instituciones. Los miembros FIRST consisten en equipos de una gran variedad de organizaciones incluyendo las educacionales, comerciales, gubernamentales y militares.

2.3.2 Funciones de un FIRST

FIRST es un prestigiado FIRST foro de respuesta a incidentes y equipos de seguridad reconocido a nivel mundial como líder en respuesta a los incidentes de

seguridad. Los miembros en FIRST permiten a los equipos de respuesta a incidentes responder más efectivamente a los incidentes de seguridad por medio de acceso a las mejores prácticas, herramientas y comunicaciones confiables con los miembros de los equipos.

Las funciones que desempeña un FIRST son las siguientes:

- Investiga y desarrolla información técnica, herramientas, metodologías, procesos y las mejores prácticas
- Alienta y promueve el desarrollo de la calidad en la seguridad de los productos, pólizas y servicios.
- Desarrolla y promulga las mejores prácticas de seguridad en cómputo.
- Promueve la creación y expansión de los equipos de respuesta a incidentes y de los miembros de las organizaciones de todo el mundo.
- Los miembros de FIRST combinan los conocimientos, habilidades y experiencia para un estable y más seguro ambiente global electrónico.

Tiene como objetivos:

- Rápida cooperación entre las instituciones de información tecnológica en la efectiva prevención, detección y restablecimientos de los incidentes de seguridad en cómputo.
- Tomar en cuenta las llamadas de alerta e información de aviso en los atentados potenciales y incidentes de situaciones de emergencia.
- Facilitar la acción y las actividades de los miembros del FIRST incluyendo investigación y actividades operacionales; y
- Facilitar la búsqueda de la información relativamente segura, herramientas y técnicas.

Cualquier equipo de respuesta a incidentes que tiene la responsabilidad de coordinar los incidentes de seguridad en cómputo son los que se integran como miembros del FIRST. Esto les permite intercambiar ideas y soluciones técnicas para problemas comunes, los equipos mejoran en gran medida su efectividad, los miembros del FIRST permiten a los equipos ver como resuelven otros equipos un determinado problema, dando a cada miembro la oportunidad de hacer una contribución válida para una solución global.

2.3.3 FIRST en el mundo

FIRST es un foro internacional de equipos a respuesta a incidentes, entre sus miembros podemos mencionar a los siguientes en las tablas 2.1 a 2.7:

Internet y otros Equipos de respuesta de red de FIRST	
Organización	Conformado
AUSCERT (Australian Computer Emergency Resp. Team) Equipo de Respuesta a Emergencias en Cómputo	Australia
CERT® Centro de coordinación	Internet
CERT-IT, (Computer Emergency Response Team Italian) Equipo de Respuesta a Emergencias en Cómputo Italiano	Internet en Italia
DFNCERT(Deutsches ForschungsNetz Computer Emergency Response Team)	Alemania
Academia de Redes de Israel	Usuarios de la Universidad de Israel
JANET-CERT	Todas las organizaciones conectadas a la red JANET.
UNAM-CERT (Mexicano CERT)	México (.mx domain)
NORDUnet (Red Nacional Nordica)	NORDUnet

Tabla 2.1 Equipos de respuesta de red de FIRST

Como se aprecia en la tabla 2.1, México también es un miembro activo del FIRST por parte del equipo de respuesta a incidentes de la UNAM, UNAM CERT.

Equipos de Respuesta en FIRST de las agencias del gobierno de los Estado Unidos	
Organización	Conformado
Departamento de Energía de CIAC	Departamento de energía de U.S. (DOE) y Sitios contratista de DOE, la Red de Ciencias de Energía Plus (ESnet)
Centro de Vuelos Espaciales	Centro de Vuelos Espaciales
NASA (Ames Research Center)	NASA (Ames Research Center)
NASA (Auto. Sys. Incid. Resp. Capability,	NASA y La Comunidad Internacional

NASA (Auto. Sys. Incid. Resp. Capability, NASIRC) Sistema Automático de Capacidad a Respuesta a Incidentes	NASA y La Comunidad Internacional Aeroespacial
NCSA-IRST (National Center for Supercomputing Applications IRST) Centro Nacional de Supercómputo Aplicado	La Comunidad Nacional de Supercómputo, en particular nuestros compañeros industriales, Colaboradores, el estado de Illinois, y 12,000 mosaicos de comunidades de aprendizaje en Illinois.
Instituto Nacional de la Salud de los Estado Unidos	Empleados del Instituto Nacional de la Salud
NIST/CSRC (The National Institute of Standards and Technology / The Computer Security Resource Center	NIST y agencias civiles de los Estado Unidos (Solo Gobernares)
Administración de la Seguridad Social de los Estado Unidos	Administración de la Seguridad Social de los Estado Unidos
Administración de Pequeños Negocios(Small Business Administration, SBACERT)	Comunidad de pequeños negocios nacionales.
Administración de la Salud de Veteranos, Equipos de Foros de Respuesta a incidentes de Seguridad	Administración de la Salud de Veteranos

Tabla 2.2 Equipos de respuesta en FIRTS

Equipos de Respuesta a Incidentes en FIRST del Ejército de los Estados Unidos	
Organización	Conformado
CERT de la Fuerza Aérea (AFCERT, Air Force CERT)	Todos los usuarios de la Fuerza Área
Agencia de Información de Sistemas de la Defensa	MILNET
(NAVCIRT, Naval Computer Incident Response Team) Equipo de Respuesta a Incidentes en Cómputo Naval	Departamento de la marina de los Estados Unidos.

Tabla 2.3 Equipos de respuesta en FIRTS del Ejército de E.U

Equipos de Respuesta a Incidentes en la Educación de los Estados Unidos en FIRST	
Organización	Conformado
Universidades del Noroeste	Universidades del noroeste Facultades/Equipos/Estudiantes
Equipo de Respuesta a Incidentes de la Universidad de Ohio (Ohio State University Incident Response Team,OSU-IRT)	Universidad Estatal de Ohio
Universidad del Estado de Pennsylvania	Universidad del Estado de Pennsylvania
Equipo de Respuesta a Emergencias en Cómputo de Purdue (Purdue Computer Emergency Resp. Team, PCERT)	Universidad de Pordue
Stanford University Network Security Team. Equipo de Seguridad en Redes de la Universidad de Stanford	Redes y Sistemas de la Universidad de Stanford

Tabla 2.4 Equipos de respuesta en FIRTS en educación de EU

Equipos de Respuesta a Incidentes Extranjeros en FIRST	
Organización	Conformado
CCTA (CONSUMER CREDIT TRADE ASSOCIATION)	Agencies de gobierno de UK
Agencia de Investigaciones de defensa en Malven	Agencias de Investigaciones de Defensa
Renater	Ministerio de Educación e Inv. de Francia

Tabla 2.5 Equipos de respuesta en FIRTS extranjeros

Equipos de Repuesta a Incidentes de comunicaciones y cómputo comercial en FIRST	
Organización	Conformado
Computadoras Apple	Computadoras Apple en la Red
Sistemas Cisco	Sistemas Cisco (empleados/contratistas)
Corporación de Equipos Digitales (Digital Equipment Corporation ,DEC SSRT)	Corporación de Equipos Digitales y sus Consumidores

BSD (Berkley Software Distribution) Libre, Inc.	Usuarios de BSD libre u otros sistemas operativos UNIX
Hewlett-Packard Company	Todos los equipos HP-UX y consumidores de MPE
IBM (International Business Machines)	Consumidores internos y externos de IBM
Equipo de Respuesta a Emergencias en Cómputo de Motorola	Motorota
Silicon Graphics Inc.	Toda la comunidad de usuarios de Silicon Graphics
SUN Microsystems, Inc.	Clientes de Sun Microsystems
Equipos de Respuesta a Emergencias en cómputo de UNISYS (UNISYS Computer Emer. Response Team UCERT)	Usuarios Internos y Externos de Unisys
Sprint	Sprint Net (X.25) y Sprint Link (TCP/IP)

Tabla 2.6 Equipos de respuesta en FIRTS en comunicaciones y cómputo comercial

Otros Equipos de Respuesta Comerciales en FIRST	
Organización	Conformado
ANS CO+RE Systems, Inc.	Usuarios de ANS
Bellcore	Bellcore
Boeing CERT (BCERT)	Boeing
General Electric Company	Trece negocios de General Electric
Compañía Goldman, Sachs	Oficinas en la Red de Goldman, Sachs
JP Morgan	Empelados y contratista de JP Morgan
Centro de Respuestas a Emergencias de Seguridad (SAIC Security Emergency Response Center)	Clientes comerciales y Gubernamentales
Corporación Westinghouse Electric	Toda la corporación.

Tabla 2.7 Equipos de respuesta en FIRTS comerciales

Todos los equipos son responsables de proveer al FIRST la última información y contactos. Los anteriores son todos los CERT que en este momento forman parte del FIRST

2.4 Conceptos necesarios para entender el análisis forense

Información forense

Según el FBI, la informática (o computación) forense es la ciencia de adquirir, preservar, obtener y presentar datos que han sido procesados electrónicamente y guardados en un medio computacional[30].

Objetivos del Análisis Forense

Poco a poco los crímenes informáticos, su prevención y procesamiento se vuelven más importantes. Esto es respaldado por estudios del número de incidentes reportados por las empresas.

El análisis forense tiene tres objetivos:

1. Compensación de los daños causados por los criminales e intrusos.
2. La persecución y procesamiento judicial de los criminales.
3. La aplicación y creación de medidas para prevenir casos similares.

Investigación tecnológica

Los investigadores del análisis forense usan una gran cantidad de técnicas para descubrir evidencia, incluyendo herramientas de software que automatizan y aceleran el análisis computacional.

Evidencia digital

La evidencia computacional es única, cuando se la compara con otras formas de evidencia documental. A diferencia de la documentación en papel la evidencia computacional es frágil y una copia de un documento almacenado en un archivo es

[30] Línea de Especialización Análisis Forense e Implicaciones Legales, DGSCA-UNAM, 2003.

idéntica al original. Otro aspecto único de la evidencia computacional es el potencial de realizar copias no autorizadas de archivos, sin dejar rastro de que se realizó una copia. Esta situación crea problemas concernientes a la investigación del robo de secretos comerciales, como lista de clientes, material de investigación, archivos de diseño asistido por computadora, fórmulas y software propietario.

Debe tenerse en cuenta que los datos digitales adquiridos de copias no se deben alterar de los originales del disco del sospechoso, para esto se utilizan varias tecnología como checksum o hash.

La IOCE (International Organitation on Computer Evidence)[31] define los siguientes cinco puntos como los principios para el manejo de evidencia computacional:

1. Sobre recolectar evidencia digital, las acciones tomadas no deben cambiar por ningún motivo esta evidencia.
2. Cuando es necesario que una persona tenga acceso a evidencia digital original, esta persona debe ser un profesional forense.
3. Toda la actividad referente a la recolección, el acceso, almacenamiento o la transferencia de la evidencia digital, debe ser documentada completamente, preservada y disponible para la revisión.
4. Un individuo es responsable de todas las acciones tomadas con respecto a la evidencia digital mientras está en su posesión.
5. Cualquier agencia que sea responsable de recolectar, tener acceso, almacenar y/o transferir evidencia digital es responsable de cumplir con estos requisitos.

Además definen los principios para la recuperación estandarizada de evidencia computarizada, se debe gobernar por los siguientes atributos:

1. Consistencia con todos los sistemas legales.
2. Permitir el uso de un lenguaje común.
3. Durabilidad.
4. Capacidad de cruzar límites internacionales.
5. Capacidad de ofrecer confianza en al integridad de la evidencia.
6. Aplicabilidad a toda la evidencia forense.

[31] http://www.ioce.org

Análisis de Discos

La clave de la computación forense es el análisis de discos duros, disco extraíbles, CDs, discos SCSI, y otros medios de almacenamiento. Este análisis no sólo busca archivos potencialmente incriminatorios, sino también otra información valiosa como passwords, logins y rastros de actividad en Internet.

Existen muchas formas de buscar evidencia en un disco. Muchos criminales no tienen la más mínima idea de cómo funcionan las computadoras, y por lo tanto no hacen un mayor esfuerzo para despistar a los investigadores, excepto por borrar archivos, que pueden ser recuperados fácilmente.

Los investigadores forenses, utilizan herramientas especiales que buscan archivos "suprimidos" que no han sido borrados en realidad, estos archivos se convierten en evidencia.

2.4.1 Pasos para la Recolección de Evidencia

El procedimiento para la recolección de evidencia varía de país a país, y por lo tanto, un análisis exacto y completo está fuera de los límites de este documento. Sin embargo, existen unas guías básicas que pueden ayudar a cualquier investigador forense:

Hardware

El hardware es uno de los elementos que se deben tener en cuenta a la hora de la recolección de evidencia, debido a que puede ser usado como instrumento, como objetivo del crimen, o como producto del crimen (por Ej. contrabando o robo), es por eso que se deben tener consideraciones especiales. Lo primero que se debe preguntar el investigador es qué partes se deben buscar o investigar.

Cuidados en la Recolección de Evidencia

La recolección de evidencia informática es un aspecto frágil de la computación forense, especialmente porque requiere de prácticas y cuidados adicionales que no se tienen en la recolección de evidencia convencional. Es por esto que:

- Se deben proteger los equipos del daño.

• Se debe proteger la información contenida dentro de los sistemas de almacenamiento de información (muchas veces, estos pueden ser alterados fácilmente por causas ambientales, o por un simple campo magnético).

• Algunas veces, será imposible reconstruir la evidencia (o el equipo que la contiene), si no se tiene cuidado de recolectar todas las piezas que se necesiten.

Herramientas para la Recolección de Evidencia

Existen una gran cantidad de herramientas para recuperar evidencia. El uso de herramientas sofisticadas se hace necesario debido a:

1. La gran cantidad de datos que pueden estar almacenados en una computadora.

2. La variedad de formatos de archivos, los cuales pueden variar enormemente, aún dentro del contexto de un mismo sistema operativo.

3. La necesidad de recopilar la información de una manera exacta, y que permita verificar que la copia es exacta.

4. Limitaciones de tiempo para analizar toda la información.

5. Facilidad para borrar archivos de computadoras.

6. Mecanismos de encriptación, o de contraseñas.

Herramientas para el Monitoreo y/o Control de Computadores

Algunas veces se necesita información sobre el uso de las computadoras, por lo tanto existen herramientas que monitorean el uso de las computadoras para poder recolectar información. Existen algunos programas simples como key loggers o recolectores de pulsaciones del teclado, que guardan información sobre las teclas que son presionadas, hasta otros que guardan imágenes de la pantalla que ve el usuario del computador, o hasta casos donde la máquina es controlada remotamente.

Herramientas de Marcado de documentos

Un aspecto interesante es el de marcado de documentos; en los casos de robo de información, es posible, mediante el uso de herramientas, marcar software para poder detectarlo fácilmente.

El foco de la seguridad está centrado en la prevención de ataques. Algunos sitios que manejan información confidencial o sensitiva, tienen mecanismos para validar el ingreso, pero, debido a que no existe nada como un sitio 100% seguro, se debe estar preparado para incidentes.

Herramientas de Hardware

Debido a que el proceso de recolección de evidencia debe ser preciso y no debe modificar la información se han diseñado varias herramientas como DIBS "Portable Evidence Recovery Unit" Unidad Portátil de Recolección de Evidencias.

2.4.2 Cadena de custodia

- Establece qué personas tendrán la custodia de la evidencia.
- Establece la continuidad de la posesión.
- Prueba de integridad del manejo de la evidencia colectada.

La cadena de custodia establece un registro de cómo ha sido manejada la evidencia. Muestra quién la manejó y mantiene acceso estricto a ella. Este sistema de identificación se conoce en casos legales (en la corte) como una cadena de custodia. Esta cadena de custodia es uno de los aspectos fundamentales que con frecuencia se soslaya.

Para cualquiera que se quiera tomar en serio la realización de un análisis forense es esencial mantener una práctica de cadena de custodia correcta.

Debido a que los medios electrónicos son muy fáciles de manipular o destruir, un investigador debe ser excepcionalmente cuidadoso cuando obtiene nueva evidencia.

Para archivos electrónicos, puede usarse la herramienta MD5[32] para ayudar a crear una firma única para cada archivo creado. Esto ayudará a asegurar que los archivos puedan ser admitidos como evidencia si se toman desde un firewall o un IDS. La herramienta MD5sum creará éstas firmas únicas. Esto es, también, muy útil si se necesita transferir archivos por una red.

Elementos de la cadena de custodia

- Fecha y hora en que se obtuvo.
- Ubicación.
- De quién se obtuvo.
- Arquitectura, modelo y número de serie.
- Número de personas que recolectaron la evidencia.
- Descripción de la evidencia.
- Nombre completo y firma de la persona que recibe la evidencia.
- Número de caso y elemento (etiqueta) de la evidencia.
- Valores hash (por ejemplo MD5sum) de la evidencia si es posible obtenerlos.
- Datos técnicos pertinentes (por ejemplo geometría del disco).

Para cada elemento que se obtiene como evidencia, es necesario determinar la información de arriba. Podría ser de un floppy, una computadora, un módem, etc. Los medios son la forma más generalizada de etiquetación de la cadena de custodia. Debe crearse una etiqueta separada para cada pieza de medios distintos. Si se incauta una computadora, entonces los discos duros deberían quitarse y etiquetarse de forma separada del sistema.

Aún si esta información no se proporciona para un seguimiento legal, podría ser crucial en una investigación interna si se decide perseguir posteriormente. Esto debería, evidentemente, ser parte del plan de manejo de respuesta a incidentes.

[32] Véase Apéndice E

2.4.3 Dificultades del Investigador Forense

El investigador forense requiere de varias habilidades que no son fáciles de adquirir, es por esto que el usuario normal se encontrará con dificultades como las siguientes:

1. Carencia de software especializado para buscar la información en varios computadoras.

2. Posible daño de los datos visibles o escondidos, aún sin darse cuenta.

3. Será difícil encontrar toda la información valiosa.

4. Es difícil adquirir la categoría de 'experto' para que el testimonio personal sea válido ante una corte.

5. Los errores cometidos pueden costar caro para la persona o la organización que representa.

6. Dificultad al conseguir el software y hardware para guardar, preservar y presentar los datos como evidencia.

7. Falta de experiencia para mostrar, reportar y documentar un incidente computacional.

8. Dificultad para conducir la investigación de manera objetiva.

9. Dificultad para hacer correctamente una entrevista con las personas involucradas.

10. Reglamentación que puede causar problemas legales a la persona.

Es por esto que, antes de lanzarse a ser un investigador forense, se necesita bastante estudio y experiencia, entre otras cosas, y si no se cumple con los requisitos, en caso de un accidente es aconsejable llamar a uno o varios expertos.

2.5 Método a desarrollar para aplicar el análisis forense.

2.5.1 Metas del investigador forense

- Encontrar señales de que el sistema está comprometido.
- Determinar el tamaño del daño.
- Responder: qué, cuándo, dónde, cómo.
- Reconstrucción cronológica de los hechos.
- Manejar y analizar la evidencia de tal forma que pueda servir para demostrar los hechos.

2.5.2 Límites del investigador forense

Debe tenerse en cuenta, durante una investigación de este tipo, especialmente cuando se trabaja en sistemas multiusuario, que con frecuencia se encontrará uno con información que no tiene "derecho" a examinar. Las ramificaciones legales pueden ser problemáticas, pero las éticas son tanto, o más, serias.

Podemos llegar a encontrarnos con información altamente sensitiva, como puede ser el correo personal, notas privadas, patrones de navegación por la web, etc. Ya que aún los archivos borrados pueden examinarse, es indispensable ser muy escrupuloso con el trato de esa información. Nunca, bajo ninguna circunstancia, debe usarse la información surgida de una investigación para tomar ventajas personales propias o de alguien más; de igual forma, únicamente se debe dedicar a labores de trabajo sin inmiscuirse en la información contenida en la evidencia. Las personas encargadas de aplicar el análisis forense deben de estar apegadas a un código de ética anteriormente establecido y si no cumplen algún punto del mismo, no deben ejercer en esta área. Por último, recomendamos que las personas que apoyen a los analistas forenses, si detectan que estas personas fueron o están involucradas en incidentes de seguridad siendo partícipes ellos como infractores de la seguridad, no permitan que sigan aplicando el análisis forense; no por el hecho de ser las personas que más saben en esta área son las mejores y las que se apegan a los códigos de ética.

Por otra parte, se debe tener cuidado en el manejo de la información porque la mayor parte de ella es volátil y con cualquier descuido se puede perder evidencia valiosa.

Un grupo de analistas forenses debe contar con el quipo necesario para aplicar el análisis forense, ya que sin éste puede truncarse la investigación o ser insuficiente por falta de recursos.

El equipo de analistas forenses al tener enfrente un caso debe conocer los obstáculos que se le presentan junto con las limitantes que existen dentro de la institución en la cual fue requerido. En ocasiones la evidencia puede perderse por las limitaciones impuestas dentro de la institución. Un claro ejemplo de esto es cuando se quiere realizar la copia de la evidencia y por políticas de la institución no permiten que la información sea copiada o que salga de la institución.

Un analista forense debe estar al tanto de las vulnerabilidades que existen hoy en día para el sistema operativo en cuestión, saber de qué forma puede ser explotada una vulnerabilidad y cómo impacta al sistema. También debe estar inscrito en listas de seguridad. En general, conocer y ampliar los conocimientos día a día en cuanto a seguridad se refiere. Sin esto, puede pasar desapercibido debido al desconocimiento y la mala preparación del investigador forense.

2.5.3 Desarrollo

El análisis forense no es una especialidad reciente. Por la importancia de las computadoras en diversas áreas, siempre ha existido la auditoría a sistemas informáticos. Además, la ciencia forense es metódica y se basa en acciones premeditadas para reunir pruebas y analizarlas. La tecnología, en caso de análisis forense en sistemas informáticos, son aplicaciones que hacen un papel importante en reunir la información y pruebas necesarias. La escena del crimen es el ordenador y la red a la cual éste está conectado.

Antes de iniciar a aplicar el análisis forense debemos estar familiarizados con tácticas y herramientas de los crackers como lo son: la forma en que llevan a cabo un ataque y cómo ocultan datos para no ser detectados. Es indispensable entender las tácticas así como las herramientas de los intrusos, conocer vulnerabilidades, exploits, rootkits, etc.

Además de lo anterior el analista forense debe estar familiarizado con:

- Bagaje forense y de la investigación.

- Preparación del incidente.
- Definiciones que sean necesarias.
- Hardware y Software que se necesite.
- Técnicas de entrevista.

Uno debe estar practicando constantemente, además de saber cómo responder a un incidente. Saber en el momento preciso qué material va a utilizar sin que esto le afecte en su trabajo.

También debemos detectar y conocer los pasos comunes de un ataque:

- Reconocimiento
- Explotación
- Operaciones ocultas
- Podría ser:
    - Destinado específicamente
    - Aleatorio

En un inicio se comienza a explorar la red en forma general, ya sea con el fin de encontrar a una víctima en específico o simplemente para detectar a un sistema con ciertas vulnerabilidades las cuales pueden ser explotadas. Una vez que se ha encontrado un sistema vulnerable, se lleva a cabo la explotación, el paso siguiente es la eliminación de la evidencia y la preservación de la estancia para futuras incursiones al sistema sin ser detectado. Esto último colocando backdoors y rootkits para el regreso al sistema y la ocultación de la intrusión.

Ante esto, un analista forense debe pensar como intruso porque algunos incidentes son sólo la punta del iceberg, es decir, muchas veces, encontrar un sistema comprometido significa que se encontrarán otros, y por tanto siempre se debe investigar con este hecho en mente. Para esto, cabe la posibilidad de observar al intruso regresando al sistema y ver qué está haciendo y seguirlo.

Los propósitos que tienen los intrusos pueden ser:

- DoS
- Alteración de información

- Piratería de software
- Espionaje
- Uso de recursos
- Venganza
- Robo
- Diversión
- Obtener dinero
- Motivos políticos
- reto, etc.

Tratar siempre de tener en mente el por qué de la situación.

Cuando se está aplicando el análisis forense debemos comenzar a formular el perfil del intruso, saber su nivel técnico, qué tan bien se oculta y responderse a la siguiente pregunta: ¿Está usando un método nuevo?

Al analizar el por qué de un acceso no autorizado o de cualquier ataque a una red o sistema es necesario tomar en cuenta el entorno en el que ocurre. Muchas veces existen condiciones de trabajo o relaciones laborales que pueden dar indicios sobre el origen del problema. En todos los casos es importante analizar cuidadosamente la información obtenida de entrevistas con las personas involucradas y mantenerse imparcial.

A pesar de que se aplican metodologías para poder llevar a cabo un análisis forense, gran parte de ella se obtiene con base en la experiencia misma, porque cada caso es diferente y además porque puede ser cualquier persona la que lo realice. Por tanto, el investigador debe tomar decisiones basándose en su experiencia y el "sexto sentido" para llegar al fondo del caso en estudio. Esta afirmación que nosotros estamos realizando es de mucho riesgo, ya que si nosotros queremos aplicar el análisis forense de forma profesional y dando un sustento a la parte afectada, no sólo de la experiencia se debe apoyar el análisis forense. Por ello, en esta parte de la tesis vamos a plantear una metodología general para poder llevar a cabo con éxito un análisis forense sobre una máquina afectada. No podemos particularizar porque existen infinidad de casos que se pueden presentar, pero con este trabajo que realizamos, cualquier área de seguridad se puede apoyar en nuestra metodología como un recurso para aplicar el análisis forense sin importar el caso que se le presente, lo que le permitirá adentrarse en este campo y conforme avance en su

incursión, con base a su aprendizaje y experiencias personales podrá particularizar o clasificar los casos.

2.5.4 Aspectos importantes para el análisis forense

- Miedo, incertidumbre, duda.
    - Hay que tener comunicación con aquellos que necesiten saber del incidente de seguridad.
- Acción rápida pero cuidadosa.
- Registrar la situación sin modificarla.
- Recolectar y manejar evidencia admisible.
    - Evitar la contaminación.
    - No duplicar.
    - No perder el control físico.
- Integridad de las herramientas

El trabajo del investigador forense comienza desde el primer momento en que se le informa de un incidente de seguridad en el cual él va a ser la persona que tomará el caso. El investigador forense debe iniciar en todos los casos con la firma de un documento en el cual debe contener la fecha, descripción a grandes rasgos del incidente por el cual fueron requeridos, el lugar, la institución, nombres completos de los administradores o responsables directos de la máquina víctima, nombres completos de los investigadores forenses que estarán involucrados en el caso, y por último se debe anexar una breve encuesta donde se tomará registro del evento en cuestión de forma superficial para tomarlo como referente al momento de tocar a la víctima.

Hablamos en este momento de las personas que tomarán el caso, debido a que nosotros, dentro de nuestra metodología planteamos el ideal de dos personas para llevar la investigación. una que realice el análisis y otra que haga todas las anotaciones necesarias, y ambas deciden la mejor acción a tomar con base a la metodología planteada, de esta forma se ahorra tiempo. Ambas deben estar en todo momento ante la evidencia y presentarse juntos ante cualquier llamado de la institución para hacer entrega del resultado de la investigación o para hacer algún tipo de aclaración. De los dos investigadores, únicamente será una persona la que toque a la víctima junto con la evidencia, y será la que determine el veredicto final. El segundo investigador forense debe

apoyar, sugerir, colaborar y anotar los acontecimientos más sobresalientes del caso. Esto porque en ocasiones se tiene que actuar de forma rápida, y la pérdida de tiempo junto con el nerviosismo puede acarrear que se pasen por desapercibido eventos que pueden servir como evidencia.

En el Apéndice B de este trabajo mostramos el formato que recomendamos tener para el documento donde se registre el caso, junto con el cuestionario que se aplicará al momento de realizar la entrevista. El cuestionario servirá como punto de partida para conocer la situación actual de la víctima, por lo que este cuestionario no va a cumplir de forma exhaustiva toda la parte de la investigación, únicamente como punto de referencia. Con este motivo, las preguntas planteadas son básicas para cumplir con este requisito, esto quiere decir que un analista forense debe incluir las preguntas que él crea convenientes en un momento dado. Aquí llegamos a cuestionarnos acerca de qué es lo que se debe preguntar, todo lo que se refiera o se involucre con el incidente de seguridad, nada se debe perder de vista acerca de esto. Por muy simple o absurda que suene la pregunta se debe hacer para beneficio de las partes afectadas y de los mismos investigadores a fin de dar solución al caso.

Una vez que se ha aplicado la entrevista a las personas involucradas en el caso, ahora ha llegado el momento, con base en las respuestas de la entrevista, de eliminar los obstáculos que existen para poder aplicar el análisis forense. Los obstáculos pueden ser:

- Máquinas críticas.
- Discos de gran capacidad.
- Imposible dar de baja.
- No hay mecanismos de respaldo ya instalados.
- La gente.
    - Es difícil hacer el trabajo con veinte personas en el lugar.
    - Falta de seguimiento estricto de las políticas de manejo de incidentes.
    - Intento de cubrir violaciones de seguridad para desprestigiarse.
- Politicas.
    - No contar con una política de manejo de incidentes.
    - Ser demasiado restrictivas para la aplicación de la investigación.
    - No permitir utilizar sniffers y analizadores de red.
- Legal

- Notificar muy pronto al equipo legal, sin que se disponga del reporte final del análisis, porque no se tendría la evidencia completa para proceder legalmente.
- Se podría definir lineamientos estrictos.
- Probablemente no se desee compartir información.

Para nuestra metodología vamos a tomar como punto de referencia las dos situaciones posibles en las que se va a encontrar a la víctima después de un incidente. La primera: se da cuando el sistema está arriba, esto es cuando se encuentra en producción; y la segunda: cuando el sistema se encuentra apagado. Para nuestra primera situación posible hay que tomar en cuenta que es la más difícil de las dos, ya que nos encontramos ante un sistema dinámico en constante cambio y ante variables completamente desconocidas. También porque se tienen más cosas que recopilar como lo es la memoria en ejecución y los procesos junto con las conexiones de los usuarios. Además del análisis de tráfico de la red que existe entre la máquina y su entorno.

Para el caso donde encontremos apagado el equipo afectado, nos podemos enfrentar a una situación mucho más simple que cuando se encuentra arriba porque el sistema es estático por lo que es más fácil la tarea de duplicación, tenemos menos cosas que recolectar, no hay que preocuparse por obtener la información de memoria ni de procesos. Mientras que la memoria y los procesos son importantes cuando se encuentra un sistema vivo, si el sistema fue apagado estos se pierden y no hay forma de obtenerlos.

Con esto en mente procedemos a definir la metodología que debe aplicar un analista forense en cualquier caso que se le presente.

PASO 1: Determinar si el equipo afectado se encuentra apagado o se encuentra arriba. Cuando el equipo se encuentra apagado procedemos a quitar el disco duro de la víctima con el propósito de comenzar a analizar la evidencia. En caso de que no se pueda quitar el disco duro para poder obtener las imágenes de las particiones, entonces se debe bootear el sistema desde floppy y montar las particiones debidamente para que éstas no se vean afectadas por nuestra actividad*. Al montar las particiones de esta forma, no podemos asegurar del todo que el incidente se encuentre controlado, ya que el sistema se va a encontrar encendido y se podría comportar como si estuviese en producción. El

*La parte práctica de este punto se describirá a detalle en el quinto capítulo.

inconveniente de bootear desde floppy es que nosotros tenemos que hacer pequeñas modificaciones sobre archivos para que esto se pueda cumplir, por tanto se recomienda que se quite el disco duro afectado o en su defecto, que nuestro disco duro de analista forense sea el que se deba agregar sobre la máquina afectada cuidando que el disco afectado quede como esclavo al momento de realizar la configuración de los jumpers. Esta actividad implica que tengamos que abrir el gabinete del sistema para agregar nuestro disco duro, modificar la configuración de los jumpers de tal forma que nuestro disco duro quede como maestro. Una vez que se realizó esto, llega el momento de encender el sistema y dejar que boote nuestro disco duro y posteriormente montar las particiones del disco afectado. Todo esto se realiza en el caso de que no se cuente con un disco duro externo. Si tenemos un disco duro externo las cosas se facilitan haciendo la transferencia de los datos de forma directa sin necesidad de apagar el sistema operativo afectado, y así, nuestro sistema operativo tome el control del disco involucrado en el incidente. Cualquiera de las dos formas es buena.

PASO 2: A partir de este momento debemos hacer anotaciones de todos los procesos y acontecimientos más importantes y sobresalientes que haya.

PASO 3: Lo siguiente es montar el disco duro. Esterilizar el disco duro donde se va a llevar a cabo la copia, es decir, limpiar por completo esa parte del disco para que no infecte la copia que se va a realizar. Con esto garantizamos que todo lo que encontremos ahí, sea lo que le pertenece al disco duro perjudicado. Una vez que se realizó la esterilización, podemos obtener las imágenes del disco duro perjudicado. Ya obtenidas las imágenes, nos disponemos a realizar las firmas digitales tanto de las copias como del original. Esto nos garantiza que al momento que obtuvimos las imágenes, tanto imagen como original son lo mismo comprobando las firmas de una con la otra. Si las firmas son iguales hemos realizado una imagen idéntica al disco duro perjudicado, en caso de no ser así, volvemos a realizar este paso hasta que las firmas coincidan. Si nos es imposible que las firmas digitales coincidan, tendremos que utilizar el disco duro original, que no se recomienda.

Una vez que hemos obtenido la copia del disco duro involucrado en el incidente o en su defecto, manipular el disco original, podemos iniciar el análisis de la evidencia sobre la copia. Esto es muy importante, todo lo que se haga debe ser sobre la copia, por ningún motivo se debe manejar el original porque éste es la evidencia más importante y cualquier

error que cometamos puede ser catastrófico; por este motivo, se recomienda que se manipule la copia, pero si no se pueden obtener las coincidencias de firmas, no queda otra posibilidad más que manipular el disco original. Al analizar la evidencia debemos comenzar basándonos en la encuesta que se realizó previamente: partir de ahí puede ahorrar tiempo y esfuerzo. También podemos iniciar buscando archivos extraños en directorios como /dev o /usr, además localizar archivos ocultos en todo el sistema operativo. Buscar también con base en los tiempos MAC en el sistema. Todo lo que hagamos será con el fin de encontrar evidencia o pistas que nos confirmen de forma rotunda la existencia de un incidente, porque en ocasiones puede ocurrir que es sólo una mala configuración que se haya realizado al servidor, o que un virus intente propagarse en sistemas Windows haciendo que esto se vea registrado en los logs del sistema únicamente. Si se detecta la existencia de un incidente, entonces comenzamos a recolectar toda la información y evidencia que se encuentre, buscando pistas tanto en el sistema mismo como en otros, como lo es en IDS, firewall, server logs y encaminadores. Con toda la información recolectada hay que comenzar a obtener el perfil del atacante, es decir, determinar los propósitos del mismo, si es un empleado o ex-empleado que busca venganza, si es un simple script-kiddie que se encontró en su camino ese sistema con deficiencias en seguridad y logró entrar. Si es un hacker experimentado que utilizó el sistema únicamente como puente para atacar otro sistema con mayor importancia, etc.

Con toda la información obtenida, comenzamos a sacar todas las conclusiones pertinentes, respondiendo a las preguntas: ¿Qué sucedió?, ¿Cuándo ocurrió?, ¿Dónde se llevó a cabo el daño?, ¿Cómo se llevó a cabo? Hay que reconstruir los hechos con base en el tiempo en que se ejecutaron. Si se da el caso en que se llegó a afectar a más sistemas, entonces determinar cuál fue el primero que sufrió un incidente y qué relación existe entre todos los sistemas afectados. Por último resguardar toda la evidencia y manipularla con mucho cuidado para demostrar qué es lo que sucedió en el reporte final que se entregará a la parte afectada.

La evidencia debe ser resguardada siguiendo la cadena de custodia.

En la figura 2.1 se muestra el diagrama de la metodología que recomendamos utilizar cuando el sistema se encuentra abajo.

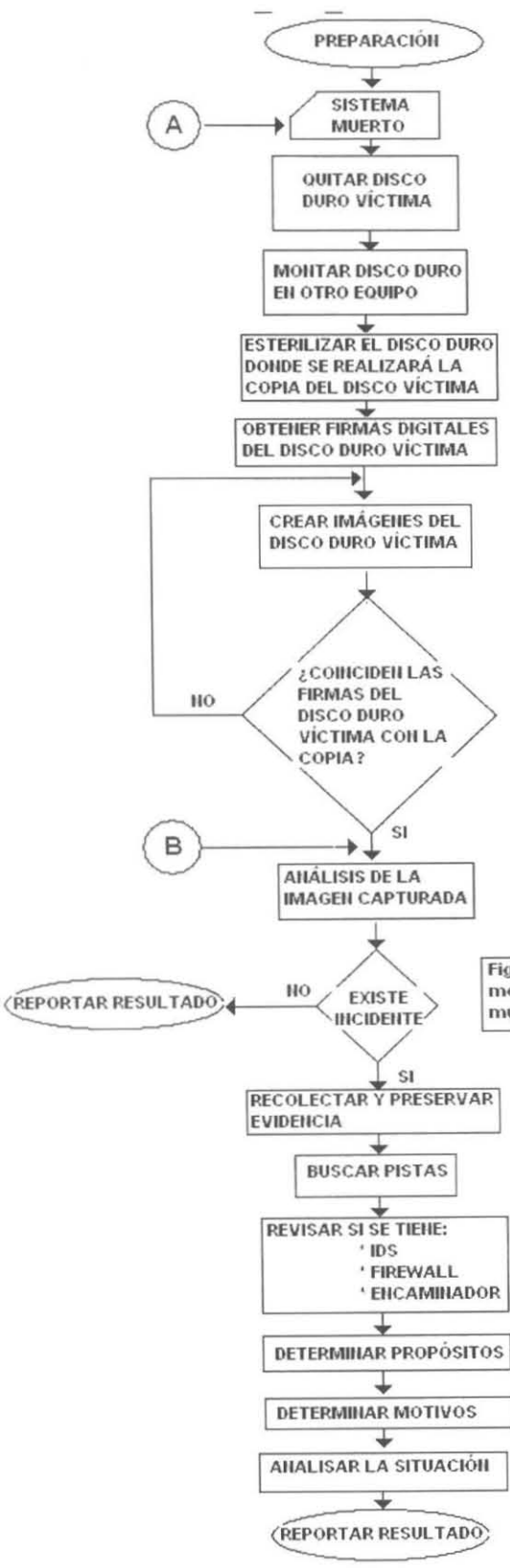


Figura 2.1 Diagrama de la metodología para un sistema muerto.

Los conectores "A" y "B" hacen referencia a la figura 2.2

Hasta el momento hemos seguido el caso en que el sistema se encuentre apagado, qué sucede ahora cuando el sistema se encuentra en producción. La situación se complica, ya que hay más variables por contemplar y desconocemos si en ese preciso momento el intruso se encuentra interactuando con el sistema perjudicado, y si es así, éste se puede dar cuenta y borrar toda evidencia. O también se puede dar el caso de que el intruso impida que alguien se loguee a la máquina víctima como administrador del sistema y esto dificultaría un poco el trabajo sobre la máquina.

Cuando el sistema se encuentre en producción vamos a tener dos opciones, la primera es donde sólo se tiene sospecha de que existe un incidente y la segunda cuando sí hay un incidente. Para el primer caso tenemos que verificar sí existe o no un incidente. De igual forma que cuando se presenta una situación en donde el sistema está apagado, en todo momento se debe de contar con un cuaderno para anotar todo y no dejar pasar nada. Algo muy importante, cuando el sistema se encuentra en producción, es ser extremadamente cauteloso al momento de comenzar a analizar el sistema y si se da el caso entonces analizar también la red, porque todo movimiento en falso puede ocasionar pérdida de evidencia y el fracaso de nuestro trabajo. Por lo tanto, siempre hay que tener en mente las cosas que deben evitarse hacer en el sistema, como las siguientes.

- Escribir en los medios originales.
- Matar algún proceso en el sistema (A menos que sea extremadamente necesario).
- Modificar marcas de tiempo accidentalmente.
- Usar comandos o herramientas no confiables.
- Ajustar el sistema antes de obtener evidencia (apagar, parchar, actualizar; a menos que sea necesario).

Y también tener presente los principios forenses:

- Minimizar la pérdida de datos.
- Registrar todo.

Al momento de verificar el incidente, debemos intentar montar el CD de herramientas forenses de acuerdo a nuestra metodología para capturar la mayor evidencia posible. Para ello es importante conocer todas las herramientas que trae consigo el CD forense. Las herramientas que nosotros proponemos que incluya este CD y que nosotros

utilizaremos dentro de nuestras prácticas se describen en el apéndice D. Este CD se recomienda que uno mismo lo realice descargando los programas fuente de la siguientes páginas: http://www.rpmfind.net, http://www.porcupine.org/forensics/tct.html, http://www.zone-h.org/en y también http://rpmfind.rediris.es/search.php. En estas páginas los programas vienen comprimidos por lo que es se tienen que descomprimir y compilar los programas fuente de forma estática. Esto es necesario ya que ante un sistema en producción, no podemos confiar en nada de lo que se encuentre ahí, esto es por el motivo de que el sistema se puede encontrar troyanizado y los comandos no hacen lo que deberían hacer. Si en alguna ocasión no se puede montar el CD de herramientas forenses debido a que el hardware nos lo impide o el mismo sistema no lo permite, entonces lo único que nos queda por hacer es analizar primero el sistema con los comandos propios de la máquina víctima, y si no es posible detectar nada, entonces pasaremos a analizar la red o hacer una copia de algunos binarios que se tienen en el CD forense a otra máquina y de ahí tomarlos hacia la máquina víctima para ejecutarlos. Esa es una forma que no se recomienda porque los programas que el intruso haya dejado en el sistema pueden detectar esta acción realizada por nosotros avisándole que estamos enterados de su ataque y poniéndolo en alerta.

Una vez que se haya montado el CD forense, nos disponemos a registrar el estado de la computadora iniciando con los procesos, es decir, ver qué es lo que se está ejecutando en el sistema y qué programa lo está haciendo. Un punto que podría cuestionarse aquí, es si debemos escanear la máquina afectada. Una de las ventajas de hacerlo sería que podríamos ver todos los puertos activos además de los programas que los abren. La desventaja es que esto afecta a la máquina y algunos programas, como los backdoors que registran cuántas conexiones tiene la máquina, pudiendo informarle al intruso y éste posiblemente borre toda evidencia. También podemos revisar los logs del sistema para detectar conexiones extrañas o en horas que no son normales para la actividad regular, debemos buscar patrones o explotación de una vulnerabilidad sobre algún servicio o posiblemente barrido de puertos; de igual forma revisar el historial de comandos del administrador para detectar si es sólo una mala configuración del sistema o tal vez el intruso aún no borra esa evidencia. Detectar nuevos usuarios dados de alta en el sistema. Buscar archivos extraños en el sistema como lo son ". . " o ". . ." , también encontrar archivos ordinarios donde no deben de estar, como por ejemplo en /dev que es donde únicamente deben de estar archivos de dispositivos, localizar todos los archivos

binarios en el sistema. Verificar por completo el contenido del directorio /tmp. En general, buscar pistas dentro del sistema que nos indiquen que sí hay un incidente de seguridad. Aquí también podemos analizar los IDS, firewall. Una vez que se haya verificado la existencia de un incidente de seguridad en el sistema, si se dio el caso que hubiese sido una falsa alarma, lo que queda por hacer es reportar los resultados obtenidos de nuestra investigación. Pero si desgraciadamente hemos constatado que existe un incidente, a partir de este momento podemos tomar varios caminos de acuerdo a la situación que se presente, que pueden ser:

- Desconectar cable de corriente
- Respaldar evidencia
- Delimitar el daño
- Analizar la red
- Desconectar de la red
- Dar de baja limpiamente

De igual forma, cuando se nos llama e indican que el sistema se encuentra vivo y además nos confirman la existencia de un incidente lo primero que se tiene que hacer es guardar la calma porque ahora sabemos que el intruso es real. En este punto se debe preguntar si se aplicaron políticas de respuesta a incidentes o no. Esto porque en muchos lugares pueden o no contar con dichas políticas. Si no cuentan con ellas o aún no aplican políticas a esta situación, entonces debemos atenernos a lo que hayan hecho dentro de la máquina víctima ya que probablemente por intentar minimizar o erradicar el problema, hayan ocasionado pérdida de evidencia. Entonces con base en lo que hayan realizado nosotros tomaremos el camino que más se adecue a la situación que se presenta. Las posibilidades son las mismas cuando se tiene en un inicio sospecha del incidente y con base en nuestro análisis hemos determinado que sí lo hay.

Al llegar a este punto, es importante tomar el camino correcto porque toda equivocación ocasiona pérdida de información. Los caminos a tomar, como ya se habían mencionado anteriormente, son los siguientes:

Desconectar el cable de corriente

Se debe hacer cuando detectemos robo de información de la máquina victima en ese momento, o que se esté utilizando como puente para atacar a otras máquinas, que es lo más frecuente. De igual forma cuando detectemos que existe una relación de confianza con otras máquinas y determinamos que también fueron vulneradas, si es demasiado grave como robo de información o están utilizándolas como puente a otras máquinas para atacar a más equipos.

Antes de hacerlo hay que tener siempre presente que tanto la información de procesos como memoria se pierde al momento de apagar el sistema y corremos el riesgo de que ya no se vuelva a levantar el sistema operativo provocando la reinstalación del mismo. La única ventaja que obtendríamos al desconectar directamente el cable de corriente es que el sistema ya no estaría vaciando contenido de memoria hacia archivos ordinarios de configuración del sistema provocando que se alteren y también los tiempos MAC. En ocasiones se puede dar que el sistema se comporte de forma totalmente inestable y por tanto, no exista otra opción más que desconectar el cable de corriente. Antes de desconectar, si es posible, tratar de capturar los procesos, conexiones al sistema y el tráfico de red que se está generando del equipo comprometido hacia los demás sistemas. Recordar que, al apagar el equipo de esta forma se pueden llegar a corromper archivo y esto seria fatal si el administrador del sistema no cuenta con respaldos de las particiones ocasionando una pérdida total de información.

Cuando se desconecta el cable de corriente caemos en el caso de un sistema muerto y seguiríamos la metodología para un sistema muerto.

Dar de baja limpiamente

Muy similar a cuando se tiene que desconectar cable de corriente, la diferencia radica en que al momento de dar de baja limpiamente lo que hacemos es, en un inicio, ejecutar el contenido de archivos y programas, y muy probablemente estos hayan sido modificados por el intruso de tal forma que si se da de baja el sistema puede tomar alguna acción, ya sea comenzar a borrar parte de la información o simplemente borre toda evidencia de que existió un incidente. Para evitar que nosotros nos demos cuenta que se realizó una modificación a ciertos archivos, lo que hacen los intrusos es dejar un programa

ejecutándose, el cual detecta el momento en que se está dando de baja el sistema para borrar toda evidencia.

Hay que tomar esta opción cuando detectemos robo de información de la máquina víctima en ese momento o que se esté utilizando como puente para atacar a otras máquinas, además cuando los administradores no cuenten con respaldo de su sistema. Debemos estar seguros de lo que hacen los programas que están en ejecución y determinar si existió alguna modificación por parte del intruso sobre algunos archivos de configuración para saber que es lo que se va a ejecutar al momento de dar de baja el sistema limpiamente.

Si se da de baja limpiamente el sistema caemos en el caso de un sistema muerto y se tomarían los pasos para un sistema muerto. Tanto en el figura 2.1 como en la figura 2.2, existe el conector que indica la relación de ambas situaciones.

Desconectar de la red

Si al momento de analizar el sistema detectamos que el intruso se encuentra en ese momento conectado o está robando información, una solución sería desconectar el cable de red para que así no exista interacción con la red y aislemos por completo al sistema afectado. También puede ocurrir que está inundando la red interna con paquetes ocasionando una denegación de servicio de algunas otras máquinas o comience a atacar a otros sistemas.

Se debe tener presente que al momento de desconectar al equipo de la red, el intruso puede dejar programas en ejecución los cuales detectan cuando ya no están en red y borrar toda evidencia.

Al desconectar la red vamos a tener las opciones de conectar a un hub, de respaldar la evidencia o por último, de apagar el sistema. Conectar a un hub nos ayudaría porque existen programas de intrusos que únicamente determinan si se encuentra en red la máquina o no. Si detectan que no es así, entonces comienzan a borrar toda evidencia, ya que esto para ellos es indicio de que el administrador ya se dio cuenta de que hay algo extraño en su sistema.

Analizar la red

Si el análisis del sistema no arroja información que nos haga ver que en ese momento algo o alguien está robando recursos o que el sistema está interactuando con el intruso, entonces tenemos que pasar a analizar el tráfico en la red, con esto sí detectaríamos de inmediato si hay o no interacción por parte del intruso. O de igual forma, determinar si está interactuando con la máquina el intruso, conocer que información está en tránsito para tomarlo como evidencia. No se recomienda ir directamente a analizar la red, porque en muchos casos las instituciones no permiten que se analice el tráfico en su red o sus políticas son muy restrictivas en ese sentido, muchas ocasiones esta opción no se podrá realizar y perderíamos evidencia valiosa. Para analizar la red se coloca un sniffer dirigida inicialmente hacia el tránsito de la máquina afectada y posteriormente sobre toda la subred para encontrar indicios de otro ataque sobre otra máquina.

Delimitar el daño

Delimitar el daño cuando existen procesos que perjudican gravemente al sistema o también si se encuentran programas que se ejecutan periódicamente y de forma automática en el sistema (crones) ejecutándose y que hayan sido implantados por el intruso. Además, si está inundando de tráfico la red o se encuentra atacando a otro sistema, de igual forma, si existe relación de confianza con otras máquinas procurar que ya no se propague el daño hacia ellas. Hay que considerar que al momento de delimitar el daño puede traer como consecuencia que se pierda evidencia, porque puede que se tengan que matar procesos, impedir que se ejecute el cron, desconectar el cable de red o apagar el sistema, todo esto puede ocasionar pérdida de información para nosotros.

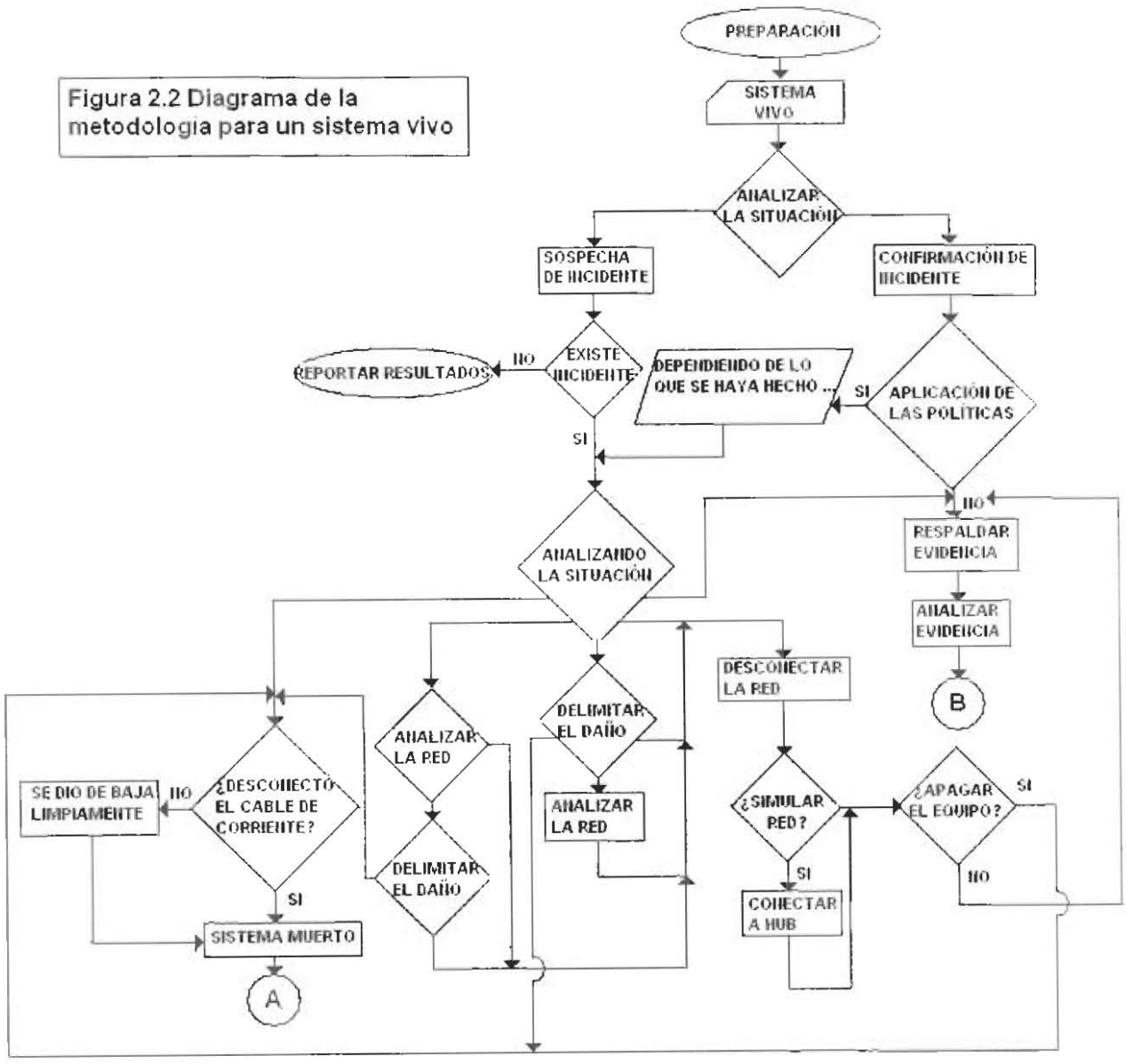
Respaldar evidencia

Tomar la decisión de respaldar toda evidencia puede repercutir exitosamente en la recolección de datos así como obtener toda prueba que sirva para comprobar y responder a las preguntas de quién, cómo, cuándo, dónde. Pero, puede perjudicar gravemente al sistema o sistemas de la subred si no delimitamos el daño de inmediato. Aquí se llega al punto de hacer una copia fiel del sistema en producción sobre otro disco duro obteniendo las imágenes tanto de datos del disco duro como de memoria que se está ocupando en ese momento, también respaldar procesos, puertos abiertos, transferencia de datos.

Una vez que se ha obtenido toda evidencia, procedemos a analizar los datos recopilados, para poder llegar a una conclusión de lo que en verdad pasó. A partir de aquí llevamos los mismos pasos a seguir como si fuera un sistema muerto. Tanto en la figura 2.1 como en la figura 2.2 colocamos los conectores adecuados para así relacionar esta parte del análisis de la evidencia.

En la figura 2.2 se muestra el diagrama de la metodología que recomendamos utilizar cuando el sistema se encuentra arriba.

Figura 2.2 Diagrama de la metodología para un sistema vivo



Los conectores "A" y "B" están haciendo referencia a la figura 2.1

CAPÍTULO III

DESCRIPCIÓN DE LA INVESTIGACIÓN FORENSE

3.1 Sistemas de detección de intrusos (IDS)

Llamaremos intrusión a un conjunto de acciones que intentan comprometer la integridad, confidencialidad o disponibilidad de un recurso[33]. A los sistemas utilizados para detectar las intrusiones o los intentos de intrusión se les denomina sistemas de detección de intrusiones (Intrusion Detection Systems, IDS); cualquier mecanismo de seguridad con este propósito puede ser considerado un IDS, pero generalmente sólo se aplica esta denominación a los sistemas automáticos (software o hardware), de manera que los IDS o Sistemas de Detección de Intrusos:

- buscan patrones previamente definidos que impliquen cualquier tipo de actividad sospechosa o maliciosa sobre nuestra red o host.
- aportan a nuestra seguridad una capacidad de prevención y de alerta anticipada ante cualquier actividad sospechosa. No están diseñados para detener un ataque, aunque sí pueden generar ciertos tipos de respuesta ante éstos.
- aumentan la seguridad de nuestro sistema, vigilan el tráfico de nuestra red, examinan los paquetes analizándolos en busca de datos sospechosos y detectan las primeras fases de cualquier ataque como pueden ser el análisis de nuestra red, barrido de puertos, etc.

Arquitectura de un IDS

Normalmente la arquitectura de un IDS está formada por:

1. La fuente de recogida de datos. Estas fuentes pueden ser un log, dispositivo de red, o como en el caso de los IDS basados en host, el propio sistema.
2. Reglas que contienen los datos y patrones para detectar anomalías de seguridad en el sistema.
3. Filtros que comparan los datos espiados de la red o de logs con los patrones almacenados en las reglas.
4. Detectores de eventos anormales en el tráfico de red.

[33] Cinthia Reyes Quezada y Sergio Rodríguez Gutiérrez, "Tesis: FUNDAMENTOS DE SEGURIDAD DE LA INFORMACIÓN"(Licenciatura en Ingeniería en Computación). México, D.F., Universidad Nacional Autónoma de México, Facultad de Ingeniería, 2001

5. Dispositivo generador de informes y alarmas. En algunos casos con la sofisticación suficiente como para enviar alertas via mail, o SMS (Short Messaging Service).

Snort[33], por ejemplo, tiene una arquitectura dividida en tres subsistemas:

a. Decodificador de paquetes
b. Motor de detección
c. Loggins y alertas

3.1.1 La razón de los IDS

Una de las primeras cosas que se debe plantear a la hora de hablar de IDS es si realmente se necesita uno de ellos en el entorno de trabajo; a fin de cuentas, se debe tener ya un sistema de protección perimetral basado en alguna herramienta de seguridad específica para control de acceso como lo puede ser un Firewall; si éste fallara, cada sistema habría de estar configurado de manera correcta, de forma que incluso sin el firewall cualquier máquina pudiera seguirse considerando relativamente segura.

Se debe esperar que en cualquier momento alguien consiga romper la seguridad de un entorno informático, y por tanto hemos de ser capaces de detectar ese problema tan pronto como sea posible (incluso antes de que se produzca, cuando el potencial atacante se limite a probar suerte contra las máquinas). Ningún sistema informático puede considerarse completamente seguro, pero incluso aunque nadie consiga violar las políticas de seguridad, los sistemas de detección de intrusos se encargarán de mostrar todos los intentos de multitud de intrusos para penetrar en nuestro entorno, no dejándonos caer en ninguna falsa sensación de seguridad: si somos conscientes de que a diario hay gente que trata de romper nuestros sistemas, no caeremos en la tentación de pensar que las máquinas están seguras porque nadie sabe de su existencia o porque no son interesantes para un intruso.

Sin importar qué sistemas vigile o su forma de trabajar, cualquier sistema de detección de intrusos ha de cumplir algunas propiedades para poder desarrollar su trabajo correctamente. En primer lugar, y quizás como característica más importante, el IDS ha

[33] Véase Apéndice A

de ejecutarse continuamente sin que nadie esté obligado a supervisarlo; independientemente de que al detectar un problema se informe a un operador o se lance una respuesta automática, el funcionamiento habitual no debe implicar interacción con un humano.

Otra propiedad, y también como una característica a tener siempre en cuenta, es la aceptabilidad o grado de aceptación del IDS; al igual que sucedía con cualquier modelo de autenticación, los mecanismos de detección de intrusos han de ser aceptables para las personas que trabajan habitualmente en el entorno. Por ejemplo, no ha de introducir una sobrecarga considerable en el sistema, ni generar una cantidad elevada de falsos positivos (detección de intrusiones que realmente no lo son) o de logs, ya que entonces llegará un momento en que nadie se preocupe de comprobar las alertas emitidas por el detector.

Una tercera característica a evaluar al momento de tener que tomar una decisión respecto a la instalación y uso de sistemas de detección de intrusos es la adaptabilidad del mismo a cambios en el entorno de trabajo. Como todos sabemos, ningún sistema informático puede considerarse estático: desde la aplicación más pequeña hasta el propio kernel de Unix, pasando por supuesto por la forma de trabajar de los usuarios, todo cambia con una periodicidad más o menos elevada. Si los mecanismos de detección de intrusos no son capaces de adaptarse rápidamente a esos cambios, están condenados al fracaso.

Los sistemas de detección de intrusos no son precisamente nuevos: el primer trabajo sobre esta materia data de 1980; no obstante, este es uno de los campos que tienen más auge desde hace ya unos años dentro de la seguridad informática.

3.2 Técnicas de detección

3.2.1 Detección de patrones anómalos

Desde que en 1980 James P. Anderson propusiera la detección de anomalías como un método válido para detectar intrusiones en sistemas informáticos, la línea de investigación más activa (esto es, la más estudiada, pero no por ello la más extendida en entornos reales) es la denominada Anomaly Detection IDS, IDS basada en la detección de anomalías. La idea es muy interesante: estos modelos de detección conocen lo que es

"normal" en nuestra red o nuestras máquinas a lo largo del tiempo, desarrollando y actualizando conjuntos de patrones contra los que comparar los eventos que se producen en los sistemas. Si uno de esos eventos (por ejemplo, una trama procedente de una máquina desconocida) se sale del conjunto de normalidad, automáticamente se cataloga como sospechoso.

Los IDS basados en detección de anomalías se basan en la premisa de que cualquier ataque o intento de ataque implica un uso anormal de los sistemas. Pero, ¿cómo puede un sistema conocer lo que es y lo que no es `normal' en el entorno de trabajo? Para conseguirlo, existen dos grandes aproximaciones: o es el sistema el que es capaz de aprenderlo por sí mismo (basándose por ejemplo en el comportamiento de los usuarios, de sus procesos, del tráfico de nuestra red...) o bien se le especifica al sistema dicho comportamiento mediante un conjunto de reglas. La primera de estas aproximaciones utiliza básicamente métodos estadísticos (medias, varianzas...), aunque también existen modelos en los que se aplican algoritmos de aprendizaje automático; la segunda aproximación consiste en especificar mediante un conjunto de reglas los perfiles de comportamiento habitual basándose en determinados parámetros de los sistemas (con la dificultad añadida de decidir cuáles de esos parámetros que con mayor precisión delimitan los comportamientos intrusivos).

En el primero de los casos (el basado en métodos estadísticos), el detector observa las actividades de los elementos del sistema, activos - sujetos -, pasivos - objetos o ambos, y genera para cada uno de ellos un perfil que define su comportamiento; dicho perfil es almacenado en el sistema, y se actualiza con determinada frecuencia envejeciendo la información más antigua y priorizando la más fresca.

El comportamiento del usuario en un determinado momento se guarda temporalmente en otro perfil, denominado "perfil actual" (current profile), y a intervalos regulares se compara con el almacenado previamente en busca de desviaciones que puedan indicar una anomalía.

1. Intensidad de la actividad. Reflejan el ratio de progreso de la actividad en el sistema, para lo cual recogen datos a intervalos muy pequeños - típicamente entre un minuto y una hora -. Estas medidas detectan ráfagas de comportamiento (por

ejemplo, una excesiva generación de peticiones de entrada/salida en un cierto intervalo) que en espacios de tiempo más amplios no podrían ser detectadas.

2. Numéricas. Se trata de medidas de la actividad cuyo resultado se puede representar en forma de valor numérico, como el número de archivos leídos por cierto usuario en una sesión o la cantidad de veces que ese usuario se ha equivocado al teclear su contraseña de acceso al sistema.
3. Categóricas. Las medidas categóricas son aquellas cuyo resultado es una categoría individual, y miden la frecuencia relativa o la distribución de una actividad determinada con respecto a otras actividades o categorías; por ejemplo, cual es la relación entre la frecuencia de acceso a un determinado directorio del sistema en comparación con la de acceso a otro. Seguramente la palabra "categoría" no es la más afortunada (por lo menos, no la más clara), ya que bajo este término se pueden englobar tanto a objetos (por ejemplo, archivos) como a eventos (por ejemplo, llamadas a la función crypt()) del sistema; esta definición genérica puede resultar más sencilla si distinguimos entre categorías globales e individuales: podemos entender las categorías globales como acciones muy genéricas dentro de un entorno, mientras que las categorías individuales serían la particularización para un elemento determinado del sistema. Así, una categoría global puede ser la formada por el conjunto de accesos remotos a la máquina, mientras que una individual sería la formada por los accesos desde una determinada ubicación física.
4. Distribución de registros de auditoría. Esta medida analiza la distribución de las actividades generadas en un recientemente basándose en los logs generados por las mismas; dicho análisis se realiza de forma ponderada, teniendo más peso las actividades más recientes, y es comparado con un perfil de actividades "habituales" previamente almacenado, de forma que permite detectar si recientemente se han generado eventos inusuales.

La segunda aproximación a la que antes hemos hecho referencia era la consistente en indicar mediante un conjunto de reglas el comportamiento habitual del sistema; suele ser denominada detección de anomalías basada en especificaciones (specification-based anomaly detection), que fue propuesta y desarrollada inicialmente por Calvin Cheuk Wang Ko y otros investigadores de la Universidad de California en Davis, durante la segunda mitad de los noventa. La idea en la que se sustentan los sistemas de

detección de anomalías basados en especificaciones es que se puede describir el comportamiento “deseable” (entendiendo por “deseable” el comportamiento “normal”) de cualquier programa cuya seguridad sea crítica; esta descripción se realiza en base a una especificación de seguridad mediante gramáticas, y se considera una violación de la seguridad (al menos en principio) a las ejecuciones de dichos programas que violen su respectiva especificación.

La idea de los sistemas de detección de intrusos basados en la detección de anomalías es realmente atractiva; no obstante, existen numerosos problemas a los que estos mecanismos tratan de hacer frente. En primer lugar podemos pararnos a pensar en las dificultades que existen a la hora de “aprender” o simplemente especificar lo habitual: si alguien piensa que por ejemplo obtener un patrón de tráfico “normal” en una red es fácil, se equivoca; quizás establecer un conjunto de procesos habituales en una única máquina resulte menos complicado, pero tampoco se trata de una tarea trivial. Además, conforme aumentan las dimensiones de los sistemas (redes con un gran número de máquinas interconectadas, equipos con miles de usuarios...) estos se hacen cada vez más aleatorios e impredecibles.

Otro gran problema de los sistemas basados en detección de anomalías radica en la política de aprendizaje que éstos sigan; si se trata de esquemas donde el aprendizaje es rápido, un intruso puede generar eventos para conseguir un modelo distorsionado de lo “normal” antes de que el responsable de los sistemas se percate de ello, de forma que el IDS no llegue a detectar un ataque porque lo considera algo “habitual”. Si por el contrario el aprendizaje es lento, el IDS considerará cualquier evento que se aleje mínimamente de sus patrones como algo anómalo, generando un gran número de falsos positivos (falsas alarmas), que a la larga harán que los responsables de los sistemas ignoren cualquier información proveniente del IDS, con los evidentes riesgos que esto implica.

3.2.2 Detección de usos indebidos (Firmas)

Los detectores de usos indebidos analizan la actividad del sistema buscando eventos que coincidan con un patrón predefinido o firma que describe un ataque conocido.

Dentro de la clasificación de los sistemas de detección de intrusos en base a su forma de actuar, la segunda gran familia de modelos es la formada por los basados en la detección de usos indebidos. Este esquema se basa en especificar de una forma más o menos formal las potenciales intrusiones que amenazan a un sistema y simplemente esperar a que alguna de ellas ocurra; para conseguirlo existen cuatro grandes aproximaciones: los sistemas expertos, los análisis de transición entre estados, las reglas de comparación y emparejamiento de patrones y la detección basada en modelos.

Los primeros sistemas de detección de usos indebidos, como NIDS (Net IDS), se basaban en los sistemas expertos para realizar su trabajo; en ellos las intrusiones se codifican como reglas de la base de conocimiento del sistema experto, de la forma genérica if-then (if CONDICIÓN then ACCIÓN). Cada una de estas reglas puede detectar eventos únicos o secuencias de eventos que denotan una potencial intrusión.

La segunda implementación de los sistemas de detección de usos indebidos es la basada en los análisis de transición entre estados; bajo este esquema, una intrusión se puede contemplar como una secuencia de eventos que conducen al atacante desde un conjunto de estados inicial a un estado determinado, representando este último una violación consumada de nuestra seguridad. Cada uno de esos estados no es más que una imagen de diferentes parámetros del sistema en un momento determinado, siendo el estado inicial el inmediatamente posterior al inicio de la intrusión, y el último de ellos el resultante de la completitud del ataque; la idea es que si conseguimos identificar los estados intermedios entre ambos, seremos capaces de detener la intrusión antes de que se haga efectiva.

El sistema de detección utiliza además una base de datos, (realmente se trata de simples archivos planos) formada principalmente por dos tablas, una donde se almacenan las descripciones de los diferentes estados (SDT, State Description Table) y otra en la que se almacenan las transiciones entre estados que denotan un potencial ataque (SAT, Signature Action Table). Al registrarse una sucesión determinada de eventos que representen un ataque entrará en juego el motor de decisiones, que emprenderá la acción que se le haya especificado (desde un simple mensaje en consola informando de la situación hasta acciones de respuesta automática capaces de interferir en tiempo real con la intrusión).

La tercera implementación que habíamos comentado era la basada en el uso de reglas de comparación y emparejamiento de patrones o pattern matching; en ella, el detector se basa en la premisa de que el sistema llega a un estado comprometido cuando recibe como entrada el patrón de la intrusión, sin importar el estado en que se encuentre en ese momento. Dicho de otra forma, simplemente especificando patrones que denoten intentos de intrusión, el sistema puede ser capaz de detectar los ataques que sufre, sin importar el estado inicial en que esté cuando se produzca dicha detección, lo cual suele representar una ventaja con respecto a otros modelos de los que hemos comentado.

Actualmente muchos de los sistemas de detección de intrusos más conocidos (por poner un ejemplo, podemos citar a SNORT o RealSecure) están basados en el pattern matching. Utilizando una base de datos de patrones que denotan ataques, estos programas se dedican a examinar todo el tráfico que ven en su segmento de red y a comparar ciertas propiedades de cada trama observada con las registradas en su base de datos como potenciales ataques; si alguna de las tramas empareja con un patrón sospechoso, automáticamente se genera una alarma en el registro del sistema.

Por último, tenemos que hablar de los sistemas de detección de intrusos basados en modelos; se trata de una aproximación conceptualmente muy similar a la basada en la transición entre estados, en el sentido que contempla los ataques como un conjunto de estados y objetivos, pero ahora se representa a los mismos como escenarios en lugar de hacerlo como transiciones entre estados. En este caso se combina la detección de usos indebidos con una deducción o un razonamiento que concluye la existencia o inexistencia de una intrusión; para ello, el sistema utiliza una base de datos de escenarios de ataques, cada uno de los cuales está formado por una secuencia de eventos que conforman el ataque. En cada momento existe un subconjunto de esos escenarios, denominado de escenarios activos, que representa los ataques que se pueden estar presentando en el entorno; un proceso denominado anticipador analiza los registros de auditoria generados por el sistema y obtiene los eventos a verificar en dichos registros para determinar si la intrusión se está o no produciendo. El anticipador también actualiza constantemente el conjunto de escenarios activos, de manera que este estará siempre formado por los escenarios que representan ataques posibles en un determinado momento y no por la base de datos completa.

Los IDS basados en la detección de usos indebidos son en principio más robustos que los basados en la detección de anomalías: al conocer la forma de los ataques, es teóricamente extraño que generen falsos positivos (a no ser que se trate de un evento autorizado pero muy similar al patrón de un ataque); es necesario recalcar ésto porque la generación de falsos positivos es un problema a la hora de implantar cualquier sistema de detección. No obstante, en este mismo hecho radica su debilidad: sólo son capaces de detectar lo que conocen, de forma que si alguien nos lanza un ataque desconocido para el IDS éste no nos notificará ningún problema, es conveniente mantener al día la base de datos de los IDS basados en detección de usos indebidos. Aún así, seremos vulnerables a nuevos ataques.

Otro grave problema de los IDS basados en la detección de usos indebidos es la incapacidad para detectar patrones de ataque convenientemente camuflados, un atacante puede evitar al sistema de detección de intrusos sin más que insertar espacios en blanco o rotaciones de bits en ciertos patrones del ataque; aunque algunos IDS son capaces de identificar estas transformaciones en un patrón, otros muchos no lo hacen.

3.3 Tipos de IDS

3.3.1 Clasificación de IDS

Según sus características es posible clasificarlos en 3 tipos:

3.3.1.1 HIDS (Host IDS)

Protege contra un único Servidor, PC o host. Monitorizan gran cantidad de eventos, analizando actividades con una gran precisión, determinando de esta manera qué procesos y usuarios se involucran en una determinada acción. Recaban información del sistema como archivos, logs, recursos, etc, para su posterior análisis en busca de posibles incidencias. Todo ello en modo local, dentro del propio sistema. Fueron los primeros IDS en desarrollar por la industria de la seguridad informática.

3.3.1.2 NIDS (Net IDS)

Protege un sistema basado en red. Actúan sobre una red capturando y analizando paquetes de red. Luego analizan los paquetes capturados, buscando patrones que supongan algún tipo de ataque.

Bien ubicados, pueden analizar grandes redes y su impacto en el tráfico suele ser pequeño. Actúan mediante la utilización de un dispositivo de red configurado en modo promiscuo (analizan,"ven" todos los paquetes que circulan por un segmento de red aunque estos vayan dirigidos a un determinado equipo). Analizan el tráfico de red, normalmente, en tiempo real. No sólo trabajan a nivel TCP/IP, también lo pueden hacer a nivel de aplicación.

3.3.1.3 DNIDS (distributed NIDS)

Este tipo de IDS, más que proteger, monitoriza la actividad entre varias redes. Tiene una visión global. El carácter distribuido dota al sistema de la escalabilidad y adaptabilidad necesarias para que pueda ajustarse a las necesidades de rendimiento de cualquier red.

Una característica deseable actualmente en los Sistemas de Detección de Intrusiones basados en Red, es que sean adaptables a distintos tipos de redes (en topología y tamaño) y que sean capaz de evolucionar con la red en la que son implantados, pudiendo adaptarse a sus necesidades crecientes (escalabilidad: dimensiones de la red, número de usuarios, dominios de colisión, velocidad de transmisión, poder migrar de 10 Mbps a 100 Mbps ó más). En ese aspecto la arquitectura mas adecuada es sin duda la distribuida.

Otra forma de clasificar los IDS sería por el tipo de respuesta:

3.3.1.4 Pasivos

Son aquellos IDS que notifican a la autoridad competente o administrador de la red mediante el sistema que sea, alerta, etc.; pero no actúan sobre el ataque o atacante.

3.3.1.5 Activos

Generan algún tipo de respuesta sobre el sistema atacante o fuente de ataque como cerrar la conexión o enviar algún tipo de respuesta predefinida en nuestra configuración.

3.4 HoneyPots y HoneyNets

3.4.1 Definición de Honeypot

Es un recurso de cómputo diseñado para capturar todo el tráfico y actividad del sistema[34]. Cuenta con servicios de red comunes en conjunción con mecanismos de captura de tráfico de red. Casi todos son diseñados para registrar y monitorear intrusos. Son diferentes de los sistemas regulares de una red, ya que estos cuentan con mecanismos de registro y control de servicios.

El objetivo es que los honeypots aparenten ser sistemas normales de producción, los cuales se encuentran proporcionando algún servicio. Pero en realidad son sistemas emulando a una cantidad de servicios y vulnerabilidades.

3.4.1.1 Propósito

Su propósito es capturar las actividades de los intrusos sin que ellos tengan conocimiento de que están siendo monitoreados y registrados. Esto no implica que un honeypot tenga el propósito de capturar intrusos.

Los honeypots pueden ayudar a una organización a mejorar sus mecanismos de seguridad ya que pueden mostrar de manera fácil la cantidad de amenazas circulando por Internet, además pueden ayudar a contrarrestar algunos ataques o bien a alimentar a otros mecanismos de protección como los son los IDS o Firewalls al proporcionar información sobre distintas amenazas de seguridad en cómputo.

Un honeypot adquiere valor una vez que ha sido atacado o comprometido. Es muy importante aclarar que los honeypots no son una solución a nuestros problemas de

[34] Línea de Especialización de Detección de Intrusos y Tecnologías HoneyPots. DGSCA-UNAM,2003.

seguridad ni mucho menos arreglarán nada, únicamente son herramientas que nos ilustrarán, dependiendo de lo que se quiera registrar, las debilidades de nuestros sistemas para tomar medidas al respecto.

La razón más importante por la cual incluimos un honeypot en nuestra tesis es porque un honeypot sirve como evidencia para el análisis forense.

3.4.1.2 Tipos de Honeypots

Los honeypots se clasifican dentro de dos grandes categorías definidas por Marty Roesh:

- Producción
- Investigación

Un honeypot de producción agrega valor a las medidas de seguridad de una organización. Estos serán implementados en los sistemas de producción con la finalidad de prevenir ataques a los servicios de producción, detectar posibles ataques y reaccionar a ellos. Muchos de estos honeypots emulan algunos servicios dentro del sistema de producción, de esta manera un intruso no conoce con certeza cuáles son los servicios de producción emulados y cuáles no. Algunos otros pueden emular cientos de sistemas y vulnerabilidades dificultando cada vez más la tarea de los intrusos.

Para las organizaciones que no tengan dentro de sus expectativas la investigación de amenazas de seguridad y deseen utilizar honeypots, pueden utilizar honeypots de producción, pero esto es un riesgo, ya que en caso de ser dañado o alterado afectará el ambiente de la organización, además la investigación de amenazas no se puede realizar por la actividad de producción del sistema.

En general la principal función del honeypot de producción será la prevención de ataques a servicios reales, la detección de ataques a los servicios emulados y en algunos casos reaccionarán ante los ataques de los servicios.

La segunda categoría honeypot agrupa a los honeypots diseñados para obtener información acerca de los intrusos. Estos honeypots no agregan directamente valor específico a la organización.

Por siglos, organizaciones militares se han enfocado a obtener información para comprender y protegerse mejor en contra del enemigo. Para defenderse en contra de una amenaza se tiene que conocer primero acerca de ésta. Sin embargo, en el mundo de la seguridad en cómputo se tiene muy poca de esta información.

Los honeypots pueden agregar valor a la investigación, ya que pueden proporcionar una plataforma de estudio de las amenazas. La mejor manera de aprender acerca de los intrusos es observarlos en acción para registrar paso a paso cómo un sistema es atacado y comprometido.

En la tabla 3.1 se describen las características de los tipos de honeypots.

Tipos de Honeypots	
Investigación	**Producción**
• No agregan directamente un valor específico a la organización. • Son utilizados para la investigación de amenazas que permitan a las organizaciones protegerse mejor en contra de estas. • Difíciles de mantener y utilizar. Capturan gran cantidad de información.	• Su propósito es ayudar a reducir los riesgos en una organización. • Generalmente tienen poca funcionalidad y son fáciles de utilizar. • Normalmente proporcionan poca información.

Tabla 3.1 Tipos de Honeypots

3.4.1.3 ¿Qué se puede obtener?

Con el uso de los honeypots se pueden obtener los comandos ejecutados por los intrusos para ingresar al sistema, así como los comandos que los intrusos ejecutan una ves que se han adueñado del mismo, además es un comportamiento típico de los intrusos que se comuniquen con otros intrusos desde los sistemas recién vulnerados, por lo cual se pueden capturar sesiones de chat las cuales ayudan al profesional de intrusiones a definir el comportamiento de dicho intruso así como sus propósitos para con el sistema.

En algunas ocasiones se pueden obtener imágenes o fotografías que los intrusos intercambian entre sí, con esto inclusive se puede conocer al intruso que ha atacado el sistema. Una parte importante de lo que se puede llegar a obtener es el ataque de nuevos o desconocidos exploits, una vez que se sabe como fue vulnerado un sistema se puede estudiar el método de ataque y alertar a la comunidad sobre un nuevo o desconocido ataque, además si el honeypot es vulnerado dentro del ambiente en producción en donde se encuentre, puede servir como anticipador para estar alertas con los demás sistemas que no son honeypots ya que probablemente también serán atacados o ya están comprometidos.

3.4.1.4 Valor de los honeypots

Este es un dispositivo diseñado para ser comprometido, no proporciona servicios de producción. Esto significa que casi no hay tráfico de producción entrando o saliendo del dispositivo. Si en cualquier momento una conexión se inicia desde el honeypot esto significará que el sistema ha sido comprometido. Como todo el tráfico de producción está fuera del honeypot, todo el tráfico es sospechoso por naturaleza.

Estos sistemas no están limitados a un propósito específico. El valor y problemas que ayuden a resolver dependerá de cómo se implementen y utilicen, es decir, el entorno de seguridad que se tenga en la organización en donde se deseé establecer el honeypot definirá en gran medida la configuración e implementación del mismo, ya que el entorno de seguridad provee información valiosa de vulnerabilidades y recursos de la organización, por otra parte, el equipo de investigadores en seguridad deberán establecer políticas que regulen sus investigaciones para no afectar a terceros, esta políticas deberán estar contempladas dentro de las políticas de seguridad de la organización

3.4.1.5 Ventajas de los honeypots

Los honeypots coleccionan gran cantidad de información, normalmente de alto valor. Esta información hace que los falsos positivos disminuyan, hace mucho más fácil la recolección de información y almacenamiento de datos. Los honeypots pueden dar la información exacta que se necesita de manera rápida y fácil, en un formato comprensible. Esta información también es normalmente de alto valor, no solamente se puede ver la

actividad de la red, además se podrá conocer que es lo qué el intruso hace una vez que ha ingresado en el sistema.

Muchas herramientas de seguridad en cómputo pueden ser superadas por el ancho de banda o actividad de la red, esto puede provocar ataques potenciales o disminución de registro de la actividad de la red. Los honeypots no tienen este problema, únicamente capturan todo el tráfico que reciben, de esta manera, no pueden ser superados por el ancho de banda, ni por la actividad de la red, ya que pueden registrar todos los eventos del sistema. Son fáciles de implementar en ambientes donde se encuentre una alta actividad de red.

Conceptualmente los honeypots son extremadamente simples, están basados en conceptos sencillos, esto ayuda a reducir la complejidad y al mismo tiempo reduce el riesgo. Muchos mecanismos de seguridad en cómputo son complejos, por lo tanto son difíciles de entender y mantener.

Al detectar actividad no autorizada dentro del ambiente de desarrollo del honeypot estos comprueban su valor mucho más rápido que algunos dispositivos como Firewalls o IDS que en ocasiones no sabemos si funcionan adecuadamente o están bien configurados.

3.4.1.6 Desventajas de los Honeypots

Los honeypots cuentan con una desventaja crítica: únicamente registrarán la actividad de red que sea enviada a ellos. Los honeypots carecerán de cualquier prueba, escaneo o ataque que no le sea enviado directamente a ellos. Por ejemplo, puede haber un ataque masivo dirigido a la red completa pero menos al honeypot, al ocurrir esto se puede pensar que el honeypot tendrá información sobre este ataque, más sin embargo el honeypot carecerá de información puesto que únicamente capturará aquella actividad que sea dirigida a él.

También, los intrusos pueden utilizar a los honeypots para sus propósitos, además pueden evitar estos sistemas, o peor aún, introducir datos falsos. Al hacer esto, los intrusos pueden dirigir sus ataques a aquellos sistemas que no son honeypots y no perder el tiempo con sistemas que no son reales.

Por otra parte, los honeypots si no son atacados no tienen ningún valor. Pueden llevar a cabo cosas interesantes, pero si ningún intruso envía algún paquete al honeypot, este carecerá de cualquier actividad no autorizada y por lo tanto no tendrá ninguna utilidad porque no proporciona ninguna evidencia o elemento sobre algún ataque dado, más sin embargo puede ser una buena medida para definir un nivel de riesgo dentro de la institución ya que si no se presentan ataques tal vez sea porque los demás mecanismos de protección están funcionando de manera adecuada.

Los honeypots pueden introducir riesgos al ambiente. El riesgo es variable, dependiendo de cómo se construya y desarrolle el honeypot. Es decir, el nivel de interacción de los honeypots es distinto, por lo cual los honeypots tienen diferentes niveles de riesgo. Algunos introducen un riesgo muy pequeño, mientras otros proporcionan al intruso plataformas enteras en las cuales pueda planear nuevos ataques.

Es por estas desventajas que los honeypots no reemplazan a otros mecanismos de seguridad. Los honeypots pueden únicamente agregar valor para el trabajo con existencia de estos mecanismos de seguridad.

3.4.1.7 Prevención

El engaño puede ser utilizado como un método para detectar intrusos. El concepto consiste en mantener a los intrusos gastando su tiempo y recursos atacando honeypots, esto sería lo opuesto al atacar sistemas de producción, que si afectarían a una organización.

Los honeypots agregan un pequeño valor a la prevención, no van a mantener a los intrusos fuera. Para que se mantenga a los intrusos fuera se requieren mejores prácticas, tales como: deshabilitar servicios no necesarios o inseguros, actualizando o corrigiendo aquello que lo requiera y utilizando mecanismos fuertes de autenticidad. Estas son las mejores prácticas y procedimientos con los cuales se puede mantener fuera a los intrusos.

Un honeypot es un sistema que será comprometido. Una implementación incorrecta del honeypot puede facilitar el ingreso a un intruso y por consiguiente una propagación de ataques. Cuando el intruso ha ingresado al honeypot se protegen los recursos de producción de la organización de un posible ataque.

Para varias organizaciones es mucho mejor gastar su limitado tiempo y recursos en la seguridad de sus sistemas, en vez invertirlo en el engaño. El engaño puede contribuir a la prevención, pero obtendrán mejores resultados colocando simultáneamente mejores prácticas de seguridad en cómputo.

3.4.1.8 Detección

Cuando los honeypots agregan valor a la prevención, también agregan valor a la detección. Para muchas organizaciones, es extremadamente difícil detectar ataques a sus sistemas de cómputo. A menudo las organizaciones son sobresaturadas por la actividad de producción, lo cual equivale a giga bytes de datos del sistema de bitácoras, esto hace que sea demasiado difícil detectar cuándo un sistema es atacado o cuándo un sistema ha sido comprometido. Los Sistemas Detectores de Intrusos pueden ser sobresaturados con falsos positivos[35]. El problema aquí es que el administrador puede recibir muchas alertas diariamente por lo que no podrá responder a todas ellas. También los Sistemas Detectores de Intrusos a menudo se pueden condicionar a ignorar este tipo de alertas falso positivos, ya que son muy frecuentes. Muchos de los sensores de los Sistemas Detectores de Intrusos dependen de las alertas para que los ataques sean fallidos, a no ser que estos falsos positivos sean reducidos.

Los honeypots nunca tendrán falso positivos, solamente tendrán pocos comparados con la mayoría de las implementaciones IDS.

3.4.1.9 Reacción

A menudo cuando un sistema dentro de una organización es comprometido hay mucha actividad de producción después de que ocurrió el incidente de seguridad, por lo cual los datos del sistema son afectados y la evidencia del ataque es contaminada por la actividad del equipo. Un equipo de respuesta a incidentes no puede determinar exactamente que ha pasado, la evidencia es mucho más difícil de obtener en dichos ambientes.

El segundo desafío después de un incidente, es la posición de muchas organizaciones, ya que frecuentemente los sistemas comprometidos no se detienen. Los

[35] Los falsos positivos son alertas que son generadas cuando un censor reconoce una firma configurada como un ataque, pero en realidad sólo fue tráfico normal

servicios de producción que ofrecen no pueden ser alterados, por lo que un equipo de respuesta a incidentes no podrá conducir un análisis forense completo.

Aún cuando no es comúnmente considerado, los honeypots también agregan valor a la reacción. Los honeypots pueden reducir o eliminar los problemas anteriormente descritos, porque con frecuencia son sistemas con poca contaminación de datos y que puede ser puesto fuera de servicio, por lo tanto se puede realizar un análisis forense completo a dicho sistema.

3.4.1.10 Nivel de Interacción

El nivel de interacción es un concepto creado para comprender mejor las habilidades de los diferentes honeypots. Es como se mide la funcionalidad que un honeypot proporciona a un intruso[36].

El nivel de interacción dependerá de lo que se quiera registrar. Los niveles de interacción tienen diferentes ventajas y desventajas. Esto puede ser crítico al decidir qué tipo de honeypot se desea y cómo se desarrollará éste.

Podemos tener tres niveles de interacción distintos: Bajo, Medio y Alto. Entre mayor sea la funcionalidad de un honeypot, serán más las actividades que un intruso podrá realizar y más información podrá ser obtenida de esto. Sin embargo, para que esto ocurra, un intruso deberá realizar mayor actividad con el honeypot y mayor será el daño potencial que un intruso podrá realizar.

- Nivel de Interacción Bajo: Una baja interacción con el honeypot reduce el riesgo al mínimo. Este tipo de honeypot es fácil de instalar, configurar, administrar y será sencillo emular algunos servicios comunes. La información es limitada, sin embargo, es poco lo que un intruso puede explotar ya que la interacción con el honeypot es casi nula.
- Nivel de Interacción Medio: Interactúa con el intruso al mínimo, ya que son enviadas respuestas a sus peticiones, estas respuestas pueden ser respuestas automáticas comunes. Es decir, con un nivel de interacción medio el honeypot puede ser capaz de responder a las distintas solicitudes que realice el intruso, con

[36] Véase la nota 34

esto es más el tiempo que el intruso invierte en el honeypot, puede interactuar un poco más y se puede llegar a analizar un poco más al intruso pero sin proporcionarle un servicio real o plataforma real.

- Nivel de Interacción Alto: Se cuenta con un gran riesgo, ya que el intruso puede comprometer el sistema y utilizar todos sus recursos. Se podrá aprender mucho más si es un sistema operativo actual, con el propósito de que el intruso lo comprometa e interactué. Por lo tanto, una alta interacción incluye un alto nivel de riesgo, ya que intruso tiene el sistema operativo reciente para trabajar, además puede interactuar con todos los recursos del sistema comprometido.

Ninguna de estas soluciones es un mejor honeypot. Todo esto dependerá de lo que se intente registrar. Se debe recordar que los honeypots no son una solución. En vez de esto, son una herramienta. Su valor dependerá de cuáles sean los objetivos en la investigación para una alerta y detección temprana.

3.4.2 Definición de Honeynet

Las honeynet son un tipo de honeypot de alta interacción, diseñado específicamente para la investigación de amenazas de seguridad, ayudan a conocer como los intrusos realizan pruebas y como consiguen explotar los recursos de una red[37].

Los honeypots son implementados en simples sistemas y su valor se encuentra en la detección de amenazas, en cambio las honeynets son implementadas en una red de sistemas de cómputo y su valor consiste en proporcionar información sobre amenazas de seguridad en cómputo.

3.4.2.1 Propósito

Las honeynets representan el extremo de la configuración del honeypot, se podrán aprender grandes cosas, sin embargo también cuentan con un alto riesgo. Su primer valor radica en la investigación, recogiendo información de amenazas que existen comúnmente hoy en día en Internet. Esto proporciona a los intrusos un completo rango de sistemas, aplicaciones y funcionalidades para atacar. No solamente se podrá aprender sus herramientas y tácticas sino además sus métodos de comunicación, organizaciones

[37] Véase la nota 34

grupales y motivos. Una gran variedad de medidas deberán de tomarse para asegurar que una vez comprometido el sistema, una honeynet no pueda ser utilizada para atacar a otros sistemas. Las honeynets son primeramente honeypots de investigación. Estas pueden ser utilizadas como honeypot de producción, específicamente para la detección o reacción, sin embargo, requerirá de mucho tiempo y esfuerzo. En la tabla 3.2 se muestran las características de los honeynets de investigación y producción.

Tipos de Honeynets	
Investigación Es una red de múltiples sistemas operativos dedicados exclusivamente a la investigación de amenazas de seguridad, es decir, los sistemas no tienen tráfico de producción, pero cuentan con servicios y aplicaciones reales.	**Producción** Los sistemas que se encuentran dentro de la red son sistemas de producción con aplicaciones y servicios que la organización está implementando en sus ambientes de producción.

Tabla 3.2 Tipos de Honeynets

La Honeynet de investigación se establece detrás de un dispositivo de control de acceso, en el cual todo el tráfico de entrada y salida es controlado y capturado. El tráfico capturado es analizado para conocer las herramientas tácticas y motivos de los intrusos.

La Honeynet de Producción se establece detrás de un dispositivo de control, el tráfico de salida es controlado. Los sistemas que se encuentran dentro de la red son sistemas de producción con aplicaciones y servicios que la organización está empleando en sus ambientes de producción. Los riesgos y vulnerabilidades descubiertas en una honeynet de este tipo son lo mismos que se pueden encontrar en cualquier organización en Internet. Solo es necesario tomar un sistema de producción y colocarlo dentro de la honeynet.

Con este tipo de honeynet es difícil realizar un buen estudio de las amenazas debido al tráfico de producción que contamina los datos. Pero puede ser utilizada como alerta de posibles ataques a los demás sistemas dentro de la organización. Pueden detectarse ataques y se puede establecer un patrón de comportamiento del intruso.

Objetivo

El Objetivo es conocer cómo trabajan los intrusos al comprometer una red, pero sin que tengan conocimiento de que están siendo observados y analizados.

Valor de las honeynets.

El primer propósito de una honeynet es recolectar información acerca de las amenazas existentes en el campo de la seguridad en cómputo. Con las honeynets se pueden detectar nuevos y sofisticados ataques.

Por lo general la información referente a la seguridad en cómputo es puramente defensiva, es decir, se proporciona información acerca de vulnerabilidades existentes, para una corrección adecuada se implementen mecanismos de defensa, métodos de cifrado, se realizan actualizaciones del sistema, etc. Todos estos mecanismos tienen por objetivo proteger lo mejor posible los recursos de la organización. Pero los intrusos desarrollan y realizan nuevos ataques a los sistemas de cómputo constantemente, esto es una gran desventaja, ya que un nuevo y peligroso ataque puede ser desarrollado y puede transcurrir un largo periodo de tiempo antes de que sea dado a conocer, durante este periodo de tiempo muchos sistemas pueden ser comprometidos por este ataque desconocido.

Las honeynets intentan cambiar esta situación al proporcionar a la organización la opción de tomar la iniciativa.

Al colocar sistemas de producción dentro de una honeynet se pueden identificar riesgos y vulnerabilidades para los sistemas de producción. Adicionalmente el equipo de respuesta puede desarrollar habilidades para detectar, reaccionar, recuperar y analizar sistemas honeypot que han sido comprometidos.

Funcionamiento

La honeynet conceptualmente es una estructura simple. Se crea una red de cómputo donde se podrá observar todo lo que ocurre dentro, se podrá monitorear a los intrusos en la red, se podrá incluir todo lo que se quiera dentro de la misma. Esta red monitoreada y controlada es la Honeynet.

Uno de los problemas de los profesionales de la seguridad es determinar dentro del tráfico de una red cuál es el tráfico de la actividad de los sistemas de producción y cual es el tráfico de actividad maliciosa provocada por un intruso, algunos mecanismos como IDS resuelven este problema utilizando bases de datos de firmas de ataques ya conocidos que son identificados dentro del tráfico de la red. Sin embargo, el exceso de información, la contaminación de los datos, actividad desconocida, falsos positivos y falsos negativos pueden causar que el análisis y determinación de la información sea extremadamente difícil.

Al igual que algunos honeypots, las honeynets resuelven este problema dedicando los recursos exclusivamente a la investigación de amenazas de seguridad, es decir, no hay tráfico de producción entrando y saliendo de la honeynet por lo cual todo el tráfico que llegue hasta la honeynet es sospechoso por naturaleza y será capturado y analizado.

Honeynet de primera generación

Una Honeynet de primera generación consta de varios elementos que permiten el monitoreo y el control de esta red.

Se permitiré que cualquier intruso comprometa los sistemas dentro de la honeynet, se le autoriza que pueda realizar algunas actividades dentro de los sistemas, pero se deberá controlar cualquier actividad maliciosa que intente comprometer a los sistemas que no son parte de la honeynet.

Requerimientos.

Los requerimientos críticos que definen a cada honeynet son el Control de Datos y Captura de Datos. Si ocurre una falla en cualquiera de estos requerimientos entonces ocurre una falla en la honeynet. Un tercer requerimiento es la Colección de Datos y se utiliza exclusivamente por aquellas organizaciones que cuenten con múltiples honeynets distribuidas dentro de sus ambientes.

Los honeynets pueden implementarse de diferentes maneras pero deberán de mantenerse los requisitos sin que los intrusos tengan conocimiento de estos, por lo tanto, no podrán saber que se encuentran dentro de una honeynet.

Control de Datos

El Control de Datos se encargará de contener la actividad de los intrusos dentro de la honeynet, se debe recordar que una honeynet es implementada para ser comprometida por los intrusos; cuando se trata con intrusos simplemente hay un gran riesgo, se deberá contar con mecanismos que permitan controlar la actividad de los intrusos y así disminuir este riesgo.

Para realizar esto se colocará un firewall que permitirá el ingreso de todo tráfico que se acredite, pero limitando el tráfico de salida.

Es importante definir cuál será el propósito de la honeynet. Por ejemplo, si es capturar y estudiar gusanos, deberá de permitir todo le trafico de entrada y no se deberá permitir ningún tráfico de salida, pero si lo que se quiere es estudiar a los intrusos deberá permitir cierto tráfico de salida, ya que es en ese tráfico en el cual los intrusos logran conseguir los recursos para llevar a cabo el ataque exitoso y continuar con sus propósitos cualquiera que estos sean.

Lo difícil de este requerimiento es implementarlo sin que los intrusos sospechen, ya que de no acreditar el suficiente tráfico de salida en una cuantos minutos el intruso abandonará la honeynet, sin embargo, si se acredita el tráfico suficiente para que una vez que el intruso a conseguido ingresar al sistema puede obtener las herramientas de su sitios web entonces podrá realizar lo que quiera en el sistema comprometido. Pero si el intruso intenta realizar un ataque desde el sistema comprometido, este deberá ser bloqueado. El control de datos consiste en implementar mecanismos que controlen el tráfico de entrada y de salida de la honeynet, es decir, se permitirá el ingreso de todo aquel tráfico que permita al intruso comprometer cualquier sistema dentro de la honeynet, pero se deberá bloquear la salida del tráfico que pueda ser utilizado para comprometer algún otro sistema fuera de la honeynet.

Captura de Datos.

La captura de datos es otro requerimiento crítico de la honeynet, será la encargada de capturar todo la actividad de los intrusos dentro dela honeynet, esta información es importante ya que será analizada para conocer las herramientas, tácticas y motivos de los intrusos.

El objetivo es capturar la mayor cantidad de información posible, para esto se hace uso de varios recursos, no se deberá depender de un solo recurso para la captura de datos, ya que un intruso podría eliminar la información de este recurso, lo cual es un comportamiento típico.

Al igual que el control, la captura de datos deberá ser implementada sin que los intrusos tengan conocimiento de que cada acción que realicen está siendo registrada.

El Firewall será el primer recurso de captura de datos, como se describió anteriormente, contará las conexiones de entrada y salida, así que deberá registrar toda aquella actividad de conexión que pase a través de él. Adicionalmente el firewall deberá notificar cuando alguna conexión sea establecida, pues significará entonces que se trata de un ataque o prueba a algún sistema dentro del honeynet.

El IDS deberá capturar toda la actividad de la red, capturando todo paquete que cruce dentro de la honeynet. El IDS se encuentra monitoreando la red de los sistemas honeypots y al mismo tiempo monitorea los elementos que serán utilizados para administrar la honeynet los cuales forman parte de la red administradora, cuando un ataque se está llevando a cabo. Con los registros obtenidos por el IDS se puede obtener información precisa sobre alguna conexión en específico.

El honeypot es otro recurso para obtener información valiosa registrando toda actividad que ocurra dentro de él hacia un sistema remoto, este servidor de registros capturará los registros de los sistemas honeypots, de esta manera si un intruso elimina los registros en el sistema comprometido no se habrá perdido nada, ya que dicha información se encuentra a salvo en otro sistema.

Se puede utilizar cualquier método que permita registrar todas las acciones del intruso dentro del sistema, redireccionando el sistema de bitácoras, se puede capturar los comandos ejecutados así como sus respectivas salidas, todo se deberá de implementar sin que los intrusos lo noten fácilmente, si llegaran a notarlo lo peor que puede pasar es que intenten eliminar sus huellas del sistema remoto, para lo cual deberán de hacer uso de mejores tácticas y herramientas, ya que este sistema remoto contará con una mejor protección que los que tienen los honeypots y algunas de las capas adicionales capturará este ataque.

Colección de Datos

El control de datos y la captura de datos son dos requerimientos para tecnologías honeypot. Cualquier organización que requiera implementar este tipo de tecnologías deberá respetar estos dos requerimientos.

La colección de datos es diferente, ésta es opcional. La colección de datos es la suma de datos de múltiples honeynets en un punto central. El objetivo es incrementar exponencialmente el valor de la información colectada.

Muchas organizaciones desarrollan únicamente una honeynet por lo cual la colección de datos no es necesaria. Pero algunas organizaciones como Honeynet Research Alliance la cual desarrolla múltiples honeynets, la colección de datos deberá de ser un estándar.

Honeynets de Segunda Generación

La honeynet de Segunda Generación tendrá todos los requerimientos combinados dentro de un dispositivo simple conocido como sensor honeynet. Esto significa que todos los requerimientos de control, captura y colección de datos tendrán que estar en un recurso simple. Esto hará mucho más fácil el manejo y desarrollo de una honeynet.

Este dispositivo simple está en capa 2 del modelo OSI (Enlace de datos). Esto permitirá que sea difícil de detectar, pues no tendrá una dirección IP asignada. Actuará como un puente, no hay un ruteo de tráfico de red. Todo el tráfico de entrada y salida pasará a través del dispositivo simple.

Con la tecnología de primera generación el control de datos es limitado ya que solo se permite cierto tráfico de entrada y salida, con la tecnología de segunda generación se podrá permitir mayor tráfico de entrada y salida si un ataque es ejecutado desde la honeynet hacia algún sistema afuera de la misma, este dispositivo deberá de ser capaz de bloquear el ataque, modificando los paquetes que sean enviados, el intruso podrá ver que su ataque es enviado pero no entenderá porqué no es exitoso. Se agregará la suficiente inteligencia para modificar lo bytes dentro del código de cualquier exploit ejecutado a través del dispositivo, así como reducir o eliminar conexiones enteras.

También tendrá la habilidad de falsificar respuestas para mantener una interacción alta con los intrusos.

El objetivo es darle al intruso mayor flexibilidad mientras proporciona un mayor control sobre las acciones de los mismos.

Este tipo de honeynet proporcionará mayor información sobre los intrusos, debido a que tendrá mejores mecanismos de captura de datos.

Con el mecanismo de cifrado cada vez más y más ataques con cifrados para mantener un mayor control sobre la víctima. La segunda generación contará con módulos del Kernel que son diseñados para capturar las actividades de los intrusos.

Honeynets Virtuales.

Las Honeynets Virtuales combinan todos los requerimientos dentro de un solo sistema físico, es decir, la captura de datos, control de datos y colección de datos son ejecutados en un sistema. Pueden soportar las tecnologías de generación I y generación II. Además los sistemas honeypot también pueden estar localizados en el mismo sistema físico. Esto tiene una gran ventaja en los recursos que se tienen dedicados para la implementación de una tecnología honeynet ya que no siempre las organizaciones pueden destinar los recursos necesarios para la investigación de amenazas.

Existen varios mecanismos que hacen realidad este tipo de honeynet, VMWARE es un software que permite a varios sistemas operativos ejecutarse al mismo tiempo. User Mode Linux es otro software que permite la ejecución de varios sistemas Linux simultáneamente.

Riesgos.

Las honeynets no son una solución que se active y de deje, sino más bien, son un complejo tipo de honeypot que requiere un mantenimiento, administración y vigilancia constante. Pare maximizar su efectividad, será necesario detectar y reaccionar al incidente lo más rápido posible.

Observando las actividades de los intrusos en tiempo real, se podrá tener una máxima habilidad de captura y análisis de datos.

Para detectar actividad desconocida, será necesaria una constante revisión de la actividad sospechosa. Esto requiere demasiado tiempo y habilidad de análisis.

Por ejemplo, en solo 30 minutos un intruso hace suficiente daño a un honeypot comprometido, en contraste se requiere de aproximadamente entre 30 y 40 horas para comprender completamente lo que pasó. También es requerido un mantenimiento constante para asegurar la operación de la honeynet. Si alguna cosa esta mal esto puede causar una falla dentro de la honeynet, algunas de las fallas serían: que el proceso de alerta falle, que los discos lleguen a su máxima capacidad, que las firmas IDS no estén actualizadas, que la configuración de los archivos sean incorrecta; además se necesitan algunos cuidados adicionales como son: que los sistemas de bitácoras sean revisados constantemente, que el firewall sea actualizado y corregido. Esto representa solo algunos de los constantes cuidados y mantenimientos que se necesitan para una honeynet exitosa. El trabajo apenas comienza cuando se implemente una tecnología honeynet.

También hay riesgos involucrados en la construcción e implementación de una honeynet. Los intrusos están atacando y comprometiendo sistemas todo el tiempo. Al levantar una red para ser comprometida se expone a todo el mundo a este riesgo. Se debe asumir la responsabilidad de asegurar que la honeynet, una vez comprometida, no pueda ser utilizada para atacar o comprometer a otros sistemas. Sin embargo, con un ambiente como este, siempre es potencial que algo se encuentre mal. Se tiene que implementar una variedad de medidas para reducir este riesgo. Sin embargo, es posible para un intruso, el desarrollar un mecanismo o herramienta que le permita pasar por encima de los métodos de control de acceso, así que la recomendación es nunca subestimar el poder de la creatividad de los intrusos.

Las honeynets son un tipo de honeypot diseñado para obtener información, específicamente de herramientas, tácticas y motivos de los intrusos. Esta información será utilizada para las organizaciones para protegerse en contra de varias amenazas. Se tienen dos diferencias en diseño entre un honeypot y una honeynet. La primera diferencia es que una honeynet no es un simple sistema, es una red de múltiples sistemas y aplicaciones. La segunda diferencia es que las honeynets cuentan con sistema de

producción, los mismos sistemas que se encuentran en la Internet hoy en día, ni los sistemas ni las aplicaciones son emuladas.

Esta combinación hace de las honeynets una buena herramienta para aprender, específicamente es un honeypot diseñado para la investigación. Sin embargo, las honeynets requieren una tremenda cantidad de trabajo.

El administrador de la honeynet es el responsable de asegurar que otros sistemas no sean atacados de la honeynet comprometida. Sin una apropiada administración, los riesgos del uso tal vez se presenten.

Esta herramienta no es la panacea de la seguridad en cómputo, y tal vez no sea una solución conveniente para todas las organizaciones. Para esto primero se recomienda a las organizaciones enfocarse en la seguridad de la organización, corrigiendo adecuadamente los sistemas o deshabilitando servicios no necesarios. Una vez aseguradas las organizaciones, tal vez sean capaces de usar honeynets como una poderosa herramienta para tomar la iniciativa y aprender más acerca del enemigo y de las organizaciones.

3.5 Cómo realizar el análisis forense de datos, recolección y preservación de evidencia.

3.5.1 Captura de la evidencia

Al realizar un análisis forense a un sistema Linux es necesario capturar toda evidencia que nos ayude a lograr nuestros objetivos. Obtener todo dato que nos indique quién fue, por qué, cómo se realizó y en qué momento. Tanto software como hardware pueden arrojar muchas pruebas de lo acontecido. Por lo que en ocasiones puede servir contar con una cámara fotográfica para tomar instantáneas del entorno en el cual se encuentra el sistema afectado, esto con el fin de obtener la mayor cantidad de información posible si existió algún incidente vía hardware como la pérdida de respaldos o el cambio de éstos. También la alteración del cableado estructurado de la institución y del cuarto de servidores.

Respecto al software, una de las primeras cosas que se debe de capturar es toda evidencia volátil, como lo es:

- Memoria del sistema
- Servicios
- Procesos
- Puertos abiertos
- Conexiones establecidas
- Usuarios logueados en ese momento
- Tráfico de red.

3.5.1.1 Memoria del sistema.

Dentro de los sistemas operativos Linux existe la memoria física y la memoria virtual llamada swap. La memoria swap es un espacio en el disco duro para poder usarse como una extensión de memoria virtual en el sistema, es decir, es una técnica utilizada para hacer creer a los programas que existe más memoria RAM de la que en realidad existe. Si algún programa necesita mayor cantidad de memoria que la que se tiene de RAM, el propio sistema operativo se encarga de pasar datos a la swap cuando se necesita más espacio.

Por lo tanto, la memoria total del sistema dentro de Linux es la suma de la memoria RAM instalada y la swap disponible. Toda esta memoria del sistema, al momento de apagar o reiniciar el sistema es volcada al disco duro. Si se quita el cable de corriente del sistema todo lo que se tenía en memoria se pierde ocasionando en muchas ocasiones archivos corrompidos.

Si es posible, se debe de capturar toda la memoria del sistema antes de apagar o reiniciar el equipo. Esto se hace de una forma fácil dentro de un sistema Linux porque sigue la filosofía de un Unix; todo lo ve como un archivo. Así es que, accediendo a un archivo podemos acceder a la memoria física del sistema y también accediendo al sistema de archivos, en este caso swap, podemos capturar todo lo que se tenía en memoria virtual. Para la memoria física accedemos al directorio dev y encontramos los archivos mem y kmem. Para la memoria virtual dependerá de la partición dentro del disco duro que se le haya asignado en cada sistema. Tanto la localización de la memoria como el respaldo pueden ser fáciles, pero su análisis es más complejo.

3.5.1.2 Servicios.

Los servicios son programas o aplicaciones cargadas por el propio sistema operativo. Estas aplicaciones tienen la particularidad que se encuentran corriendo en segundo plano (Background). Por defecto, con la instalación se instalan y ejecutan una cierta cantidad de servicios. Dependiendo de las necesidades de cada sistema se pueden necesitar todos o no.

Para poder determinar los servicios que ofrece un sistema podemos hacer lo siguiente: verificar los procesos que se están ejecutando, determinar el nivel de ejecución del sistema y obtener los servicios que hay en dicho nivel, determinar las conexiones de red, listar todos los archivos abiertos en el sistema, escanear el sistema operativo, verificar el estado de cada uno de los servicios que puede ofrecer el sistema, analizar el tráfico de red.

Tomar en cuenta que cada una de las opciones indicadas en el párrafo anterior, alteran el estado del sistema, por lo que la decisión que se tome para obtener los servicios en ejecución impactará directamente en el rumbo del análisis forense. Para este caso recomendamos cargar binarios estáticos del CD forense o floppy y comenzar a determinar los procesos en ejecución y determinar las conexiones de red.

3.5.1.3 Procesos.

Un proceso es una instancia de un programa en ejecución y también la unidad básica de planificación en Linux[38]. Lo consideraremos como un programa en ejecución. Todos los procesos se encuentran en memoria ya sea física o virtual, por lo que al respaldar la memoria capturamos los procesos en ejecución. Pero, como anteriormente se había dicho, esto es más complicado de analizar, por lo que si es posible determinar los procesos en ejecución del sistema por medio de comandos es mejor. Para este caso, necesitamos el CD forense o floppy y ejecutar los comandos necesarios para obtener esta información. En esta información podemos encontrar si el intruso dejó un programa ejecutándose en segundo plano y qué servicios se ofrecen. También podemos determinar

[38] http://www.linux.cu/manual/basico-html/node101.html

la cantidad de memoria y de CPU que están requiriendo cada uno de los procesos, y además podemos ver el nombre de los programas en ejecución.

3.5.1.4 Puertos abiertos.

Un canal por donde entra y sale información en cualquier computadora conectada a internet, los datos van de un puerto a otro, de una a otra maquina. Hay un total de 65536 puertos en cada maquina[39]. En principio, los puertos 1024 hacia abajo están reservados a las rutinas del sistema operativo, o mejor dicho a los programas estándares del TCP/IP.

Un puerto puede estar*:*

- **Abierto:** Acepta conexiones. Hay una aplicación escuchando en este puerto. Esto no quiere decir que se tenga acceso a la aplicación, sólo que hay posibilidad de conectarse.
- **Cerrado:** Se rechaza la conexión. Probablemente no hay aplicación escuchando en este puerto, o no se permite el acceso por alguna razón. Este es el comportamiento normal del sistema operativo.
- **Bloqueado o Sigiloso:** No hay respuesta. Este es el estado ideal para un cliente en Internet, de esta forma ni siquiera se sabe si el computador está conectado. Normalmente este comportamiento se debe a un firewall de algún tipo, o a que la computadora está apagada.

Los puertos son abiertos a petición de las aplicaciones o programas que van a utilizarlos. Un programa que precise comunicarse con una aplicación que está funcionando en otra máquina, utilizará un puerto determinado, que tiene reservado para su propio uso, por medio del cual realizará una petición a otro puerto situado en la máquina remota.

Al capturar la evidencia en el sistema debemos determinar los puertos abiertos que tiene. Conocer cuales de ellos aceptan conexiones del exterior o son únicamente de conexión interna de la subred, indagar si existe un firewall y conocer las reglas del mismo para saber que puertos protege y cuales no. Al analizar los puertos abiertos nos ayuda a

[39] http://www.lcu.com.ar/faq.php

encontrar puertas traseras o servicios que el administrador no había configurado en el sistema; aunque esto no es del todo indicativo de que se tiene un intruso en el sistema porque se puede dar el caso de que el administrador sea nuevo en la materia y no tenga idea de lo que hace o tiene su sistema. Con comandos básicos como netstat, nmap o con el envío de paquetes de forma remota podemos verificar los puertos abiertos en el sistema. Estos comandos se deben de ejecutar desde el CD forense o desde floppy, no utilizar los comandos del propio sistema.

3.5.1.5 Conexiones establecidas

Conocer las conexiones establecidas en el sistema en ese preciso momento, para conocer qué usuarios intentan loguearse o conectarse a algún puerto en específico. Las conexiones establecidas en ese momento nos pueden ayudar a encontrar puertas traseras, conexiones del otro lado del mundo que nada tienen que hacer hasta este sistema. En muchos casos, es posible encontrar que el intruso transfiere información del sistema a otro sistema o intenta atacar a otros sistemas desde la máquina perjudicada. Con esto nos podemos ayudar a tomar decisiones críticas en el transcurso del análisis forense, como puede ser el tener que quitar el cable de red o el cable de corriente ocasionando una gran pérdida de evidencia. Con la utilización de comandos como netstat, w, ps, lsof, finger podemos verificar las conexiones establecidas.

3.5.1.6 Cuentas de usuarios y grupos

Al momento de iniciar la captura de la evidencia nos podemos encontrar que el sistema en producción puede tener usuarios logueados dentro del sistema. Este registro de usuarios puede ser de gran ayuda porque en muchas ocasiones el ataque puede ser de la red interna y de una persona que conoce sistema y tiene forma de entrar al mismo con facilidad. El intruso puede regresar en cualquier momento y es importante tener el registro de las personas que se han logueado tanto anteriormente como en ese preciso momento. Tomar en cuenta que ciertos caballos de troya pueden borrar los registros del sistema ocasionando que no detectemos los logueos del intruso y sólo veamos logueos de los demás usuarios del sistema. También nos puede ayudar para tomar decisiones oportunas como es analizar la red y detectar la fuente real del ataque, desconectar el cable de red por si el intruso ocasiona daños al sistema o a otros sistemas o también quitar el cable de corriente para evitar daños irreversibles al sistema.

3.5.1.7 Tráfico de red

Cuando detectemos usuarios logueados en el sistema, lentitud tanto en la red como en las actividades rutinarias de los sistemas, así como demasiados puertos abiertos y con conexiones establecidas es recomendable utilizar un analizador de tráfico de red. En muchas ocasiones es sólo un falso positivo debido a una mala configuración de un sistema y éste se encuentre inundando de paquetes la red haciendo peticiones sin sentido. También es posible encontrar gusanos que intentan propagarse dentro de la red e intentan realizar conexiones a máquinas Linux. Colocando un sniffer o un IDS en la red es suficiente determinar ciertos comportamientos extraños dentro de la red. Si la institución ya cuenta con éstos, entonces se tiene que acudir a ellos para capturar y analizar toda la información arrojada por ellos. No se puede confiar del todo en ellos porque es posible que estos se encuentren comprometidos también, pero si puede ser un punto de apoyo para encontrar cabos sueltos en la investigación. Lo recomendable es utilizar nuestras propias herramientas para analizar la red interna. Para ellos podemos utilizar un tcpdump, ethereal o snort.

También se debe de capturar la evidencia de rootkits, de discos duros completos y para llevar a cabo esto es necesario la utilización de herramientas de duplicación de discos.

3.5.1.8 Discos

Toda actividad que realicemos dentro del disco duro original afecta la evidencia rotundamente porque nosotros podemos ocasionar pérdida o modificación de la información impidiendo un desarrollo favorable para el caso. Por tanto, es recomendable manipular una copia de la evidencia para no perjudicar al sistema original. Es cierto que, tanto procesos, como conexiones de red y utilización de los recursos del sistema en tiempo real es difícil duplicarlos y hacer una copia completa del sistema, por lo que no queda de otra que capturar esas partes dentro de archivos siendo estos el respaldo de esta información.

3.5.1.9 Herramientas de duplicación de discos

Para capturar toda la información sin perjudicar al sistema original es importante contar con herramientas de duplicación de discos. Estas herramientas deben de garantizar una copia fiel del sistema, es decir, que no sólo los archivos del sistema deben de copiarse íntegros, si no que también archivos borrados del sistema, memoria asignada y no asignada, ínodos perdidos, disco duro no ocupado, etc. Todo el disco duro debe ser duplicado. Para esto, la herramienta de duplicación debe manejar bit a bit la duplicación de la información para garantizar que ninguna evidencia se pierda al momento de realizar la duplicación del disco duro.

Una herramienta que realiza esta actividad y se amolda a nuestras necesidades de analistas forenses es la herramienta estándar de Unix “dd” la cual realiza una copia bit a bit del archivo que le indiquemos hacia otro dispositivo o archivo seleccionado. Esto asegura una copia fiel de un disco duro completo sin perder detalle alguno del contenido del mismo, es decir, además de todos los archivos que se encuentran en el sistema también incluye datos perdidos como pudieran ser archivos borrados o truncados que no se guardarían comúnmente con una simple copia realizada por la herramienta *cp*.

3.5.1.10 Análisis de la información de disco

Al momento de recopilar toda la información o en su defecto, de tener el sistema original y tener que aplicar el análisis forense directamente sobre de él es necesario analizar la información de todo el sistema en busca de evidencia que el intruso haya olvidado borrar. Esta búsqueda debe ser minuciosa y detallada. En este punto se requiere mucha paciencia para posiblemente analizar Gb de información en el sistema. Para facilitarnos las cosas, podemos realizar nosotros mismos programas bajo lenguaje C, Perl o Shell Scripts que nos ayuden a realizar esta ardua tarea. Los scripts deben tener la característica de amoldarse a las necesidades de los diferentes Linux que hay actualmente; y si se tiene que hacer alguna modificación sobre de ellos, que sea mínima para no perder tiempo en algo que no debe requerir tiempo excesivo. Recomendamos utilizar scripts en los cuales se realicen búsquedas de archivos especiales, como son los de carácter, de bloque, sockets. También dentro de este script se debe incluir un programa que busque rootkits, archivos comprimidos en lugares donde no deberían de estar. Si se maneja encriptación de archivos, tener bien ubicados todos ellos, para un

análisis posterior de los archivos, si es necesario. Por otra parte, dentro de la fase de análisis de la información, se debe analizar la memoria del disco afectado. Por tanto, los puntos a considerar en este apartado son los siguientes:

3.5.1.11 Archivos especiales

Una de las tareas del analista forense es buscar todos los archivos especiales que existan dentro del sistema. En condiciones normales, todos los archivos especiales se encuentran dentro del directorio /dev, los cuales hacen referencia tanto a particiones de disco duro, unidades de cinta para hacer respaldos, unidades de CD-ROM, la unidad de floppy, si hay impresoras, etc. En general, cualquier dispositivo con el cual cuente el sistema afectado. Si se encuentra un archivo especial en otra parte del sistema que no sea el directorio /dev pasa algo extraño. De igual forma, los sockets normalmente están alojados en el directorio /tmp. Si existen sockets en otra parte es un buen indicio que la máquina sufrió un incidente. La búsqueda debe ser exhaustiva para encontrar este tipo de archivos dentro del sistema.

3.5.1.12 Rootkits

Antes de comenzar a localizar y analizar el comportamiento de un rootkit vamos a definirlo. Un rootkit es una herramienta utilizada por hackers la cual consiste en un conjunto de programas que ayudan al intruso a ocultar su presencia dentro de un sistema comprometido[40]. Un rootkit facilita al intruso el control de la máquina. Los rootkits fueron descubiertos a mediados de los '90. En aquella época, los administradores de sistemas Unix de Sun comenzaron a ver un comportamiento extraño en el servidor, la falta de espacio de disco, ciclos extra en la CPU y las conexiones de red que no se mostraba con el comando netstat. El nombre Rootkit se origina a partir de la idea de que quien lo utiliza puede acceder fácilmente al nivel de root, o de administrador del sistema, una vez que la herramienta ha sido instalada.

Las herramientas sustituyen programas propios del sistema con troyanos que ocultan los directorios creados por los hackers y también ocultan los procesos y programas ejecutados por ellos. Este tipo de herramientas ya vienen empaquetadas y listas para poder ejecutarse y dejar que haga las cosas por nosotros, por lo que para un

[40] http://www.nicatech.com.ni/r.htm

administrador es complicado detectar un rootkit a primera vista porque los rootkits ya están preparados para la tarea de ocultación de información y así, un intruso no se tiene que ver en la necesidad de hacer todo paso a paso y además, un rootkit difícilmente se equivoca y un intruso, si lo hace todo, sin ocupar un rootkit puede cometer muchos errores ocasionando que de inmediato un administrador se de cuenta de su intrusión.

De acuerdo a la tecnología empleada, existen tres clases principales de Rootkits disponibles hoy en día:

1. **Kits binarios**: alcanzan su meta substituyendo ciertos archivos del sistema por sus contrapartes Troyanizadas.
2. **Kits del núcleo:** utilizan los componentes del núcleo (también llamados módulos) que son reemplazados por troyanos.
3. **Kits de librerías:** emplean librerías del sistema para contener Troyanos.

El principio operativo de los rootkits es el de reemplazar archivos de programa del sistema con versiones modificadas, para que se ejecuten determinadas operaciones. A estas versiones modificadas se les conoce con el nombre de troyanos. Un rootkit es, en esencia, una colección de programas troyanos.

El objetivo de los troyanos es imitar exactamente el comportamiento de las aplicaciones originales, pero escondiendo los archivos, acciones y evidencias del intruso. En otras palabras, una vez instalado el rootkit, en principio, el intruso podrá utilizar el sistema sin ser detectado por el administrador.

Los archivos que suelen modificar los rootkits son:

- login, su, telnet, netstat, ifconfig, ls, find, du, df, libc, sync,
- passwd, ps, nmap, finger, last, w, who, cat, more, etc.

3.5.1.13 Archivos comprimidos

Para buscar archivos comprimidos es necesario determinar el fin del servidor y conocer donde se puede encontrar toda la información de los usuarios. Esto nos ayuda para personalizar las búsquedas en el sistema, es decir, buscar primero archivos comprimidos en lugares donde no debe de haber, y posteriormente, buscar archivos

comprimidos en los directorios personales de los usuarios. También cabe la posibilidad, de que si es un servidor de ftp, haya archivos comprimidos en el directorio raíz del ftp, por tanto todos estos archivos deben ser analizados minuciosamente.

Antes de tomar alguna acción como el análisis del contenido de un archivo comprimido de un usuario en específico, es vital tener anteriormente cierta evidencia que nos orille a comenzar a analizar el contenido de los archivos comprimidos. Pero, en el caso en que no se cuente con la evidencia necesaria es difícil iniciar esta tarea porque estaríamos buscando algo sin rumbo fijo.

3.5.1.14 Archivos cifrados

Buscar archivos cifrados y analizarlos es una tarea complicada porque no se cuenta con el password para descifrar los archivos y por tanto, no podemos acceder a esta información. Posiblemente, el intruso haya dejado troyanos, virus, gusanos, o cualquier archivo malicioso que en futuros accesos al sistema pueda llegar a utilizarlos para atacar otros sistemas. Lo único que nos queda es encontrarlos y determinar la procedencia de los mismos. Si se da el caso en el que los usuarios manejen encriptación, entonces apoyarnos en el análisis de su historial de comandos para determinar qué archivos fueron cifrados.

3.5.1.15 Memoria en disco

La memoria forma parte crucial dentro del análisis forense porque dentro de ella podemos encontrar las actividades realizadas por el sistema en producción, como es la apertura de archivos, ejecución de programas que están ocupando recursos, accesos a memoria, información de usuarios conectados o que se encuentran ocupando un recurso, si un usuario está ejecutando un programa que a simple vista con comandos básicos de Unix no los podemos encontrar, sí lo podemos hacer en memoria. Dentro de esta parte se va a tener muchísima información, pero así como puede haber información valiosa para nosotros únicamente puede haber información irrelevante que no sirve para nosotros. El análisis de la memoria en disco es la tarea más difícil en la parte de análisis de la información porque se va a tener información excesiva de la cual únicamente unos fragmentos de ella nos puede servir. A pesar de que ya existen herramientas que nos

ayudan a depurar la información contenida en memoria aún es insuficiente con las necesidades que tiene un analista forense.

3.6 Análisis de la imagen capturada

3.6.1 Clasificación de archivos por fechas

El sistema operativo Linux guarda los tiempos de acceso, modificación y creación de un archivo cualquiera dentro del sistema. Estos registros que se tienen sobre todo archivo es imprescindible para un analista forense porque con esos tiempos, llamados comúnmente tiempos MAC, podemos determinar el momento en que se llevó a acabo una intrusión, la instalación de rootkits, la alteración de archivos de arranque, creación de nuevos archivos o descargados de otro servidor que ya haya sido comprometido, etc. Con los tiempos MAC podemos encontrar muchísima información que nos guía a determinar la fecha exacta en que fue comprometido el sistema y posiblemente también los momentos en que el intruso regresa al sistema para seguir apoderándose de él y de otros equipos utilizando ahora como puente nuestro sistema afectado. La búsqueda debe ser minuciosa y con extremada paciencia para no perder detalle de todo lo que aconteció dentro del sistema. Con comandos básicos de Linux encontramos los tiempos MAC por lo que lo único necesario para este caso es contar con el CD de herramientas forense para aplicar este punto. El comando utilizado dentro de Linux para llevar a cabo esto es el comando *find* con diferentes parámetros dependiendo de lo que se desea buscar, si es por modificación, creación o únicamente por acceso a cierta información. Este comando nos ofrece la ventaja de que analiza todo el sistema y además puede realizar búsquedas en base a una fecha determinada, es decir, si deseamos encontrar los archivos modificados desde hace cinco días o si preferimos los archivos que se accedieron desde hace más de tres días y a su vez los que se accedieron hace menos de diez días, etc.

3.6.2 Borrado de archivos

Cuando se borra un archivo dentro de Linux lo que se hace es quitarle la liga que apunta al espacio ocupado dentro del disco duro, es decir, los datos contenidos son perdidos por completo hasta que ese espacio sea ocupado por otro archivo. Mientras no sea ocupada por otro archivo la información seguirá ahí aunque nuestro sistema operativo marque que ya no existe. Esta característica de los sistemas operativos es de gran ayuda

para poder aplicar el análisis forense, ya que así nosotros podemos determinar los archivos borrados e intentar la recuperación de éstos para determinar qué función cumplían dentro del sistema y cuál fue el motivo de que el intruso haya decidido eliminarlo. El inconveniente radica en que si el sistema en producción tenía una alta interacción con los usuarios y se tenía mucha transferencia de información del disco duro, esta información eliminada puede ser ocupada rápidamente por otros archivos ocasionando que ya no haya evidencia. Asimismo, el intruso también puede aplicar mecanismos para que los archivos eliminados se pierdan por completo llenando el espacio libre del disco con ceros, es decir, limpia el espacio libre sin dejar huella alguna.

3.6.3 Espacio libre y espacio de relleno

Cuando deseamos obtener una copia del disco de la máquina perjudicada, es necesario, antes de crear las imágenes, contar con el espacio suficiente para poder realizar las copias, es decir debe existir espacio libre suficiente que no afecte la copia que se va a realizar. Esto se requiere debido a que no se debe permitir que se mezclen cosas que supuestamente ya se borraron pero que aún así, al momento de realizar el análisis forense, se va a detectar archivos borrados y el contenido de éstos, ocasionando que haya una incongruencia al momento de analizar los archivos borrados. Para dejar limpio por completo el espacio libre que se va a utilizar, se recomienda llenar este espacio con ceros, para que todo lo que se haga como imagen sea realmente lo que pertenecía al disco original afectado. Recordar que al momento de hacer la imagen de las particiones del disco duro perjudicado se realiza bit a bit, es decir, todo lo que tenga la partición, será volcado a la imagen.

3.6.4 Ocultación de archivos

Un intruso lo que desea es pasar desapercibido dentro del sistema al cual comprometió, por lo que, comúnmente lo que hacen es crear archivos y directorios ocultos en el sistema y en lugares en los cuales no se tiene mucho acceso continuo, como es en el directorio /dev/ o también en /usr. En esos lugares los intrusos se instalan normalmente, por lo que nuestra tarea es, determinar y localizar el directorio o directorios en los cuales se ha instalado el intruso y encontrar todos los archivos que ha colocado en el sistema para completar su control. A los archivos ocultos se les asocia con un punto al inicio de su nombre; con base en esto, un intruso lo que puede hacer es crear un

directorio llamado ".. ", (punto punto espacio) , también directorios "..." (tres puntos), o directorios " " (espacio), etc. En general directorios que no se detectan a simple vista y que pasan desapercibidos.

3.6.5 Búsqueda de programas maliciosos

El intruso al instalarse dentro del sistema, trae consigo su propio kit de herramientas, las cuales pueden incluir sniffers, crackeadores de passwords, programas de irc, caballos de troya, exploits, gusanos, etc. Todos estos programas pueden cumplir el fin de atacar a otro sistema, de garantizar su permanencia dentro del sistema, o comenzar a capturar información de la red a la cual se ha accedido.

Aquí radica la importancia de conocer nombres de este tipo de herramientas para así, poder buscarlas dentro del sistema y localizarlas de inmediato. También nos ayuda realizar una búsqueda en base a cierto tipo de archivos, es decir, comenzar a buscar en base a archivos binarios, archivos con permiso de ejecución, sockets, archivos en un cierto lenguaje de programación como pueden ser perl, lenguaje C o C++, java, shell scripts.

3.7 Análisis de sistemas cliente.

Mucho más abundantes que las intrusiones en servidores son las intrusiones en sistemas cliente. En muchos casos, estas intrusiones se realizan teniendo como objetivo el acceso a servidores a los que el usuario víctima tiene derecho. Esto es debido a que normalmente los puestos de los usuarios son el eslabón más débil de la cadena de seguridad de un sistema.

Por otra parte, este punto de la investigación forense tiene especial importancia, ya que recordando las amenazas humanas, el 70 % de los incidentes de seguridad los realizan los "insider" (personal que labora dentro de la misma organización), por lo que no es de extrañarse que se lleguen a encontrar responsables dentro de la organización misma; para el analista forense ellos son los principales sospechosos, es por eso la relevancia del análisis forense de sistemas clientes de "insiders" dudosos que utilizan los recursos del servidor comprometido.

3.7.1 Interacción con Internet

Casi cualquier servidor en red se encuentra conectado a la red mundial Internet, es por esta red por la que se pueden perpetrar ataques y por lo tanto es importante identificar toda interacción vía Web, también el envió y recibo de correo electrónico, los Chats, las descargas de archivos, ya sea por ftp o http, la detección de sistemas vivos en la red por medio del envió de paquetes aleatorios buscando algún sistema para comprometerlo, entre otros.

3.7.1.1 Detectores de intrusos en sistemas clientes

No solo los sistemas detectores de intrusos se deben instalar no sólo para analizar los paquetes de los servidores, sino también en los sistemas clientes, ya que en el IDS del cliente se guardan los estados de alarma por patrones anómalos, conductas sospechosas del cliente, y se tiene un historial de los acontecimientos más completa y veraz de toda la red interna de la institución. Podemos detectar la entrada y salida de paquetes de los sistemas clientes y posibles ataques a ellos utilizándolos como puente para futuros ataques a servidores.

3.7.1.2 Navegación Web

Dentro de la navegación en la Web, se tiene que examinar los historiales y archivos temporales de los navegadores más populares sobre Linux, como lo son mozilla, netscape, galeon, Konqueror, entre otros. Así como también el servicio de web (Apache por ejemplo) verificando el logs de los acontecimientos, para cada caso, se tienen que verificar si las páginas visitadas son de índole peligroso, así como también los paquetes y programas que se hallan descargado.

3.7.2 Correo electrónico

El correo electrónico que ofrece el servidor debe ser examinado por los analistas forenses siguiendo un código de ética profesional y respetando las políticas de seguridad de la organización. Solo se debe analizar el correo electrónico ajeno en una situación muy indispensable.

Este análisis se puede realizar sobre el archivo /var/spool/mail/usuarioX, o sobre la cola de correo del sistema. Este archivo contiene de manera cronológica los correos que recibió el usuarioX, otro archivo de interés es el $HOME/mbox, que cumple la misma función que /var/spool/mail/usuariox, para un sistema Linux. El sistema registra los correos que envió el usuario en el archivo $HOME/mail/sent-mail, además de que también guarda un historial mensual en otro archivo por ejemplo: $HOME/mail /sent-mail-feb-2004. Así mismo, almacena el correo que el usuario recibió y guardó en $HOME/mail/saved-messages. Con estos archivos se puede revisar el correo del sospechoso aún cuando él ya los haya borrado de su mailbox.

3.7.3 Análisis de documentos

Al Igual que en los correos electrónicos se debe de tener una ética profesional elevada y apegarse a las políticas de seguridad de la Organización para poder revisar el Home-directory de los sospechosos, se tendrán que analizar sus archivos personales, manuales que se tengan, ya que muchos de ellos enseñan a los usuarios a atacar un servidor, notas sospechosas de los usuarios, archivos ocultos, con la finalidad de encontrar alguna evidencia de ataque por ese usuario.

3.7.4 Análisis de programas sospechosos

En este punto se debe poner especial atención en los archivos binarios, Shell Scripts, archivos en lenguaje C, Java, PERL, archivos ocultos, paquetes y aplicaciones descargados de Internet, con el fin de buscar programas que puedan hacer daño o estén monitoreando de manera indebida al servidor analizado, como archivos troyanos, sniffers, exploits, virus, entre otros.

3.8 Estructura de los archivos binarios

Revisar en cada sistema la integridad de los archivos binarios, con el fin de detectar si hay archivos troyanizados. Los programas troyanos buscan cambiar los archivos binarios por los propios troyanizados, con el fin de obtener el control del sistema y además, pasar desapercibidos. El análisis de archivos binarios incluye la verificación de los tiempos MAC, el chequeo de firmas MD5, si es que existen, y también el análisis del contenido del archivo binario. Por otra parte, se debe de buscar dentro del árbol de archivos del sistema todos los archivos binarios, ya que no sólo pueden haberse

sustituido algunos binarios, sino que también el intruso pudo haber esparcido por todo el sistema archivos binarios que le ayuden a sus tareas de control. Al localizar todos los binarios del sistema, comenzar a dar prioridades acerca de cual va a ser el primero en analizarse. Recomendamos que se analicen los archivos binarios que no se encuentren dentro de los directorios que contienen a los binarios del sistema, es decir, archivos binarios que no deben de estar en un directorio que no les corresponde.

3.8.1 Análisis en ejecución

Si se detectan archivos binarios dentro del sistema y que estén en un lugar en el cual no les corresponde dentro del sistema de archivos, y además, estos se están ejecutando, hay que actuar de inmediato. Hay varias posibilidades, detectar los procesos hijos de este binario en ejecución, ver qué recursos del sistema está consumiendo, determinar si utiliza sockets para la conexión en red, conocer qué puertos ha abierto, determinar hacia dónde se conecta, o simplemente, determinar qué fin busca dentro del sistema, porque es posible que únicamente esté detectando conexiones del administrador, que para él pueden ser sospechosas y si corre riesgo, este programa binario comenzará a borrar toda evidencia dentro del sistema. Al momento en que detectemos los programas en ejecución, se debe obtener como información adicional qué usuario del sistema se encuentra ejecutando dicho programa, o en el peor de los casos, si es el propio administrador el que lo está haciendo; esta última situación es indicativo de que muy posiblemente el sistema ya haya sido comprometido. El análisis en ejecución también incluye la revisión de la memoria ya que el programa se encuentra cargado en ella.

3.8.2 Entorno seguro de pruebas

Al realizar búsquedas de archivos binarios, si se llegan a detectar programas que no se sabe muy bien cual es su fin, se deben analizar ejecutándolos nosotros mismos para conocer en plenitud su comportamiento dentro de un sistema en producción. La ejecución de este tipo de programas se debe hacer bajo un medio completamente controlado, porque desconocemos el impacto que pueda causar dentro de la red interna de la institución o dentro del laboratorio forense. Recomendamos que estos programas se ejecuten dentro del laboratorio de los analistas forenses y no dentro de la propia institución, porque así es posible controlar por completo nuestro entorno, en cambio, si se

realiza dentro de la institución no vamos a poder manejar todas las variables que estén en juego dentro de la red. El medio controlado incluye un firewall y una red interna, con un límite de ancho de banda de red, un límite de conexiones establecidas hacia fuera de nuestra red interna, un mínimo de equipos conectados dentro de la red interna, un máximo de restricción dentro del equipo que se utilice para la ejecución de estos programas. La restricción dentro del equipo puede ser la misma a la de un Honeypot. Por lo que nosotros concluimos que el equipo utilizado para este fin debe ser un honeypot dedicado, en el cual ya se tiene configurado todo el mecanismo de seguridad, y únicamente colocamos nuestro programa y no tenemos que improvisar un equipo para cada vez que se requiera. El no contar con un honeypot puede implicar pérdida de tiempo y muy posiblemente que lleguemos a comprometer nuestro laboratorio de analistas forenses por no contar con los mecanismos de seguridad necesarios para este tipo de pruebas.

3.8.3 Interacción con el sistema

Verificar y analizar la forma en cómo se comportan todos los binarios al interactuar directamente con el sistema. Algunos de ellos, puede que estén troyanizados y su efecto sea claro en el sistema: ocultar evidencia. Pero, aparte de estos binarios, puede haber otros que al ejecutarlos muestren resultados inesperados para el administrador y en este caso para nosotros, porque éstos pueden mostrarnos información incongruente con la obtenida de otros comandos o lo registrado en bitácoras. Al encontrarnos con estos binarios en ningún momento debemos de fiarnos de ellos porque no tenemos la certeza de que no hayan sido modificados o suplantados dentro del sistema. Lo mejor es buscar los tiempos MAC de este tipo de archivos binarios en el sistema, y determinar si fueron alterados y comenzar a concluir cuáles fueron todos los archivos binarios modificados y conocer el troyano o rootkit que ejecutó el intruso en el sistema.

3.9 Teoría de la evasión forense

El término de evasión forense (anti-forensics) se refiere a las técnicas de eliminación y/o de ocultación de pruebas para complicar o imposibilitar la efectividad del análisis forense. Es un intento de mitigar la cantidad y calidad de información que un investigador puede encontrar. Cada uno de los pasos del análisis puede ser explotado y subvertido.

Los mecanismos principales para alcanzar este objetivo son: técnicas de destrucción y ocultación de datos.

El estudio forense es extremadamente vulnerable a subversión cuando la información, en su estado básico (raw data image), se convierte en pruebas (por ejemplo emails). Cada paso de la conversión es vulnerable por la complejidad y procesos abstractos que se realizan sobre los datos. Cuando se trabaja con los datos cualquier despiste puede resultar en pérdida de detalles y los detalles, en realidad, son las partes más importantes del rompecabezas. Cuando los detalles desaparecen se crean vacíos importantes que pueden ser explotados. El despiste no es la única fuente de errores, sino también los fallos en las herramientas forenses frecuentemente utilizadas. Bugs en las implementaciones de estas herramientas proporcionan mejores oportunidades de explotación para agentes evasivos.

Pocas cosas pueden ser hechas de forma remota por un agente evasivo a fin de prevenir que el sistema de archivos se salve. Se puede frenar la captura de información a través de destrucción o ocultación de la misma. De estos dos mecanismos la destrucción de datos es la más fiable ya que no deja ninguna pista. Esta técnica proporciona medios para la eliminación segura de huellas y pruebas existentes para cubrirse las huellas del atacante de forma más efectiva. La ocultación de información sólo es efectiva cuando el analista no sabe dónde buscarla, siendo la integridad del medio de almacenamiento imposible de garantizar a largo plazo. Por esa razón, la ocultación de datos debe ser combinada con ataques a las fases de estudio de datos recogidos (por ejemplo formatos propietarios de archivos) y de examen (por ejemplo cifrado). Técnicas de ocultación son útiles en caso de que los datos se tengan que guardar durante un periodo esencial de tiempo.

CAPITULO IV

ENTORNO DE SEGURIDAD Y LOS REQUERIMIENTOS PARA EL ANÁLSIS FORENSE EN LA FACULTAD DE INGENIERÍA

4.1 Análisis de la seguridad en cómputo del Laboratorio de Redes y Seguridad de la Facultad de Ingeniería.

Descripción del objeto a evaluar

Nuestro propósito es proponer y aplicar una metodología forense de manera general en un medio controlado, como lo es el laboratorio de Redes y Seguridad de la Facultad de Ingeniería, analizando los entornos más representativos que existen en la Facultad, con el fin de contar con una metodología formal para evaluar y describir los incidentes de seguridad que puedan surgir.

Dicho laboratorio forma parte del área de Redes de Computadoras del Departamento de Computación. El Departamento de Computación es parte de la División de Ingeniería Eléctrica de la Facultad de Ingeniería de la Universidad Nacional Autónoma de México. Este laboratorio tiene como finalidad realizar tanto prácticas como investigaciones en materia de redes de computadoras y seguridad en cómputo.

Entorno Físico

El laboratorio se encuentra localizado en la planta baja del la División de Ingeniería eléctrica (“Valdez Vallejo”) y se tienen pasos restringidos para los servidores del laboratorio, por medio de un cerrojo con control de acceso electrónico, y cubículos bajo llave. Por lo general las personas que trabajan en el laboratorio son lo suficientemente responsables, se asignan sus propias labores, todos cuentan con sus propias llaves y sus claves de acceso; los usuarios tienen información muy importante para sus fines por lo que no comparten con nadie la información y la configuración de los equipos que manejan, cada uno de ellos cuenta con un horario asignado para trabajar, pero no se dispone con un registro de entradas o salidas y se tiene vigilancia a todas las instalaciones

La instalación eléctrica de los equipos cuenta con un UPS (fuente de potencia ininterrumpida) para los equipos críticos, las instalaciones se encuentran en buen estado ya que el laboratorio es de reciente creación, también se tiene una buena ventilación que regula de manera eficiente la temperatura y el humo en caso de incendios en el laboratorio, a pesar de que no se cuenta con aire acondicionado. No se tienen paredes, puertas y techos impermeables en caso de inundación.

Para casos de incendio se tienen extintores en varios laboratorios y centros de cómputo de edificio “Valdez Vallejo”, dependiendo de las divisiones y de los recursos de cada área, para nuestro laboratorio no se cuenta con un extintor; por lo general el laboratorio se encuentra bien situado, lejos de áreas combustibles o donde se manejen sustancias inflamables, explosivas, etc; pero las paredes, el material, el suelo, los techos y las puertas no son contra incendios.

Varios equipos que se encuentran en el laboratorio, no tienen sus respectivas especificaciones técnicas. El mantenimiento corre por cuenta de quien hace uso del mismo. Algunos aparatos son muy antiguos por lo que llegan a ser incompatibles con los demás, casi ningún equipo es nuevo, pero se tienen en buenas condiciones para trabajar.

Entorno lógico

No se tienen errores en la configuración, diseño, implementación y operación de los sistemas. Se cuenta con seguridad vía software para proteger la información externa en discos de 3.5 pulgadas, cuando la información no es muy grande. No se tienen los suficientes recursos para proteger la información por medio de espejos y copias a gran escala. En la Red Local se encuentran muy pocos usuarios activos, además se no se han reportado fallos por el cableado estructurado, pero no se siguió ningún estándar para realizar el mismo, porque se baso en lo que se tenía en el laboratorio. Debido a los pocos recursos con los que se cuentan, no existe una planta emergente de red, ya que la misma es muy costosa.

Recursos

Para realizar esta investigación nosotros contamos con un Equipo HP con 128 Mb de memoria RAM, 40 Gb de disco duro, una tarjeta de red ethernet broad com, cuenta con un microprocesador Pentium 4 a 2.4 GHz, DC-ROM, Floppy y su IP es xxx.xxx.xxx.xxx, Este equipo tiene una partición de 2.2 Gb en la cual se encuentra instalado Windows XP con sus actualizaciones, Internet Explorer 6.1 con sus actualizaciones, Microsoft Office XP, Power Quest Patition Magic 7.0, Secure Shell SSH 3.0 y como software inseguro tiene MSN Menssenger 6.1 y 4.7 además de Norton 2003 caducado. En su partición de 18 Gb tiene instalado Fedora 1.0.

El siguiente equipo es una computadora “armada” con 128 Mb en memoria RAM, disco duro de 4 Gb, tarjeta Ethernet com, CD-ROM 8x20X, floppy, Microprocesador ADM K6 a 500 MHz y tiene una IP xxx.xxx.xxx.xxx. Este equipo tiene una partición de 2 Gb, en la que se encuentra instalado Windows Advance Server 2000 server Family, en su otra partición de 2 Gb tiene instalado Red Hat 7.0.

Además, también se cuenta con otro equipo LANIX con 32 Mb de RAM. Este equipo tiene un disco duro de 5.7 Gb, el microprocesador es Pentium I a 100 MHz, la tarjeta de red es Ethernet con sistema operativo Linux Red Hat 6.2. Tiene una IP no homologada xxx.xxx.xxx.xxx. Este equipo es armado y dispone con una unidad de CD-ROM y otra unidad de floppy.

El último equipo es una computadota “armada” con disco duro de 8Gb, microprocesador ADM K6 II a 400 MHz, memoria RAM de 64 Mb, dos tarjetas de red 3 COM 509B, sistema operativo Linux Red Hat 9.0, tiene una IP real xxx.xxx.xxx.xxx y no cuenta con CD-ROM. Esta computadora es la que da salida a Internet.

Todos los equipos están en red local conectados a un switch.

Usuarios

Básicamente los usuarios que asisten al laboratorio son tesistas, usuarios que asisten a cursos y alumnos de la Facultad.

4.2 Amenazas

Hackers[41].

Un Hacker es la persona que está en continua búsqueda de la información. Vive para aprender y todo para él es un reto, lucha por la difusión libre de la información, distribución de software sin costo y la globalización de la comunicación.

[41] Cristian F. Borhelio, Tesis “Seguridad Informática: Sus Implicaciones y Implementación”, 2001

Crackers[42].

Los Crackers en realidad son Hackers cuyas intenciones van más allá de la investigación. Es una persona que tiene fines maliciosos o de venganza, quiere demostrar sus habilidades pero de la manera equivocada o simplemente personas que hacen daño sólo por diversión. Los Hackers opinan de ellos que son Hackers mediocres, no demasiado brillantes, que buscan violar un sistema.

Lamers o Script kidders[43].

Son aficionados jactosos que prueban todos los programas que llegan a sus manos. Generalmente son los responsables de soltar virus y bombas lógicas de la red sólo con el fin de molestar y que otros se enteren de que usan tal o cual programa. Son aprendices que presumen de lo que no son aprovechando los conocimientos del Hacker y lo ponen en práctica sin saber.

Newbie[44].

Son los novatos del Hacker. Se introducen a los sistemas de fácil acceso y fracasan en muchos intentos solo con el objetivo de aprender las técnicas que puedan hacer de él un Hacker reconocido.

Wannaber[45].

Son aquellas personas que desean ser Hacker pero consideran que su coeficiente no da para tal fin. A pesar de su actitud positiva difícilmente consiguen avanzar hacia sus propósitos.

Eavesdropping –Packet Sniffing[46].

Esto se realiza con Packet Sniffers los cuales son programas que monitorean el tráfico que circula por la red. Los Sniffers pueden ser colocados tanto en una estación de

[42] Véase la nota 41
[43] Véase la nota 41
[44] Véase la nota 41
[45] Véase la nota 41
[46] Véase la nota 41

trabajo conectada a la red, como a un equipo Router o a un Gateway de Internet, y esto puede ser realizado por un usuario con legítimo acceso, o por un intruso que ha ingresado por otras vías.

Cada máquina conectada a la red verifica la dirección destino de los paquetes IP. Si estas direcciones son iguales asume que el paquete enviado es para ella, caso contrario libera el paquete para que otras placas lo analicen. Un sniffer consiste en colocar a la placa de red en un modo llamado promiscuo, el cual desactiva el filtro de verificación de direcciones y por lo tanto todos los paquetes enviados a la red llegan a esta placa (computadora donde se instaló el sniffer).

Snooping-Dawnloading[47].

Los ataques de esta categoría tiene el mismo objetivo que el sniffing. Obtener la información sin modificarla. Sin embargo los métodos son diferentes. Aquí además de interceptar el tráfico de red, el atacante ingresa a los documentos, mensajes de correo electrónico y otra información guardada realizando en la mayoría de los casos un downloading (copia de documentos) de esta información a su propia computadora, para luego hacer un análisis exhaustivo de ésta.

Spoofing-Looping[48].

Spoofing puede traducirse por hacerse pasar por otro y el objetivo de esta técnica, es justamente actuar en nombre de otros usuarios, usualmente para realzar tareas de snooping o tampering.

Una forma común de Spoofing es conseguir el nombre y el password de un usuario legítimo para, una vez ingresado al sistema, tomar acciones en nombre de él. El intruso usualmente utiliza un sistema para obtener información e ingresar en otro, y luego utiliza este para entrar en otro, y así sucesivamente. Este proceso llamada looping, tiene la finalidad de evaporar la identificación y la ubicación del atacante.

[47] Véase la nota 41
[48] Véase la nota 41

El camino tomado desde el origen hasta el destino puede tener muchas estaciones, que exceden los límites de un país. La investigación de procedencia de un Looping es casi imposible, ya que el investigador debe contar con la colaboración de cada administrador de cada red utilizada en la ruta.

El envío de falsos e-mails es otra forma de spoofing que las redes permiten. Aquí el atacante envía e-mails a nombre de otra persona con cualquier motivo u objeto.

Muchos atacantes de este tipo comienzan con ingeniería social, y los usuarios por falta de cultura, facilitan a extraños sus identificaciones dentro del sistema usualmente a través de una simple llamada telefónica.

Spoofing[49].

Este tipo de ataques sobre protocolos suelen ser un buen conocimiento del protocolo del que se va a basar el atacante. Los ataques de tipo spoofing son IP spoofing, el DNS spoofing y el web spoofing.

IP spoofing[50].

El atacante genera paquetes de Internet con una dirección de red falsa en el campo from, pero que es aceptada por el destinatario del paquete. Su utilización más común es enviar los paquetes con la dirección de un tercero, de forma que la víctima ve un paquete proveniente de esa tercera red y no la dirección real del intruso. El objetivo es que si la víctima descubre el paquete, ve otra IP como su atacante y no la del verdadero atacante.

DNS Spoofing[51].

Este ataque se consigue mediante la manipulación de paquete UDP pudiéndose comprometer el servidor de nombre de dominios (Domain Name Server). Si se permite el método de recursión en la resolución de nombre y/o dirección IP en el DNS, es posible controlar algunos aspectos en el DNS remoto. La recursión consiste en la capacidad de un

[49] Véase la nota 41
[50] Véase la nota 41
[51] Véase la nota 41

servidor de nombres para resolver una petición de dirección IP a partir de un nombre que no figura en su base de datos. Este es el método de funcionamiento por defecto.

Web Spoofing[52].

El atacante crea un sitio web completo (falso) similar al que la víctima desea entrar. Los accesos a este sitio están dirigidos por el atacante, permitiéndole monitorear todas las acciones de la víctima, desde sus datos hasta los passwords, numero de tarjetas de crédito, etc. El atacante también es libre de modificar cualquier dato que este transmitiendo entre el servidor original y la víctima o viceversa.

IP Splicing Hijacking[53].

Se produce cuando un atacante consigue interceptar una sesión ya establecida. El atacante espera a que la víctima se identifique ante el sistema y tras ello le suplanta como usuario autorizado.

Este es el procedimiento:

1. El cliente establece una conexión con el servidor enviando un paquete que contendrá la dirección origen, destino, número de secuencia (para luego armar el paquete) y un número de autenticación utilizado por el servidor para reconocer el paquete siguiente en la secuencia.
2. El servidor luego de recibir el primer paquete contesta al cliente con paquete echo (recibido).
3. El cliente envía un paquete ACK (de confirmación) al servidor, sin datos, donde le comunica lo perfecto de la comunicación.
4. El atacante que ha visto, mediante un sniffer, los paquetes que circularon por la red calcula el número de secuencia siguiente: al actual + tamaño del campo de datos.
5. Hecho esto el atacante envía un paquete.

[52] Véase la nota 41
[53] Véase la nota 41

El servidor al recibir estos datos no detectará el cambio de origen ya que los campos que ha recibido como secuencia y ACK son los que esperaba recibir. El cliente, a su vez, quedará esperando los datos como si su conexión estuviera colgada y el atacante podrá seguir enviando datos mediante el procedimiento descrito.

Virus de Mail[54].

El último grito de la tecnología en cuestión de virus. Su modo de actuar, al igual que los anteriores, se base en la confianza excesiva por parte del usuario: a este le llega vía mail un mensaje con un archivo comprimido (.ZIP por ejemplo), el usuario lo descomprime y al terminar esta acción, el contenido (virus ejecutable) del archivo se ejecuta y comienza el daño.

Reproductores-Gusanos[55].

Son programas que se reproducen constantemente hasta agotar totalmente los recursos del sistema huésped y/o recompilar información relevante para enviarla a un equipo al cual su creador tiene acceso.

Caballos de Troya[56].

De la misma forma que el antiguo caballo de Troya de la mitología griega escondía en su interior algo que los troyanos desconocían, y que tenía una función muy diferente a la que ellos podían imaginar; un Caballo de Troya es un programa que aparentemente realiza una función útil pero además realiza una operación que el usuario desconoce y que generalmente beneficia al autor del troyano o daña el sistema huésped.

Consiste en introducir dentro de un programa una rutina o conjunto de instrucciones, no autorizadas y que la persona que lo ejecuta no conoce, para que dicho programa actúe de forma diferente a como estaba previsto.

[54] Véase la nota 41
[55] Véase la nota 41
[56] Véase la nota 41

Bombas Lógicas[57].

Este suele ser el procedimiento de sabotaje más comúnmente utilizado por los empleados descontentos. Consiste en introducir un programa o rutina que en una fecha determinada o dado algún evento particular en el sistema, bien destruye y modifica la información o provoca baja en el sistema.

4.3 Requerimientos básicos para aplicar el análisis forense.

Es necesario tener a la mano una computadora para la respuesta a incidentes. Algunas veces, cualquier computadora puede servir, pero es mejor estar preparado. Se puede usar la computadora para ayudar a buscar evidencia, además, servirá como repositorio de la misma.

-Computadora para respuesta a incidentes:

- oPC manejable
  - ▪2 discos IDE principales de 100 [Gb] (de preferencia).
  - ▪CD-RW
  - ▪Cinta
  - ▪Discos externos, recomendable100 [Gb] (interfaces IDE, SCSI)
- oLaptop
  - ▪Disco externo, de preferencia 100 [Gb] (interfaces IDE, SCSI)

Si no se tiene el tiempo para construir personalmente un sistema propio, entonces puede adquirirse uno. La computadora construida o adquirida puede servir para monitorear la red, recolectar datos, y en algunos casos, para hacer imágenes de discos.

Muchas veces es conveniente una PC por la facilidad en que se pueden remover elementos de su hardware. Sin embargo, también puede usarse una laptop. Sólo hay que mantener un disco configurado listo antes del incidente, ya que lo último que se desea es tener que configurar una computadora con un nuevo sistema operativo en unos minutos cuando se este trabajando en un incidente y de esta manera perder tiempo muy valioso.

[57] Véase la nota 41

También necesitamos un hub pequeño de 4 puertos y cable de red suficiente. Hay que considerar la posibilidad de hacer algún análisis desde el sistema "vivo" y será necesario pasar la información desde el sistema víctima al sistema forense.

También se necesita un CD o floppy de respuesta a incidentes para ayudar a buscar y recoger evidencia de forma segura, sin afectar al sistema víctima.

La mayoría de los programas se construyen dinámicamente usando bibliotecas compartidas en la computadora local. Se pueden construir cuidadosamente binarios estáticos o poner las bibliotecas compartidas en el CD-ROM y reconstruir los programas para que usen esas bibliotecas.

¿Qué tipo de herramientas? Es difícil definir exactamente qué será necesario, pero al menos debería contarse con netstat, dd, find, ls, ps, lsof, strings, last, ifconfig, uptime, etc.[58], junto con las herramientas forenses,

Toma mucho tiempo, paciencia y mucha voluntad crear un buen kit de herramientas para respuesta a incidentes. La mejor forma de hacerlo es probar y anticipar los sistemas que pueden sufrir intrusiones, aunque en realidad, eso es como tratar de anticipar "dónde" ocurrirá una intrusión. Una vez que se tiene práctica en juntar un grupo de herramientas decente, se va haciendo con mayor eficiencia. Es posible que se requiera un grupo de herramientas específico para cada kernel que esté corriendo sobre el sistema operativo.

Hay que tener presente que la elección de las herramientas impactará directamente a la metodología cuando se responde a un incidente. La respuesta dependerá de las herramientas que se tengan disponibles. En muchos casos, es difícil predecir este conjunto de herramientas ya que no se puede estar seguro sobre qué situación se va a lidiar. Cada situación puede requerir un conjunto de herramientas específico. Aquí es donde la planeación y prácticas previas son útiles. El orden específico en que pueden o deben usarse algunas herramientas se define con el conocimiento de las razones y efectos que tienen algunas de ellas.

[58] Véase Apéndice D

Por ejemplo, algo que borra una bitácora accidentalmente modifica una marca de tiempo crucial. Entonces, primero deben obtenerse estas bitácoras para la evidencia usando otra herramienta antes de usar la que hace la modificación.

Adicional al conjunto de herramientas principal, hay muchos otros elementos que deben estar disponibles cuando se realiza la investigación. Estos elementos incluyen, principalmente, cosas que ayudarán a documentar las actividades en el área. Las cámaras fotográficas y cuadernos de notas son elementos esenciales para cualquier investigador. Los dispositivos de registro como cámaras ayudarán posteriormente, ya que la mayoría de los incidentes suceden muy rápido y es difícil contabilizar todas las actividades en la escena. Hay que eliminar la posibilidad de probar y recordar cosas, es mejor documentarlas con elementos fáciles de recordar.

Algo que definitivamente es necesario es un resguardo. ¿Cómo se puede confiar y testificar sobre la correcta cadena de custodia si no se tuviera la posibilidad de mostrar que la evidencia no estuvo fuera de control en ningún momento?

CAPÍTULO V

IMPLANTACIÓN DE UN HONEYPOT Y LA APLICACIÓN DEL ANÁLISIS FORENSE

5.1 Diseño de la topología.

El diseño de la topología es el siguiente; se tiene un equipo configurado con un firewall que da salida a cinco equipos a Internet, además todos se encuentran conectados a un switch que conforma la red local en la que se encuentran los cinco equipos antes mencionados más el firewall, en estas condiciones se configuró un HoneyPot y un sistema de control para analizar los diferentes ataques que pudiesen suscitarse o a los cuales se va a someter el sistema a analizar , en tanto que los demás equipos conectados en la red local son los posibles atacantes, como se muestra en la figura 5.1.

Cabe recalcar que los ataques suscitados en las prácticas no se propagarán a otros equipos porque se configuró una red sin conexión a Internet utilizando únicamente un switch desconectando el cable que conecta el firewall con el switch.

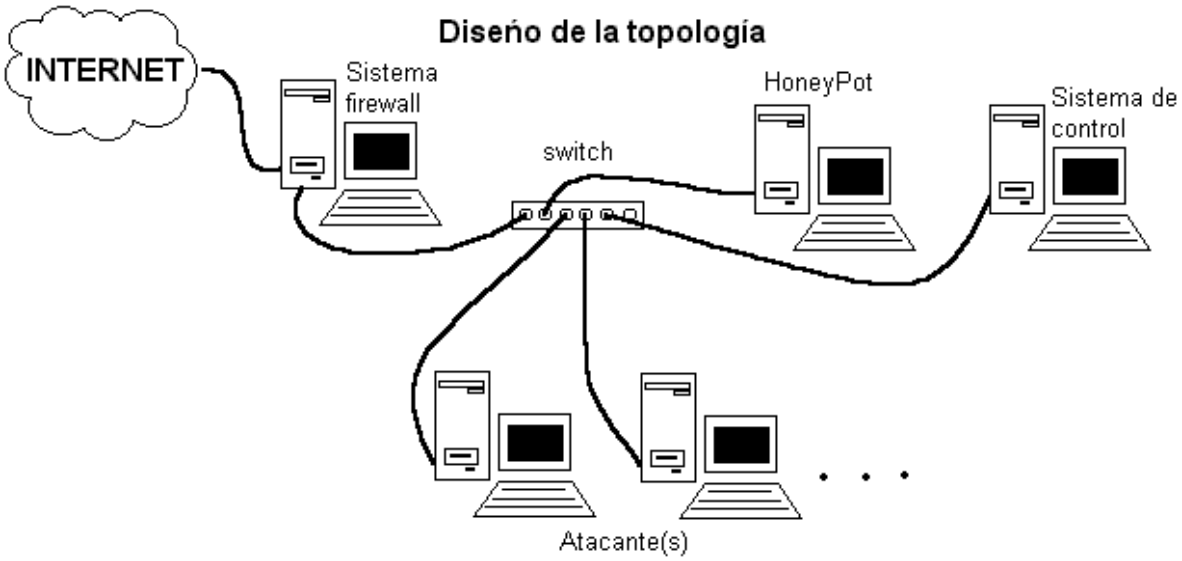


Fig 5.1 Diseño de la topología

La información documentada de los siguientes casos prácticos se encuentran en el Apéndice C.

Las direcciones IP de los casos prácticos no se muestran por cuestiones de seguridad.

5.2 Caso Práctico 1

5.2.1 Preparación para el análisis

Para realizar este caso se instaló a una computadora el sistema operativo Red Hat 6.2 el cual se dejó a cargo de un administrador del laboratorio de Redes. Esta persona se encargó de configurar servicios, ssh, samba, Sendmail, FTP, WEB Apache y Base de Datos MySQL, asimismo dicho administrador agregó usuarios para el sistema, que pueden hacer uso del mismo de forma legítima.

En la entrevista realizada al administrador en cuestión descubrimos que el password de root solo lo posee el responsable de la administración del sistema, también esta persona realizó la última instalación del equipo, desde hace un año a la fecha en la que se está realizando esta investigación es quien lo ha estado administrando y no se ha actualizado el sistema ni se le ha realizado antes una auditoría, aunque cabe comentar que justamente está por llevarse a cabo una actualización de equipos y sistema. La razón que expuso el administrador para indicarnos su sospecha de un incidente fue que las bitácoras del sistema fueron limpiadas, de esto se percataron el 15 de Junio como a las 18:00 horas, antes de esta fecha el sistema funcionaba correctamente. El equipo comprometido no tiene relaciones de confianza con otros equipos, por lo que creemos que no fueron afectados por el incidente, además el responsable de la administración del sistema no hace sus labores administrativas de manera remota.

Siguiendo con la entrevista nos percatamos que el equipo comprometido no cuenta con un Server log, ni con alguna herramienta de seguridad o respaldos del sistema, también no se contaba con algún IDS para analizar la red, ni se tenía nada resguardado en el momento en que se involucró el sistema y actualmente se cuenta con políticas de seguridad y ésta es totalmente prohibitiva de manera que sólo se habiliten los puertos necesarios para las prácticas o proyectos de investigación que se vayan a realizar.

El administrador del equipo revisó los registros del sistema percatándose que habían sido borradas algunas de ellas, finalmente apagó el equipo por la fuerza oprimiendo el botón de encendido, por lo que procedimos a realizar el análisis de un

sistema muerto. El administrador del sistema está interesado en saber qué fue lo que pasó.

5.2.2 Almacenamiento de Pruebas

En otro equipo que nosotros preparamos para el análisis montamos el disco duro del equipo afectado para poder realizar la copia correspondiente. Al montar el disco duro se hizo de manera sólo lectura, sin permitir la ejecución de comandos y sin modificar marcas de tiempo, como se mencionó en el capítulo dos, la clave de la computación forense se basa en el análisis de los medios de almacenamiento.

De acuerdo con nuestra metodología, iniciamos esterilizando el medio de almacenamiento para evitar contaminación del mismo. Posteriormente hicimos una copia del disco duro en el medio esterilizado generando firmas digitales con el método MD5, ya que la cadena de custodia[59] nos obliga a realizar un análisis sobre información veraz de los archivos de sistema, las firmas se realizaron tanto del disco original como de la copia y comparamos dichas firmas para ver si coincidían, de esta manera pudimos obtener una copia fiel del disco duro del sistema.

5.2.3 Análisis con herramientas estándares de UNIX

Ya obtenida la copia comenzaremos con una búsqueda que tiene como objetivos encontrar archivos especiales, RootKits, archivos comprimidos, archivo ocultos, archivos binarios, programas sospechosos en lenguajes de programación como perl o C, correo electrónico sospechoso, archivos de Internet, bitácoras y los tiempos MAC.

Primero realizamos un análisis forense básico, es decir, ejecutamos comandos básicos de Linux que pudieran revelarnos la información descrita en el párrafo anterior. Lo encontrado con esto fue lo siguiente:

Detectamos un .bash_history sobre el directorio raíz lo cual no es normal porque el directorio personal del administrador es “/root” y no “/”, en bitácora messages se detectó

[59] Véase Apéndice C

una sesión cerrada por el usuario1, siendo esta bitácora incongruente porque debía de encontrarse su respectiva conexión al sistema.Las bitácoras tenían información registrada solo después del incidente, ya que el atacante borró el contenido de las bitácoras desde el día en que se comprometió el sistema y anteriores. El sistema fue comprometido aproximadamente el 15 de junio a las 14:36 horas.

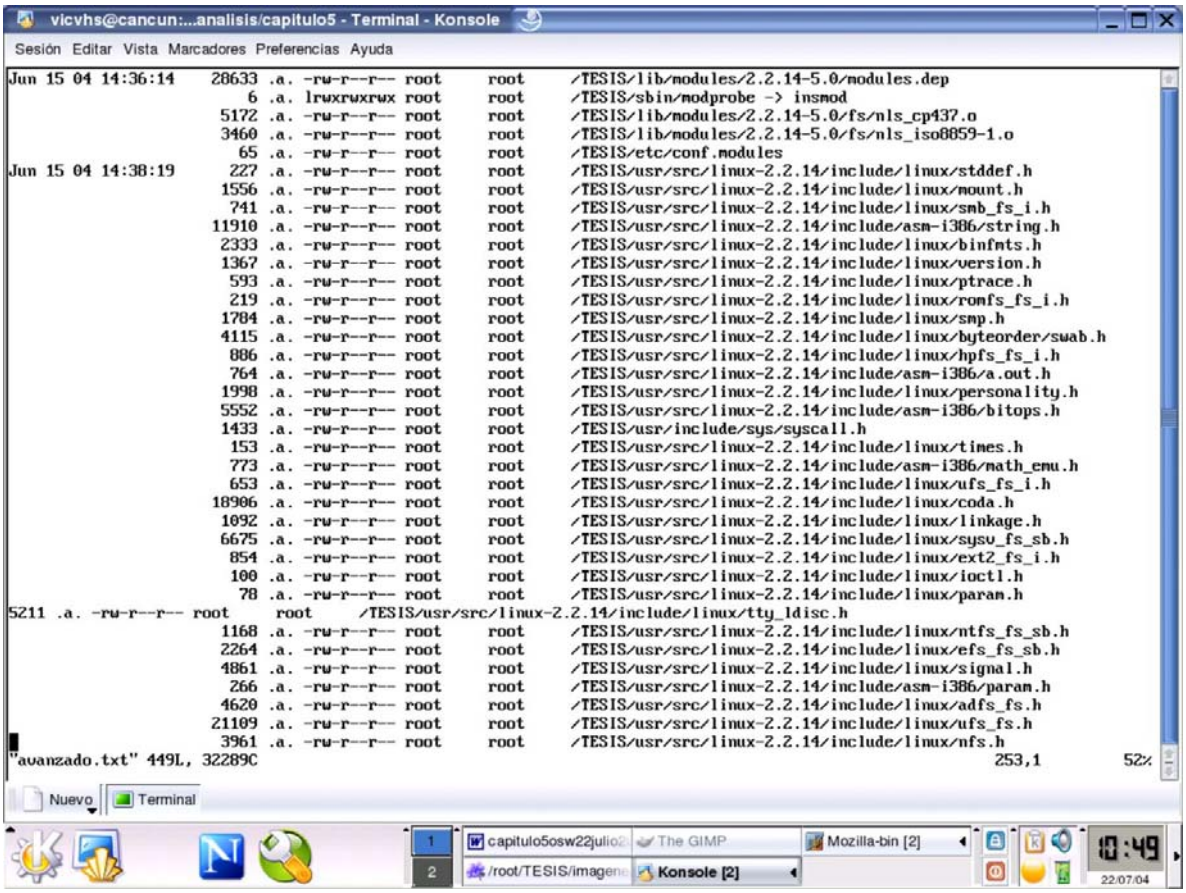


Fig 5.2 Tiempos mac-1 práctica 1

Como se muestra en la figura 5.2 encontramos que se tuvo acceso a distintas librerías y archivos de los módulos del núcleo del sistema a la hora que se había mencionado. Los accesos a estas librerías es una situación anormal dentro de un sistema en producción.

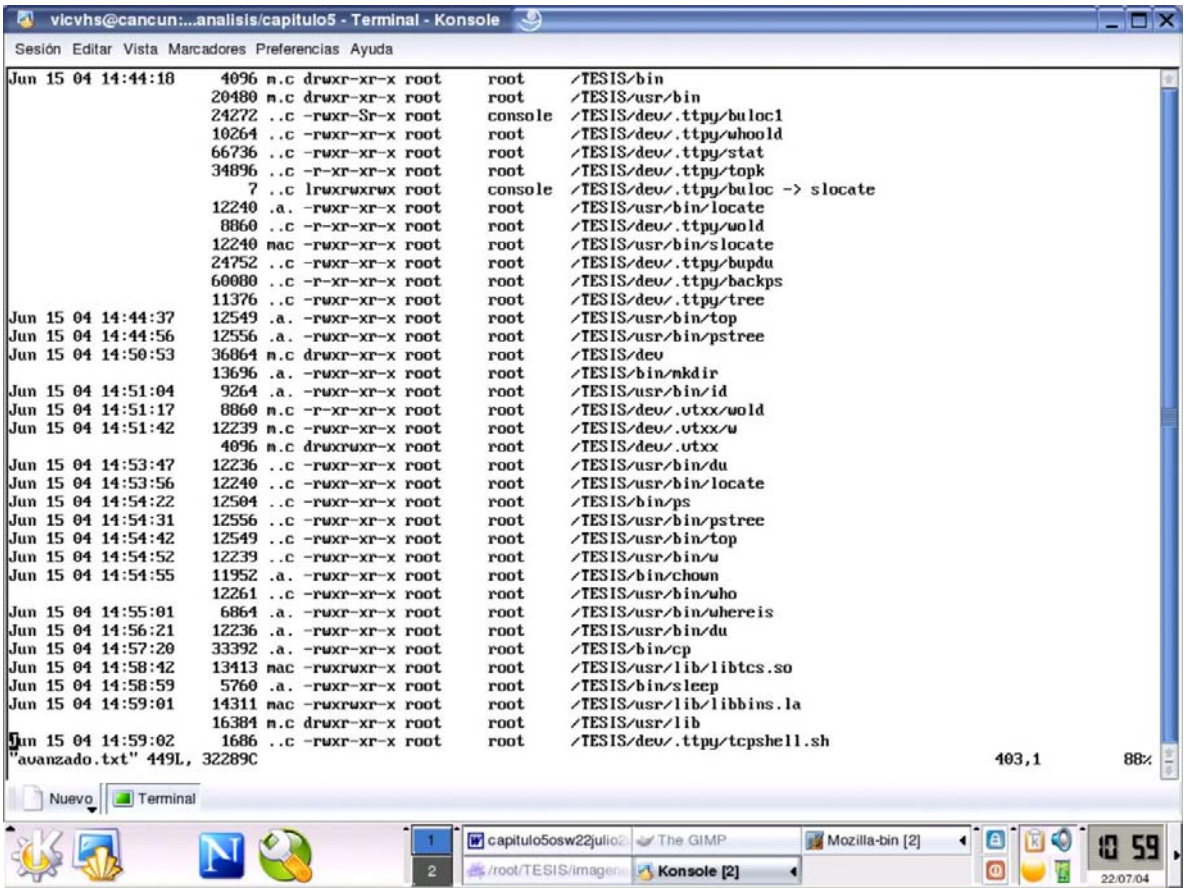


Fig. 5.3 Tiempos mac-2 práctica 1

En la figura 5.3 se observa cómo se ha instalado un rootkit el cual se creó en el directorio /dev/. Por los directorios creados (.ttpy y el .vtxx) y por todos los archivos colocados dentro de estos directorios mencionados creemos que el usuario instaló el rootkit Adore.

Encontramos dos archivos ejecutables con los nombre w y wold dentro del directorio .vtxx; además, bajo el directorio .ttpy existen once archivos que se modificaron el 15 de junio a las 14:44 horas.

En maillog se encontró un correo enviado por el superusuario root a usuario gene@localhost. Es decir, se trató de enviar un correo a un usuario sospechoso, posteriormente nos percatamos que este usuario es propio de un rootkit que se puede encontrar en la red.

En wtmp se encontró un acceso al sistema del usuario1 el 15 de Junio a las 14:36 horas y cerró sesión a las 15:26 horas, tiempo en el que realizó el ataque, también encontramos que el usuario1 no tiene .bash_history.

Dentro de los archivos borrados que recuperamos en este análisis, encontramos un exploit para sendmail, un programa que ocultaba módulos del Kernel, un Script que carga el módulo del Root Kit de Adore y que lo oculta, finalmente encontramos un exploit en Perl.

5.2.4 Análisis con herramientas forenses

Posteriormente se procedió realizar un análisis forense avanzado, en este análisis utilizamos herramientas para recuperar evidencia, ya que existe una gran cantidad de datos y formatos almacenados sin que aún podamos determinar cuál es su finalidad, además de que esta recopilación debe ser exacta y rápida, como se mencionó anteriormente, la selección de herramientas forenses adecuadas influenciará en gran medida el resultado de la investigación.

Para este análisis utilizamos las siguientes herramientas forenses: TCT, Unrm, Autopsy y Lazarus. Nosotros asistimos al curso "Análisis Forense e Implicaciones Legales" impartido por la Dirección General de Servicios de Cómputo Académico, DGSCA, en el cual se nos proporcionó las herramientas necesarias para elaborar los casos prácticos de este trabajo.

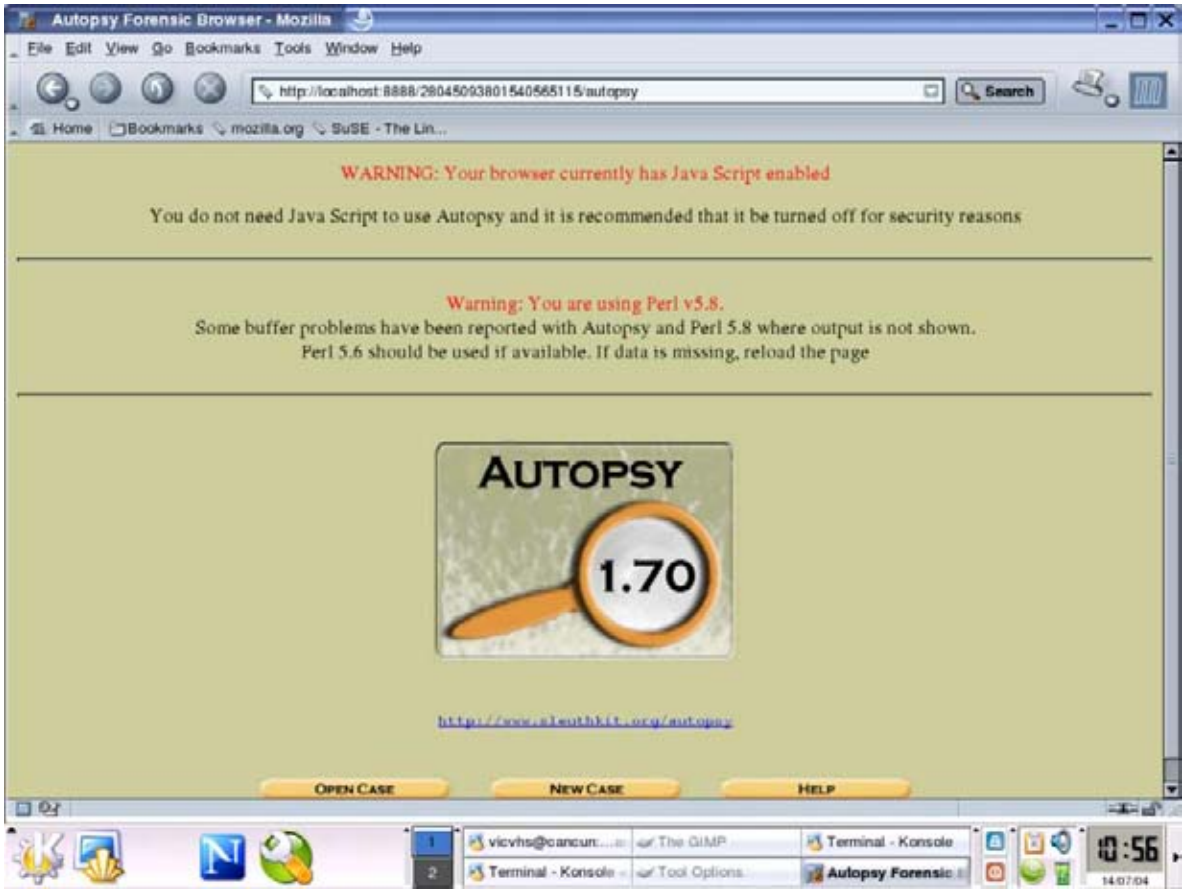


Fig 5.4 Autposy-1 práctica 1

En la figura 5.4 vemos la pantalla de bienvenida de la herramienta forense Autopsy que interactúa con los resultados arrojados por la herramienta The Sleuth Kit, ambas nos facilitarán el análisis de los tiempos MAC y sobre los i-nodos borrados, es decir, archivos de todo el sistema que fueron eliminados.

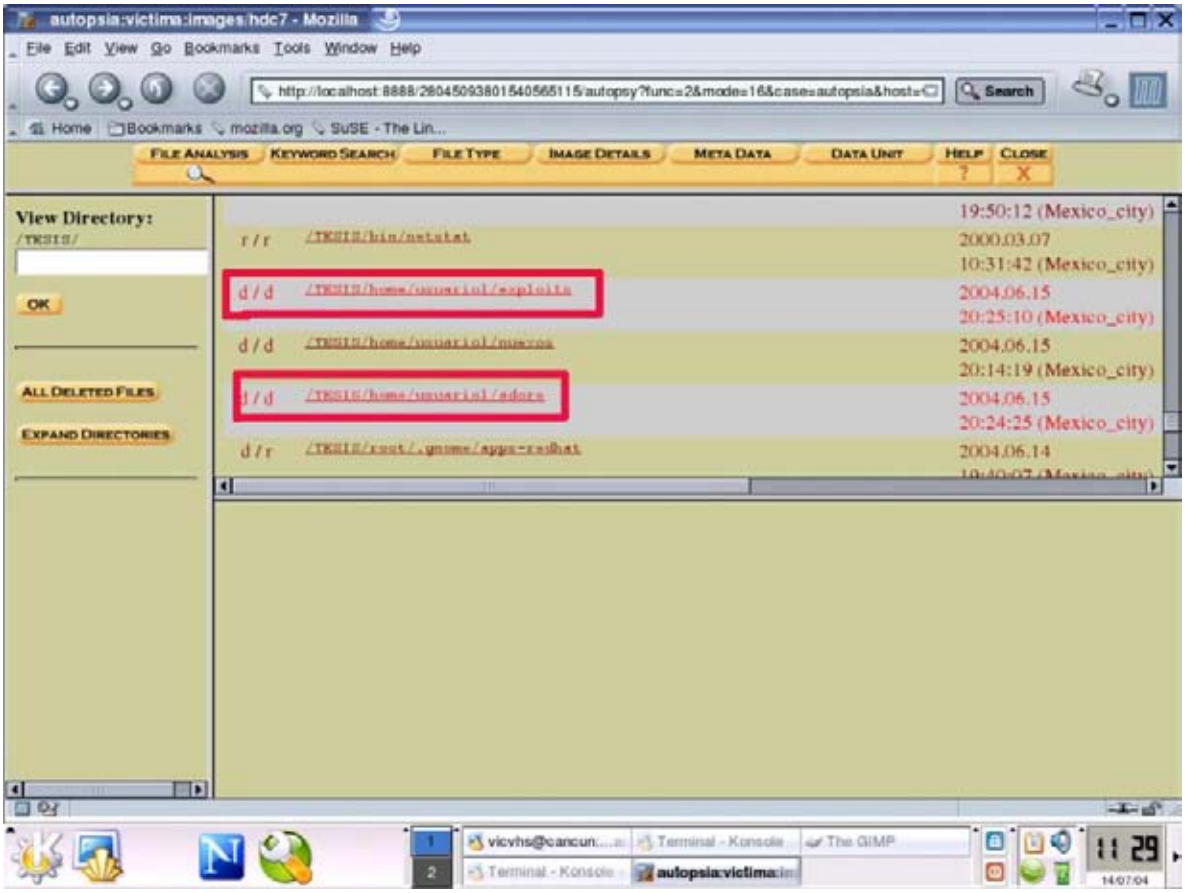


Fig. 5.5 Autposy-2 práctica 1

En la Figura 5.5 se resalta con cuadros la existencia de dos directorios borrados: uno llamado exploits y otro con el nombre de adore. Dentro de Autopsy encontramos muchos archivos borrados en el sistema, y varios de ellos fueron borrados en el lapso en que se efectuó el ataque. La ventaja de esta herramienta es que dando un sólo clic en alguno de los archivos, podemos observar su contenido aún si es un archivo borrado. Cabe aclarar, que los horarios mostrados en la Figura 5.5 corresponden a los horarios tomados del sistema en el cual se copiaron las imágenes respaldadas del equipo comprometido, por tanto, no concuerdan en lo absoluto con la hora de la intrusión.

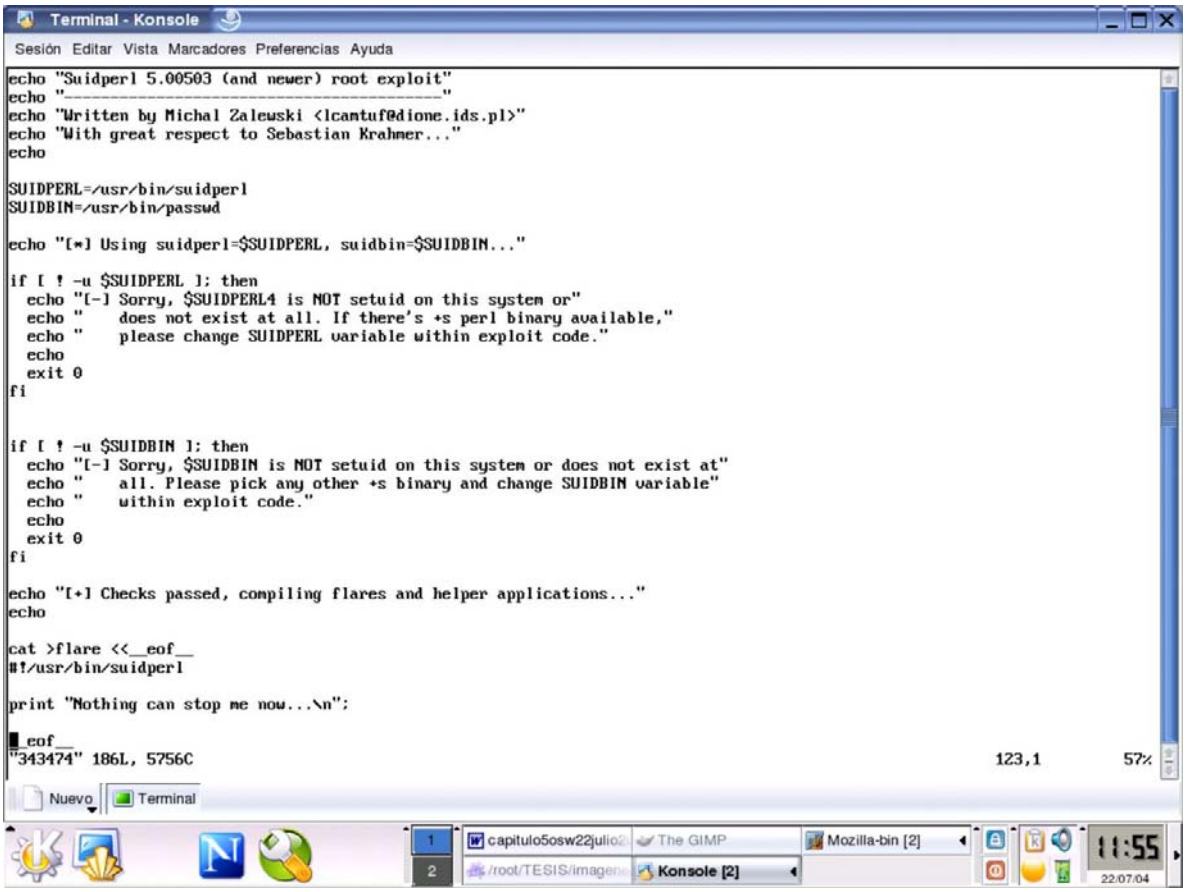


Fig. 5.6 Autopsy-3 práctica 3

Uno de los exploits que recuperamos que pertenecían al usuario1 se muestra en la figura 5.6. Este exploit se apodera de una sesión de administrador del sistema explotando una vulnerabilidad en SuidPerl de Red Hat 6.2. Esta recuperación se realizó con The Coroner's Toolkit.

Al ejecutar estas herramientas forenses encontramos la misma información que se encontró con las herramientas básicas de UNIX, nada más que en este caso, hemos podido determinar paso a paso el orden cronológico de los hechos y además, cómo fueron accedidos y modificados todos los archivos del sistema. Para el caso de la herramienta Lazarus se requirió mucho procesamiento en máquina para la obtención de resultados.

5.2.5 Análisis de Resultados y Reporte Final

Nuestros principales objetivos como investigadores forenses en ésta y todas las investigaciones posteriores son los siguientes:

- Encontrar señales de que el sistema está comprometido.
- Determinar el tamaño del daño.
- Responder: qué, cuándo, dónde, cómo.
- Reconstrucción cronológica de los hechos.
- Manejar y analizar la evidencia de tal forma que pueda servir para demostrar los hechos.

Trataremos de reconstruir cómo el intruso realizó su reconocimiento, la explotación del sistema y las operaciones ocultas, para que de esta forma podamos elaborar un perfil del atacante y su propósito.

El equipo xxx.xxx.xxx.xxx fue instalado el día 12 de Junio del 2004. Se instaló el sistema operativo Red Hat 6.2 sin ninguna actualización.

El sistema fue comprometido el día 15 de Junio del 2004 entre las 14:36 horas y las 15:26 horas.

El sistema comprometido tenía los servicios de red activos, X11, portmap, mysql, httpd, wu-ftpd, samba, ssh, sendmail. Los servicios httpd, wu-ftpd, samba, sendmail tenían importantes problemas de seguridad.

Hasta aquí concluimos que el equipo fue comprometido por un usuario interno del sistema, por lo que el ataque fue perpetuado por un insider: el usuario con el login ‘usuario1’. Este usuario descargo varios tipos de exploit, encontrando programas que explotan vulnerabilidades de sendmail y también de SuidPerl; el primero de ellos escrito en lenguaje C y el segundo en Shell Script.

El ‘usuario1’ ingresó al sistema de forma legítima, una vez dentro, descargó los exploits de sendmail y de SuidPerl. Una vez vulnerado el sistema y obtenido una sesión

de administrador, se dispuso a instalar un RootKit llamado Adore que reside en el núcleo como módulo, interceptando determinadas llamadas de sistema para lograr su objetivo.

De esta forma, Adore puede ocultar síntomas que indicarían la presencia de un intruso en el sistema, tales como conexiones desde direcciones remotas desconocidas, intérpretes de comandos con privilegios de administrador que estén ejecutándose, o ficheros con contraseñas capturadas.

Adore se encargó de las siguientes tareas:

* Ocultar ficheros y directorios
* Ocultar procesos
* Ocultar conexiones de red

Adore se instala en los directorios ocultos /dev/.vtty y /dev/.vtxx. Este programa se encargó de mandar el correo sospechoso al usuario gene@localhost. Una vez que tenía privilegios de administrador y con el rootkit instalado se dispuso a borrar todas las bitácoras para eliminar sus huellas, incluyendo su historial de comandos y los exploits que descargó junto con el rootkit, pero olvidó borrar posteriormente las bitácoras que registran la salida de sesión del usuario.

Como se ha podido ver el tiempo transcurrido desde la instalación del sistema hasta su compromiso ha sido de tres días. El sistema contaba con muchos servicios activos y ninguna actualización. De los servicios activos cuatro de ellos y programas internos como lo es SuidPerl tienen serios problemas de seguridad.

Se recomienda utilizar mecanismos de seguridad aparte de los passwords, como lo son: firewall interno, programas que verifiquen la integridad del sistema, herramientas que detecten vulnerabilidades en el sistema. También se recomienda mantener actualizado el sistema y sería conveniente colocar un sistema operativo reciente como lo es Red Hat 9.

5.3 Caso Práctico 2

5.3.1 Preparación para el análisis

En la entrevista nos percatamos que el password de root solo lo posee el responsable de la administración del sistema, y realizó la última instalación del equipo. Esta persona es responsable del sistema desde hace un año y además nunca antes se había tenido la necesidad de actualizar y de realizar una auditoria porque es un sistema dedicado a la investigación de seguridad en cómputo para este trabajo de tesis.

Se instaló a una computadora el sistema operativo Red Hat 7.1 el cual se dejó a cargo de un administrador del laboratorio de Redes. Esta persona se encargó de configurar los servicios, FTP, NFS, Impresión, SSH y Correo. Este administrador creó un usuario para el sistema.

Otros sistemas no fueron afectados por el incidente ya que el equipo comprometido no tiene relaciones de confianza de ningún tipo; nunca se ha administrado el sistema remotamente. La razón por la que el administrador sospecha la existencia de un incidente es que se encontraron conexiones extrañas de FTP el 23 de Julio como a las 9:00 horas, antes de esta fecha el sistema funcionaba correctamente.

La entrevista nos reveló que no se disponen de respaldos, el equipo comprometido no cuenta con un Server log o un IDS ni con alguna herramienta de seguridad.

Cuando se descubrieron estas conexiones extrañas de FTP, se apagó el sistema de manera convencional y se puso en runlevel 3.

Ahora nosotros procederemos a realizar el análisis de un sistema muerto.

5.3.2 Almacenamiento de pruebas

Siguiendo nuestra metodología se crearon imágenes del disco duro, se realizaron las respectivas firmas digitales con el método MD5 para compararlas con las firmas del original y de esta forma verificar que la información sea íntegra, posteriormente estas

imágenes se montaron en un disco esterilizado de manera solo lectura, sin permitir la ejecución de comandos y sin modificar marcas de tiempo para evitar que la información se altere.

5.3.3 Análisis con herramientas estándares de UNIX

Procedimos a analizar el sistema para encontrar evidencia suficiente y así llegar a una conclusión satisfactoria. Encontramos que el runlevel en el cual se encontraba el sistema era efectivamente el nivel 3, sin ambiente gráfico. El sistema es un Red Hat 7.1.

Procedimos a buscar archivos ocultos dentro del sistema encontrando un directorio oculto, por lo que revisamos que archivos han sido modificados posteriormente a este directorio:

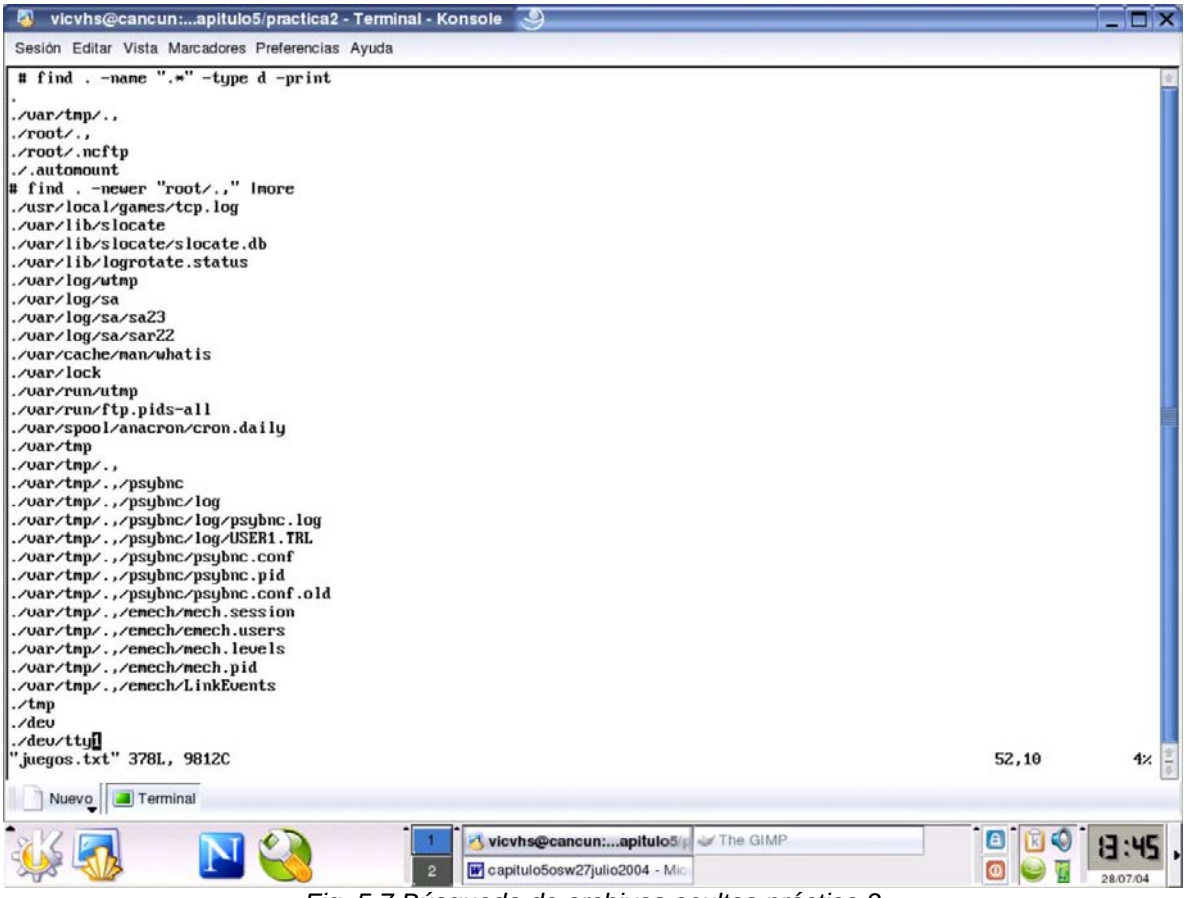


Fig. 5.7 Búsqueda de archivos ocultos práctica 2

Encontramos que el intruso creó dos directorios ocultos, uno bajo el directorio root y el segundo dentro del directorio /var/tmp, ambos con el nombre ".,", como se muestra en al figura 5.7. Dentro del directorio /root/., descargo dos paquetes los cuales al parecer son un cliente IRC(Internet Relay Chat) y el otro un scaneador de máquinas el cual busca vulnerabilidades en el servicio ftp. Dentro del directorio /var/tmp., localizamos dos directorios los cuales al parecer son también dos clientes de IRC.

Respecto a la revisión de bitácoras, lo que encontramos fueron varias conexiones en el servicio de ftp iniciándose desde las 08:47 horas del día 22 de Julio del 2004. Este suceso fue constante durante un periodo de aproximadamente 30 minutos. Posteriormente, al día siguiente se inician de nuevo conexiones al servicio de ftp a las 08:30:31 y finalizando a las 9:22:47.

Dentro del directorio /etc/init.d se detectaron varias modificaciones en scripts de inicio de los servicios de functions, syslogd, sshd y atd. Estos demonios se encargan de definir un conjunto de funciones que son utilizadas en el resto de los scripts, envío de los registros a los archivos de bitácoras, permitir conexiones de ssh y establecer la ejecución de tareas at, respectivamente.

Dentro del archivo de contraseñas se encontraron dos nuevos usuarios, uno de ellos con el login 'ssh' y con el id 0, el cual es el que utiliza el administrador del sistema; el segundo agregado al archivo de contraseñas es el usuario 'nerod' con el id 501. Cabe recalcar que el usuario 'ssh' no cuenta con password, por lo que en cualquier momento se puede acceder con dicha cuenta.

Se encontró una entrada en el crontab, el cual enviaba periódicamente información a una cuenta de correo de yahoo.com. Esta información consistía en el estado de las tarjetas de red, del nombre del host, y los registros generados de un sniffer que el intruso descargó en el directorio /usr/local/games.

5.2.4 Análisis con herramientas forenses

Lo primero que realizamos fue ejecutar un detector de rootkits para conocer si es un nuevo rootkit o uno ya conocido que se puede descargar de la red fácilmente. Esta herramienta lo que hace es verificar el estado de comandos, archivos extraños en directorios en los cuales no deben existir este tipo de archivos. La salida generada por este comando es la mostrada en la figura 5.8:

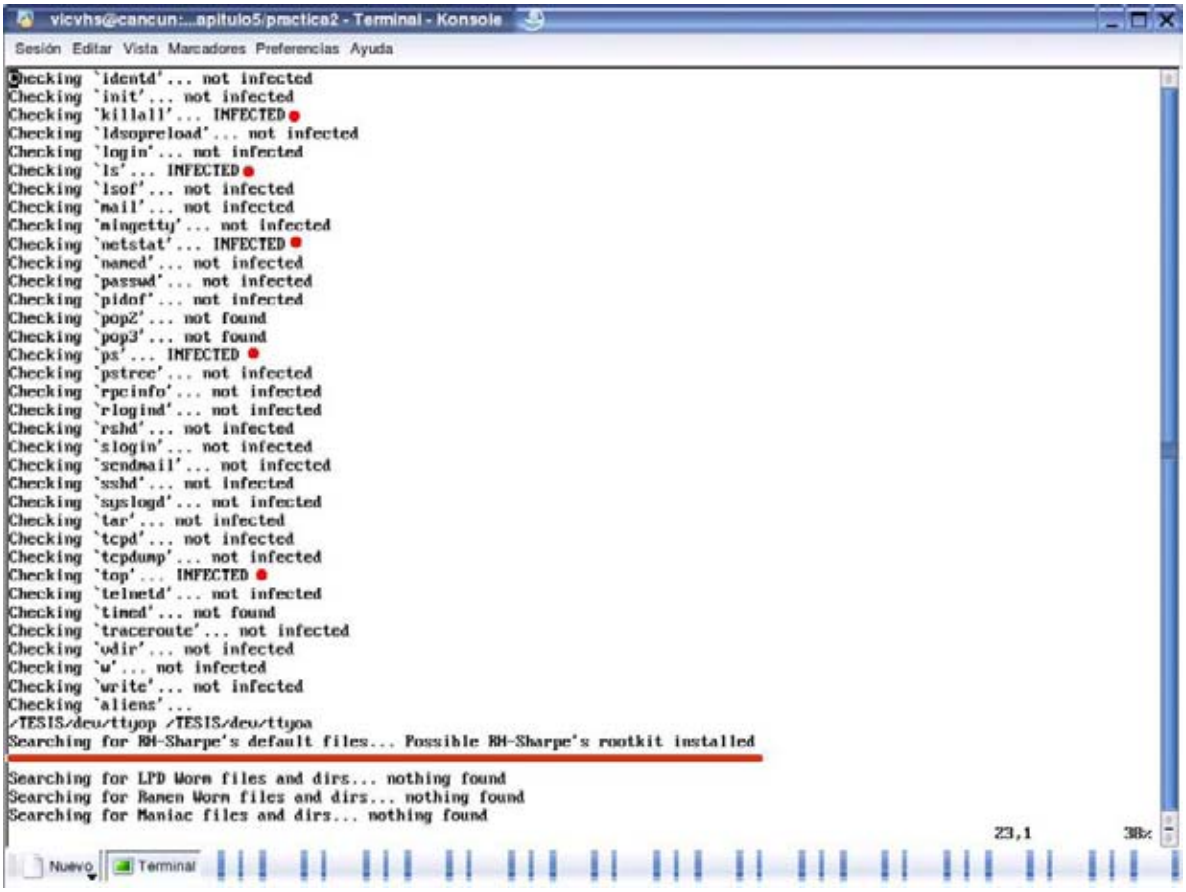


Fig. 5.8 Rhkrootkit práctica 2

En la Fig. 5.8 los renglones marcados con un punto nos indican que Rhkrootkit ha encontrado comandos del sistema comprometidos o infectados (INFECTED). Buscamos comandos que muestran información de memoria, procesos, estado de la red, conexiones, listado de directorios, etc.; estos comandos son alterados muy frecuentemente. En el renglón subrayado de la Fig. 5.8 vemos como esta herramienta busca los patrones comunes del rootkit RH-Sharpe, lo cual determina que es posible que éste sea el rootkit instalado, lo que nos indica que es una variante de este rootkit con

algunas modificaciones. Rhkrootkit busca otro tipo de rootkits que estén posiblemente instalados, pero no se ha encontrado ningún otro.

Se obtuvo un registro de todos los archivos con sus permisos, al dueño y grupo al que pertenecen, tamaño del archivo, inodo, ligas relacionadas al archivo, tamaño de los bloques de dispositivo, y algo muy importante, los tiempos MAC (modificación, acceso y creación de los archivos). Con esto podemos determinar que archivos dentro de todo el sistema comprometido fueron alterados en un instante, únicamente accedidos o utilizados, por si es el caso de un comando que se haya ejecutado.

Como se puede observar en la salida de la herramienta "grave-robber" que generó toda esa información llevando un registro de todos los archivos así como los que fueron creados por el intruso (el directorio /root/.,/ que se encuentra subrayado) , en este caso sólo se muestra un fragmento del contenido del directorio /root/ que pertenece al administrador del sistema. Dentro del directorio /root/.,/ encontramos que el directorio aw y el directorio psybnc fueron generados por archivos comprimidos que el intruso descargó en la máquina comprometida una vez que ingresó a ella, como se muestra a continuación:

/root/.Xresources|1792|60245|33188|-rw-r--r--|1|0|0|0|1126|1030098998|809204558|1029954787|4096|4
/root/.bash_logout|1792|60246|33188|-rw-r--r--|1|0|0|0|24|1030098998|960670815|1029954787|4096|2
/root/.bash_profile|1792|60247|33188|-rw-r--r--|1|0|0|0|266|1030098998|960671044|1029954787|4096|2
/root/.bashrc|1792|60248|33188|-rw-r--r--|1|0|0|0|176|1030098998|809204670|1029954787|4096|2
/root/.cshrc|1792|60249|33188|-rw-r--r--|1|0|0|0|210|1030098998|960671342|1029954787|4096|2
/root/.tcshrc|1792|60250|33188|-rw-r--r--|1|0|0|0|196|1030098998|963330791|1029954787|4096|2
/root/.bash_history|1792|60350|33152|-rw-------|1|0|0|0|1109|1030098998|1030098717|1030098717|4096|4
/root/.,|1792|6080|16877|drwxr-xr-x|4|0|0|0|1024|1030068191|1030059778|1030059778|4096|2
/root/.,/aw|1792|12171|16877|drwxr-xr-x|2|0|0|0|1024|1030068191|1030059811|1030059811|4096|2
/root/.,/aw/auto|1792|12184|33261|-rwxr-xr-x|1|0|0|0|205|1030059778|1011741687|1030059778|4096|2
/root/.,/aw/awu|1792|12187|33261|-rwxr-xr-x|1|0|0|0|1291|1030059811|1011741627|1030059778|4096|4
/root/.,/aw/awu.list|1792|12188|33188|-rw-r--r--|1|0|0|0|231|1030059778|1011755118|1030059778|4096|2
/root/.,/aw/Makefile|1792|12189|33188|-rw-r--r--|1|0|0|0|597|1030059778|1011652706|1030059778|4096|2
/root/.,/aw/nodupe.c|1792|12190|33188|-rw-r--r--|1|0|0|0|5550|1030059778|1011509187|1030059778|4096|12
/root/.,/aw/oops.c|1792|12191|33188|-rw-r--r--|1|0|0|0|1060|1030059778|1011511396|1030059778|4096|4
/root/.,/aw/pscan2.c|1792|12192|33261|-rwxr-xr-x|1|0|0|0|5870|1030059778|1011721037|1030059778|4096|12
/root/.,/aw/ss.c|1792|12193|33188|-rw-r--r--|1|0|0|0|6220|1030059778|1017952148|1030059778|4096|14
/root/.,/aw/ssvuln.c|1792|12194|33261|-rwxr-xr-x|1|0|0|0|3350|1030059778|1011652374|1030059778|4096|8
/root/.,/aw/targets|1792|12195|33184|-rw-r-----|1|0|0|0|5015|1030059778|1011509187|1030059778|4096|10

/root/.,/aw/wu|1792|12196|33261|-rwxr-xr-x|1|0|0|0|382072|1030059778|1011511284|1030059778|4096|754
/root/.,/aw/x2|1792|12197|33261|-rwxr-xr-x|1|0|0|0|1393996|1030059778|1012254719|1030059779|4096|2738
/root/.,/aw/nodupe.o|1792|12198|33188|-rw-r--r--|1|0|0|0|3848|1030059778|1017952244|1030059779|4096|8
/root/.,/aw/oops.o|1792|12199|33188|-rw-r--r--|1|0|0|0|1656|1030059778|1017952245|1030059779|4096|4
/root/.,/aw/oops|1792|12200|33261|-rwxr-xr-x|1|0|0|0|13085|1030059778|1017952245|1030059779|4096|28
/root/.,/aw/outpu|1792|12201|33188|-rw-r--r--|1|0|0|0|3715|1030059778|1017107164|1030059779|4096|8
/root/.,/aw/doit4me|1792|12202|33188|-rw-r--r--|1|0|0|0|389|1030059778|853820889|1030059779|4096|2
/root/.,/aw/ss|1792|12203|33261|-rwxr-xr-x|1|0|0|0|16964|1030059778|1017952246|1030059779|4096|36
/root/.,/aw/awu.log|1792|12204|33188|-rw-r--r--|1|0|0|0|23064|1030059778|853798912|1030059779|4096|48
/root/.,/aw/test.c|1792|12205|33188|-rw-r--r--|1|0|0|0|0|1030059778|853810832|1030059779|4096|0
/root/.,/aw/test.f|1792|12206|33188|-rw-r--r--|1|0|0|0|5|1030059778|853810849|1030059779|4096|2
/root/.,/aw/pscan2|1792|12207|33261|-rwxr-xr-x|1|0|0|0|15781|1030059811|1017952245|1030059779|4096|34
/root/.,/aw/nodupe|1792|12208|33261|-rwxr-xr-x|1|0|0|0|15138|1030059778|1017952246|1030059779|4096|32
/root/.,/aw/ssvuln|1792|12209|33261|-rwxr-xr-x|1|0|0|0|15012|1030059778|1017952246|1030059779|4096|32
/root/.,/aw/12.216.pscan.21|1792|12210|33188|-rw-r--r--|1|0|0|0|0|1030059811
|1030059811|1030059811|4096|0
/root/.,/psyBNC.tar.gz|1792|6081|33188|-rw-r--r--
|1|0|0|0|557964|1030058251|1030058215|1030058215|4096|1098
/root/.,/psybnc|1792|42351|16877|drwxr-xr-x|7|0|0|0|1024|1030068191|1030088888|1030088888|4096|2
/root/.,/awu.tgz|1792|6082|33188|-rw-r--r--|1|0|0|0|1603918|1030059779|1028926584|1030059711|4096|3150
/root/.,/psybnc/tools|1792|56288|16893|drwxrwxr-x|2|500|500|0|1024|1030068191|972141628|1030058251|4096|2
/root/.,/psybnc/tools/convconf|1792|56289|33277|-rwxrwxr-
x|1|500|500|0|957088|1030058257|972721081|1030058251|4096|1880
/root/.,/psybnc/CHANGES|1792|42355|33188|-rw-r--r--
|1|500|500|0|18124|1030058250|972719983|1030058250|4096|38

De forma más legible podemos observar en la siguiente salida del comando “mactime” las fechas de modificación, creación y acceso de los archivos. Los tiempos MAC se encuentran en el tercer campo (en la octava línea se subraya), los permisos del archivo, dueño, grupo y por último el nombre del archivo.

En este caso mostramos los archivos con sus tiempos MAC y la hora correspondiente, se observa el momento exacto en que se efectúa la intrusión y la posterior descarga del archivo /var/ftp/nerod.tar.gz (subrayado en la quinta línea) que trae consigo el rootkit que el intruso instaló en el sistema posteriormente.

Jul 23 04 09:22:47 2132 m.c -rw------- root root /var/log/secure
Jul 23 04 09:22:48 104 .a. -rw------- root root /etc/ftphosts
168 .a. -rw------- root root /etc/ftpusers

```
                4096 mac -rw-r--r-- root     root     /var/run/ftp.rips-all
Jul 23 04 09:24:19   544317 m.c -rw-r--r-- root     root     /var/ftp/nerod.tar.gz
Jul 23 04 09:24:45     1024 m.c drwxr-xr-x root     root     /var/ftp
              589824 .a. -rwxr-xr-x 503      503      /var/ftp/nerod/core
                   0 .ac -rwxr-xr-x 503      503      /var/ftp/nerod/sshd/install.log
                 336 .a. -rwxr-xr-x 503      503      /var/ftp/nerod/sshd/ssh_host_key.pub
                1425 .a. -rwxr-xr-x 503      503      /var/ftp/nerod/inet
                 512 .a. -rwxr-xr-x 503      503      /var/ftp/nerod/sshd/ssh_random_seed
                 880 .a. -rwxr-xr-x 503      503      /var/ftp/nerod/sshd/ssh_config
                1047 .a. -rwxr-xr-x 503      503      /var/ftp/nerod/install.log
               19840 .ac -rwxr-xr-x 503      503      /var/ftp/nerod/sshd/ifconfig
Jul 23 04 09:24:46      194 ..c -rwxr-xr-x 503      503      /var/ftp/nerod/me
              544317 .a. -rw-r--r-- root     root     /var/ftp/nerod.tar.gz
                 336 ..c -rwxr-xr-x 503      503      /var/ftp/nerod/sshd/ssh_host_key.pub
                8368 ..c -rwxr-xr-x 503      503      /var/ftp/nerod/imp
                 278 ..c -rwxr-xr-x 503      503      /var/ftp/nerod/crontab-entry
                 512 ..c -rwxr-xr-x 503      503      /var/ftp/nerod/sshd/ssh_random_seed
                1425 ..c -rwxr-xr-x 503      503      /var/ftp/nerod/inet
              589824 ..c -rwxr-xr-x 503      503      /var/ftp/nerod/core
                 538 ..c -rwxr-xr-x 503      503      /var/ftp/nerod/.1addr
                 685 ..c -rwxr-xr-x 503      503      /var/ftp/nerod/sshd/sshd_config
              649827 ..c -rwxr-xr-x 503      503      /var/ftp/nerod/sshd/sshd
                1250 ..c -rwxr-xr-x 503      503      /var/ftp/nerod/clean
                 531 ..c -rwxr-xr-x 503      503      /var/ftp/nerod/sshd/ssh_host_key
                1024 ..c drwxr-xr-x 503      503      /var/ftp/nerod/sshd
                1091 ..c -rwxr-xr-x 503      503      /var/ftp/nerod/sshd/sshd-install
                 636 ..c -rwxr-xr-x 503      503      /var/ftp/nerod/functions
                2210 ..c -rwxr-xr-x 503      503      /var/ftp/nerod/sysinfo
                2960 ..c -rwxr-xr-x 503      503      /var/ftp/nerod/shad
                4060 ..c -rwxr-xr-x 503      503      /var/ftp/nerod/sense
                 554 ..c -rwxr-xr-x 503      503      /var/ftp/nerod/.1proc
                 319 ..c -rwxr-xr-x 503      503      /var/ftp/nerod/.1file
                6100 ..c -rwxr-xr-x 503      503      /var/ftp/nerod/wp
                 335 ..c -rwxr-xr-x 503      503      /var/ftp/nerod/.1logz
               15675 ..c -rwxr-xr-x 503      503      /var/ftp/nerod/install
                5888 ..c -rwxr-xr-x 503      503      /var/ftp/nerod/linsniffer
                 880 ..c -rwxr-xr-x 503      503      /var/ftp/nerod/sshd/ssh_config
Jul 23 04 09:25:02    49148 ..c -rwxr-xr-x root     root     /sbin/ifconfig
```

En la última línea subrayada se muestra cómo el archivo comprimido “nerod.tar.gz” traía consigo también el sniffer linsniffer que se utilizó para capturar información de forma

periódica y ser enviada posteriormente a un correo que suponemos pertenece al intruso. La mayor parte de los archivos mostrados en el fragmento anterior de la salida del comando “mactime” pertenecen al directorio /var/ftp/nerod creado por el intruso. Aquí se detallan todos los comandos que se van a instalar en lugar de los que tiene el sistema, es decir, se van a instalar los comandos troyanizados por el usuario.

En el fragmento siguiente se muestra la salida de un script el cual combina el comando “ils” y el comando “icat”, dando como salida los archivos borrados dentro de la partición raíz del sistema comprometido. Los archivos se crean de acuerdo al número de inodo al cual podría corresponder el archivo, es decir, al que se encontraba ligado originalmente antes de ser borrado del sistema.

 # ls
13 16187 20087 22313 28264 28280 28282 30136 34145 34343 **38170** 46221 6083 6085
14100 16200 22312 28121 28265 28281 28283 32142 34328 38165 38171 46222 6084 6086
/tmp/borrados #

En el siguiente fragmento mostramos el contenido de uno de los archivos recuperados: el archivo **38170**, donde se observa información relacionada con el sistema. Esta información sería enviada al intruso vía correo electrónico de forma periódica una vez que se colocó un cron dentro del sistema, como se muestra a continuación:

 # more 38170

Network info:

Hostname : localhost.localdomain (xxx.xxx.xxx.xxx.xxx)
Alternative IP : 127.0.0.1
Host : localhost.localdomain
Distro: Red Hat Linux release 7.1 (Seawolf)
Uname -a
Linux localhost.localdomain 2.4.2-2 #1 Sun Apr 8 19:37:14 EDT 2001 i586 unknown
Uptime
12:25am up 1 day, 5:24, 0 users, load average: 0.60, 0.16, 0.05
Pwd
/var/ftp/nerod
ID

uid=0(root) gid=0(root) groups=50(ftp)

Yahoo.com ping:

PING 216.115.108.243 (216.115.108.243) from 192.168.2.7 : 56(84) bytes of data.

--- 216.115.108.243 ping statistics ---
6 packets transmitted, 0 packets received, 100% packet loss

Hw info:

CPU Speed: 166.196MHz
CPU Vendor: vendor_id : GenuineIntel
CPU Model: model name : Pentium 75 - 200
RAM: 48 Mb
HDD(s):

Filesystem	Type	Size	Used	Avail	Use%	Mounted on
/dev/hda8	ext2	251M	52M	185M	22%	/
/dev/hda1	ext2	53M	3.4M	47M	7%	/boot
/dev/hda6	ext2	1.5G	44k	1.4G	1%	/home
/dev/hda5	ext2	1.5G	479M	1023M	32%	/usr
/dev/hda7	ext2	251M	20M	218M	9%	/var

5.3.5 Análisis de Resultados y Reporte Final

El usuario particularmente está interesado en saber de donde provino el ataque. En este análisis, trataremos de reconstruir los hechos, elaborar un perfil del atacante y sus propósitos.

La máquina tiene instalado el sistema operativo Red Hat 7.1 con el número IP interno xxx.xxx.xxx.xxx y fue atacada repetidas veces siendo el último ataque efectivo el día 23 de Julio del 2004 a las 09:22:47 horas, explotando la vulnerabilidad en el servidor wu-ftpd 2.6.1-16.

A fin de obtener el acceso no autorizado al servidor, el atacante ha utilizado una serie de herramientas de terceros distribuidas. El kit de herramientas del atacante incluía un auto-rooter llamado autowu modificado. El objetivo principal de esta herramienta es

localizar de forma automática servidores vulnerables wu-ftpd (<= 2.6.1), explotarlos y propagarse, utilizando cada máquina comprometida como base de operaciones.

Finalmente, la máquina fue vulnerada desde el siguiente número IP xxx.xxx.xxx.xxx a las 09:22:47 horas del 23 de Julio del 2004. A las 09:24:19 horas se trasfería al servidor el archivo /var/ftp/nerod.tar.gz. A las 09:24:45 horas el archivo se descomprimía y posteriormente a las 09:25:02 se iniciaba el proceso de instalación del rootkit incluido en nerod.tar.gz.

El archivo de instalación del rootkit /var/ftp/nerod/install es un script de shell que a las 09:25:04 deshabilita el servicio de registro syslogd/klogd con la señal 15, sustituye algunos binarios legítimos del sistema, suplantándolos por sus versiones modificadas, altera varios archivos de configuración de servicios de red y de arranque de servicios de la máquina.

A las 09:25:10 el script de instalación de la puerta trasera SSH crea una cuenta con el nombre de usuario ssh y UID 0. La cuenta no dispone de contraseña lo que permitirá al intruso acceder al sistema sin tener que introducir ninguna clave.

La puerta trasera SSHD así como el linsniffer se controlan a través del archivo /etc/rc.d/init.d/functions modificado durante el proceso de instalación del rootkit, lo que confirma el posterior reinicio de xinetd a las 09:25:15 horas.

Esta versión de linsniffer disfrazada del binario identd guardaba la información capturada en el archivo /usr/local/games/tcp.log. El agresor planeaba obtener las capturas de forma periódica automatizando el envío del archivo a su correo electrónico desde una entrada crontab creada para el usuario operator. Del archivo /var/spool/cron/operator localizamos la dirección de correo electrónico del posible agresor suntsfant2@yahoo.com. La entrada crontab fue creada a las 09:25:19 horas.

A las 09:27:09 horas el intruso crea con el comando adduser la cuenta de usuario nerod.

La sesión FTP creada por alguna de las utilidades del auto-rooter del agresor caduca a los 16 minutos, es decir, a las 09:38:01.

A posteriori, el agresor crea el directorio /root/.,, donde usando diferentes métodos intenta descargar un proxy para redes IRC y los propios archivos de auto-rooter - hasta las 10:16:55 lo consigue.

A las 09:56:39 el intruso decide deshabilitar el acceso anónimo al servidor para prevenir posibles ataques por la misma vía de acceso.

Finalmente, a la 10:16:55 después de varios intentos, descarga usando el programa wget el archivo PsyBNC.tar.gz de Geocities, lo descomprime y tras compilarlo lo ejecuta a las 10:17:41 horas.

Como paso seguido a las 10:41:51 se descarga con wget el archivo awu.tgz (la misma colección de herramientas para buscar servidores vulnerables wu-ftpd y explotarlos, aunque el archivo también contiene algunos exploits para ssh) del servidor público Geocities.

Se descomprime el archivo awu.tgz y éste crea el directorio /root/.,/aw/. El agresor decide lanzar un barrido de la red 191.168.2.x. Después de casi 3 minutos a las 10:46:19 decide parar la búsqueda, limpiar la pantalla de la terminal con el comando /usr/bin/clear y pasar a inactividad.

A las 11:17:09 el intruso o su colaborador accede a la máquina utilizando la puerta trasera SSHD (o login), comprueba si hay otros usuarios conectados al servidor, cambia al directorio /var/tmp y crea una carpeta “.,”.

El agresor decide instalar otra aplicación relacionada con redes IRC, un IRC-BOT, que descarga usando el programa wget desde el servidor público Geocities.

El archivo descargado /var/tmp/.,/manu.tgz contiene la versión 2.8 de EnergyMech. El intruso usa el editor de texto pico para configurar el IRC-BOT, añadiendo el nick de George que llegó a sustituir el nick de manu. El IRC-BOT.

Durante el proceso del análisis se sospecha que la versión de EnergyMech 2.8 incluye una puerta trasera o virus ya que en el directorio /dev a la hora de ejecutar el IRC-BOT aparecen los archivos /dev/hdx2 y /dev/hdx1 y los binarios como /bin/ping muestran tener sus registros MAC-time alterados. Se sospecha que puede tratarse de algún tipo de programa para sistemas Linux que puede estar modificando las versiones actuales de archivos ELF.

A través de la información incluida en el aviso detectamos con mayor facilidad que los siguientes binarios están infectados: chgrp, chmod, chown, cp, cpio, dd, df, dnsdomainname, domainname, de, hostname, ln, ls, mail, mkdir, mknod, mktemp, mt, mv, netstat, nisdomainname, ping, red, setserial y ypdomainname.

A las 11:20:29 el IRC-BOT se está ejecutando correctamente bajo el pid 7989, finalmente el intruso borra el archivo /var/tmp/.,/manu.tgz.

A las 15:36:30 el servidor se deshabilita por el administrador.

5.4 Caso Práctico 3

5.4.1 Preparación para el Análisis

Para este caso práctico nosotros configuramos un ambiente completamente controlado por medio de una Honeypot de investigación, por lo que fue nuestra labor administrarlo y vigilarlo, por este motivo no contamos con un administrador de sistema y tampoco se tuvo que aplicar el cuestionario que usamos en las prácticas anteriores.

Generamos nuestros propios binarios estáticos, que ocupan más capacidad de almacenamiento ya que en ellos se encuentra incluida todas las librerías que necesitan para su ejecución, es decir, no se utiliza en lo más mínimo algún recursos del sistema, esto se hizo con el fin de no utilizar los comando de Linux de la máquina local que pudieran estar troyanizados. En este CD se podrá encontrar también sniffers y herramientas de IDS. También se agregaron las herramientas forenses adecuadas para ser utilizadas en cualquier instante.

Esta práctica se realizó gracias al apoyo de la Unidad de Servicios de Cómputo Académico de la Facultad de Ingeniería facilitando un nodo y el equipo de cómputo necesario para realizar las pruebas pertinentes.

5.4.2 Configuración del sistema de Monitoreo

Se instaló y configuró un sistema de monitoreo el cual nos permite obtener la mayor cantidad de registros posibles de la honeypot que se configurará. En el sistema de monitoreo se instaló Linux Suse 9.0 Profesional. Para la parte de configuración se instalaron primero herramientas de seguridad propias para el sistema. Estas herramientas nos permitieron tener controladas las conexiones, bloquear otros sistemas, verificar la integridad del equipo, establecer conexiones cifradas; en este sistema únicamente se tendrá habilitado el servicio de Secure Shell para conexiones cifradas y el servicio de Syslog para la recepción de bitácoras remotas, por tanto, todo el sistema quedará filtrado por un Firewall interno que será configurado.

Para el sistema de monitoreo, se configuró el servicio de syslog con la finalidad de poder aceptar los paquetes enviados de la honeypot, es decir, todos los registros generados por la honeypot serán enviados al instante al sistema de monitoreo. Esto nos ayuda a tener respaldados los registros de la honeypot por si el intruso llegase a borrar o alterar las bitácoras de la misma.

Posteriormente se configuró un keylogger servidor para poder recibir todo lo tecleado por el usuario dentro de la honeypot. Esto es muy importante para poder dar un seguimiento paso a paso de lo que el usuario fue haciendo dentro del sistema una vez que lo comprometió, informándonos de los archivos borrados, de los directorios creados, de los programas que ha descargado dentro de la honeypot para poder tener un control completo del mismo.

Por último se configuró un IDS para poder conocer todo el tráfico de red que existe hacia nuestra honeypot y nos de alertas de patrones de paquetes que se han enviado hacia ella.

5.4.3 Apertura de los servicios disponibles

Aquí se configuró la honeypot instalando el sistema operativo Linux Red Hat 7.1 sin actualizaciones. En ella se modifica el servicio syslog para que pueda enviar todos sus registros al sistema de monitoreo. También se descarga un kernel (núcleo del sistema) nuevo para que se pueda configurar el keylogger cliente, es decir, para que pueda enviar todo lo que se teclee al sistema de monitoreo. Una vez que se descargó el kernel se procede a compilarlo e instalarlo dentro del sistema Red Hat. Posteriormente se configura y se instala el keylogger.

Los servicios que estarán activos en la honeypot son:

- ssh
- sendmail
- web
- X11
- ftp
- portmap
- nfs
- samba
- syslog

Todos ellos estarán aceptando conexiones de cualquier lugar de la red local sin ningún firewall de por medio, esto hace que cualquier máquina del Laboratorio de Redes pueda ser el intruso que comprometa a nuestra honeypot. El sistema una vez instalado y configurado se le entregará a un administrador del Laboratorio de Redes para que genere a los usuarios, revise bitácoras y verifique el buen funcionamiento del sistema.

5.4.4 Monitoreo de los sistemas

Una vez que se instaló el sistema de monitoreo y se configuró la Honeypot, procedimos a inicializar ambos sistemas conjuntamente para dar inicio a la captura de datos. Todo paquete que llegue directo al sistema de monitoreo o a la Honeypot se considerará tráfico hostil debido a que a dichos sistemas no deben aparecer ningún

intento de conexión. Por tanto, todo paquete será considerado como un posible ataque a la Honeypot. En este caso únicamente estarán recibiendo información el IDS y el servicio de syslog, el keylogger permanecerá en espera de que el intruso ya se encuentre en el sistema para comenzar a capturar todo lo tecleado por él.

5.4.5 Congelación de la escena del crimen

Todo el sistema en conjunto se terminó de configurar el día Viernes 13 de Agosto del 2004, fue comprometido el día Sábado 14 de Agosto del 2004, teniendo un alto grado de efectividad considerando que los sistemas Red Hat 7.x son vulnerados en un lapso no mayor a dos semanas. La Honeypot continuó en producción hasta el día Lunes 16 de Agosto del 2004 permitiendo al intruso instalarse por completo en la misma, dándonos oportunidad de observar su comportamiento una vez que se perpetró la intrusión en el equipo.

Nosotros determinamos dar por finalizado la práctica el día Lunes 16 de Agosto debido a que creímos que era suficiente la información recabada en el sistema de monitoreo y lo realizado por el intruso. No podíamos permitir que desde nuestra Honeypot se perpetraran ataques hacia otros sistemas. Antes de dar por finalizada la práctica revisamos los procesos, conexiones y procedimos a respaldar la memoria pero hubo un error en conexión por lo que decidimos reiniciar la tarjeta de red, esta situación fue un error debido a que esto ocasionó que el sistema se bloqueara impidiendo la recolección de la memoria. La forma de revisar las conexiones, procesos y recolección de la información localizada en memoria fue con base en nuestra metodología planteada. El error radicó en haber reiniciado la tarjeta de red, olvidando las posibles consecuencias.

5.4.6 Almacenamiento de pruebas

Al estar el sistema muerto procedimos a esterilizar el medio en el cual se iban a colocar las imágenes del disco duro víctima. Posteriormente se obtuvieron las imágenes de las particiones del disco duro afectado. Volvimos a utilizar la firma digital por MD5 para comparar todas las firmas tanto de las particiones del disco duro como de las imágenes obtenidas. Por último montamos las imágenes sobre el sistema para poder recuperar la información y analizarla posteriormente.

5.4.7 Análisis con herramientas estándares de UNIX

Iniciamos el análisis determinando el runlevel 5 en el cual el sistema se encontraba. La salida de uno de los comandos ejecutados cuando el sistema aún estaba arriba fue de todos los archivos abiertos en el sistema. En esa parte encontramos que se andaba ejecutando un programa con el nombre de emech-2.8 como se puede observar en la figura 5.9 en la parte subrayada.

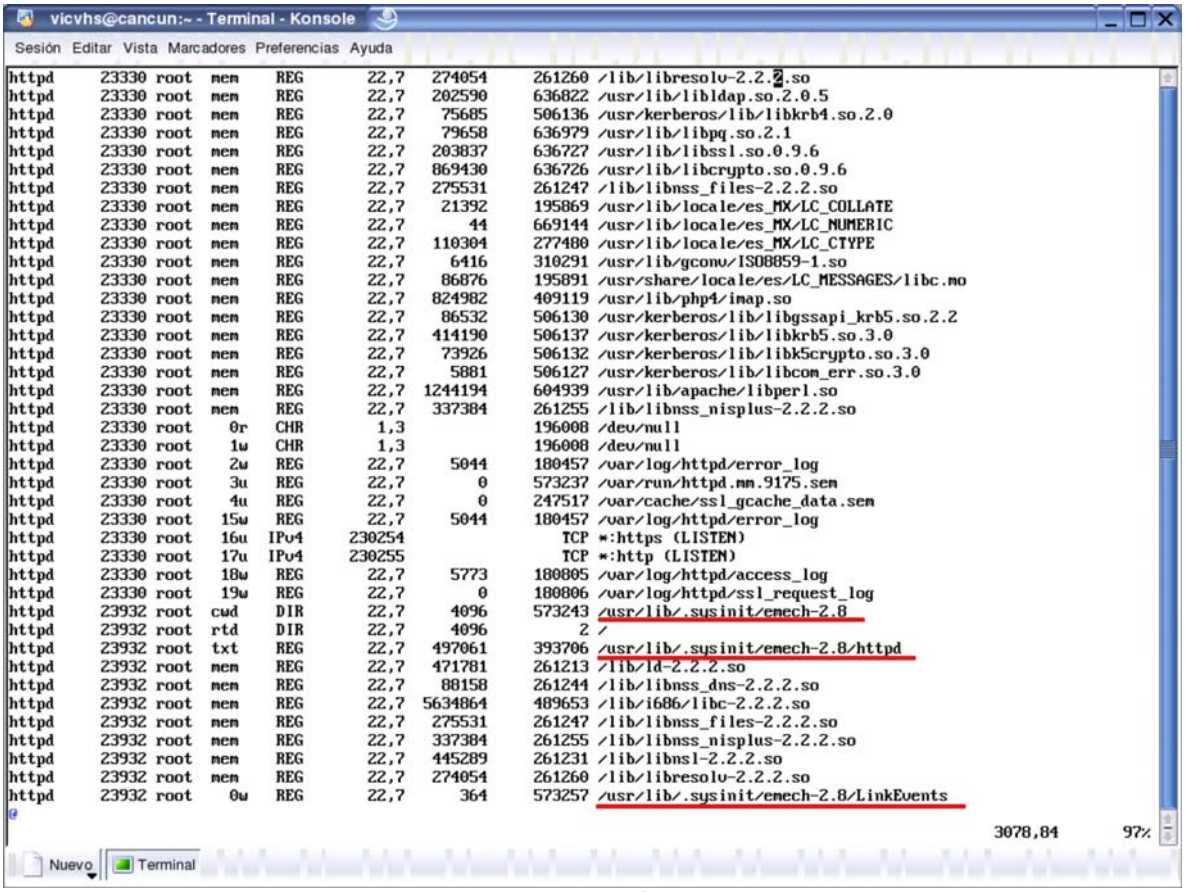


Fig 5.9 Lsof práctica 3

Por parte de las conexiones de red, nos percatamos que se encontraban abiertos los puertos 1245 y 1267 estableciendo conexión con otras IP's. Esto se puede observar en la figura 5.10

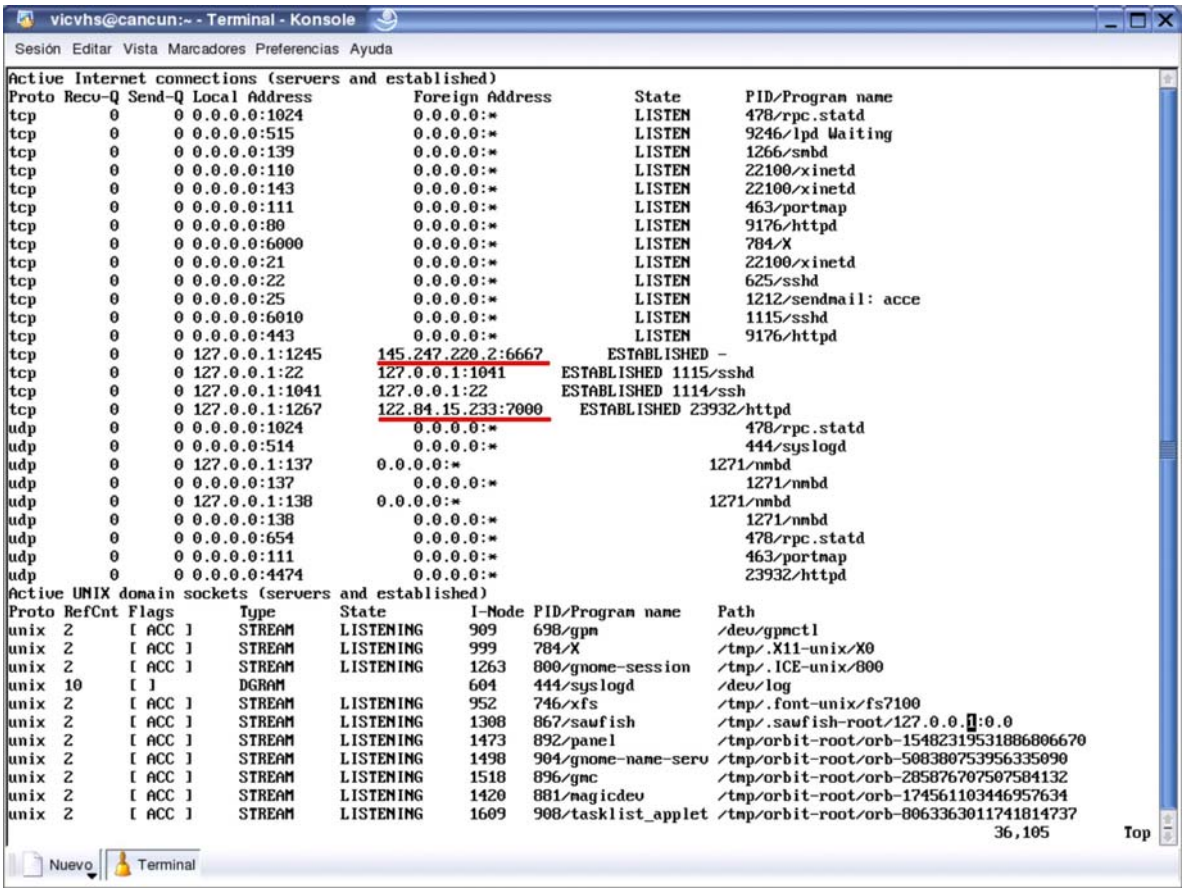


Fig 5.10 Estado de red práctica 3

Se detectó que el usuario “root” no tenía .bash_history siendo éste borrado por el intruso. Dentro del archivo .ssh/known_hosts2 encontramos que se guardó la llave rsa de una máquina a la cual se estableció una conexión por medio del secure shell, cosa que hizo el intruso.

Dentro del directorio /tmp se encontró un archivo tar.gz con el nombre de “so.tgz”. Este archivo contenía el directorio super que estaba bajo el directorio /tmp. También encontramos el directorio creado “/usr/share/locale/sk” el cual tenía la salida de un sniffer colocado por el intruso en el sistema. Dentro de ese mismo directorio se encontró otro directorio, el “/usr/share/locale/sk/.sk12/psybnc” que es un IRC. En el directorio “/usr/lib/.sysinit” oculto creado por el atacante se encontró el archivo tar.gz “emech-2.8.tar.gz” el cual fue descomprimido por el intruso generando el directorio “/usr/lib/.sysinit/emech-2.8”. Este directorio contiene un proxy IRC para el anonimato de las conexiones por este medio.

Detectamos la creación de un usuario con el nombre “cmi” el cual tiene el uid y gid igual a cero, es decir, entra con privilegios de administrador directamente al loguearse en el sistema. Dentro del directorio /root se crea el directorio /root/.mc el cual contiene únicamente tres archivos con los nombres: Tree, history e ini. Por último se encontramos la creación de un archivo con el nombre “/sbin/initsk12”. Este archivo probablemente venga de un rootkit.

Nos dimos cuenta que el comando insmod fue utilizado, por lo que muy posiblemente fue agregado un módulo al kernel para poder ocultar procesos o conexiones. Respecto a las bitácoras generadas por el sistema, estas no fueron alteradas por el intruso, ya que coincidían con las que se almacenaron en el sistema de monitoreo. Dentro de ellas se observaron conexiones de la IP 127.0.0.10 el día 14 de Agosto del 2004 a las 23:38. Estas conexiones fueron constantes durante un periodo de 10 minutos vía ftp; desde otra IP 127.0.0.11 se encontraron registros de intentos de conexión vía secure shell a las 23:56 del mismo día. En esta ocasión fueron intentos de la misma dirección IP pero con distintos usuarios. Esto es indicio de que ejecutaron un programa con un diccionario para poder acceder al sistema.

En la bitácora /var/log/messages el día 15 de Agosto del 2004 a las 19:44 el sistema registro que la tarjeta de red se encontraba en modo promiscua. Esto nos indica que dentro del sistema se ejecutó un sniffer para capturar los paquetes que viajaban en la red. En la misma bitácora, encontramos que el kernel mandó un aviso el día 16 de Agosto del 2004 a las 04:02 indicando que existía un error con el archivo “tty_io.c”. Esto más que nada se debe a la interacción del kernel con algunos módulos. Por último se encontró que el intruso agregó un usuario al sistema el día 16 de Agosto del 2004 a las 08:06 con el nombre de “cmi” teniendo como uid y gid cero.

5.4.8 Análisis con herramientas forenses

Comenzamos ejecutando un chkrootkit para detectar si el intruso instalo un rootkit en el sistema y determinar qué hace dicho rootkit. Al realizar la ejecución del chkrootkit se encontró que el usuario instaló el rootkit Suckit el cual instala un sniffer, un ocultador de procesos, un ocultador de archivos, y una puerta trasera de entrada a nivel de núcleo. En

la figura 5.11 se observa en la línea subrayada como la herramienta chkrootkit ha encontrado la alteración del archivo /sbin/init.

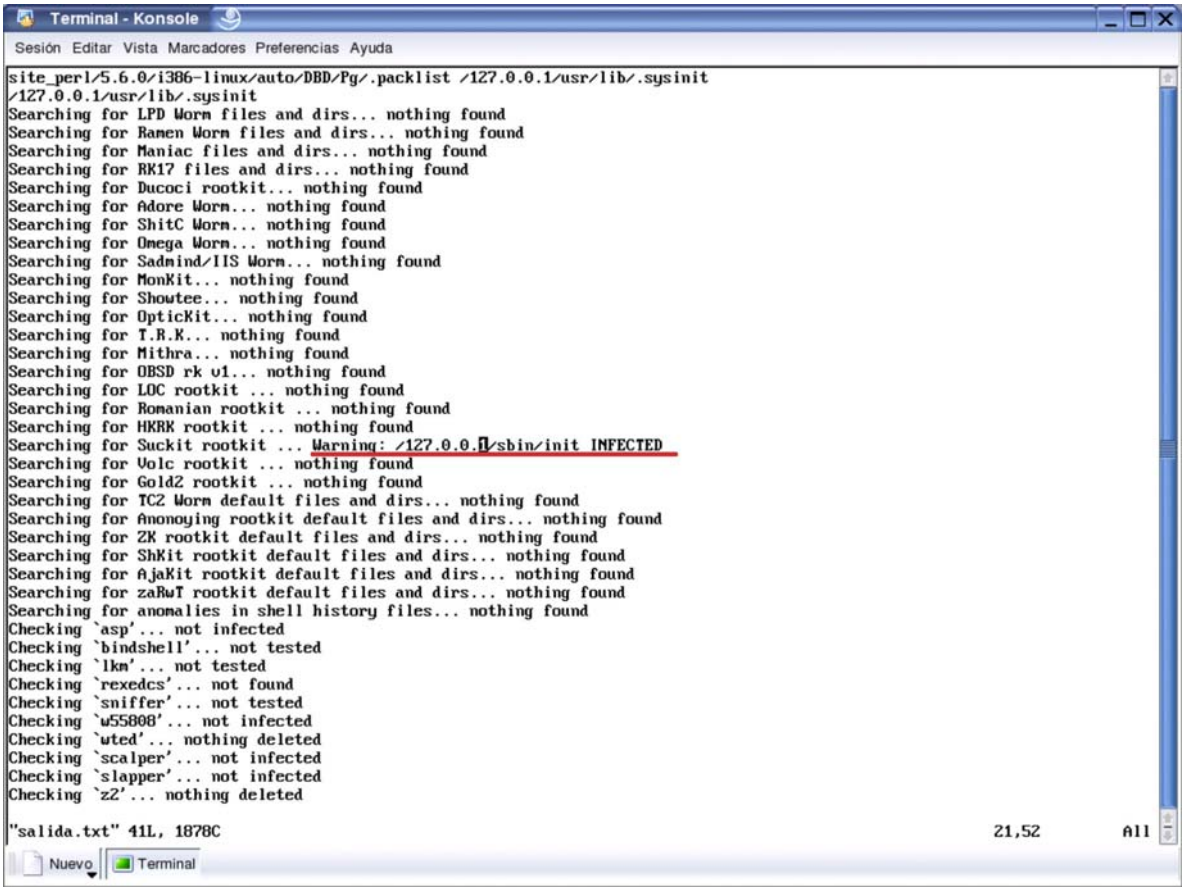


Fig 5.11 Chkrootkit práctica 3

Al ejecutar la herramienta Sleuhtkit observamos como el intruso borró archivos al momento de estar dentro de la honeypot. Uno de estos archivos fue descargado por él mismo. En la figura 5.12 se subrayaron dos de los archivos borrados: “/usr/share/locale/sk/.sk12/q.tgz” y “/usr/share/locale/sk/.sk12/ssh”. Dentro de la misma figura, vemos como no solamente fueron estos dos archivos eliminados sino también durante la instalación y ejecución del IRC se eliminaron más archivos que son mostrados en la figura 5.12

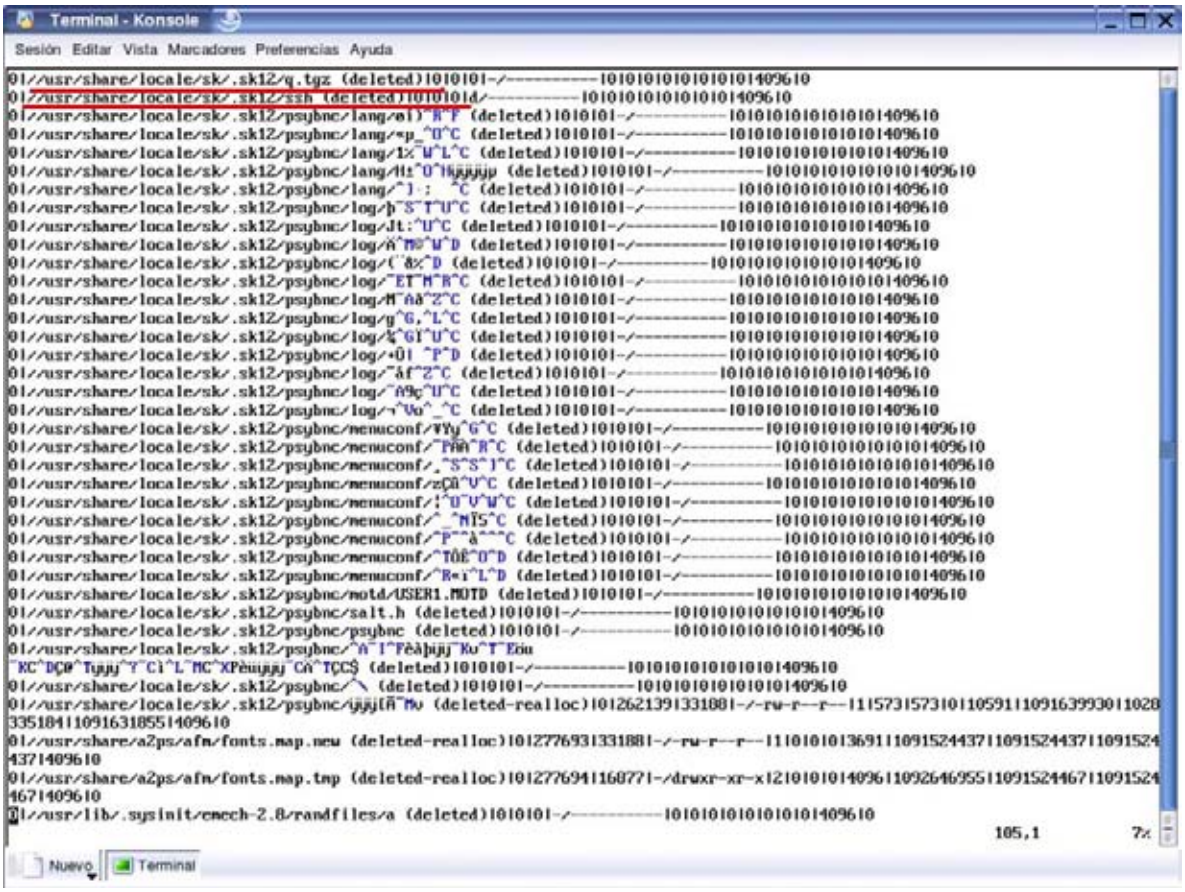


Fig 5.12 Sleuhtkit práctica 3

Obteniendo el mactime de todos los archivos, encontramos la hora exacta en la cual se efectuó la intrusión dentro de la honeypot. En la figura 5.13 en el primer subrayado de arriba hacia abajo, vemos el archivo que se creo a las 23:46:52 horas del día 14 de Agosto del 2004 siendo éste el momento en que el intruso accedió al sistema. Posteriormente se observa en la misma figura, en el subrayado segundo subrayado de arriba hacia abajo, como el intruso intenta apoderarse de inmediato del sistema descargando el archivo "/tmp/so.tgz" a las 23:48:02 del mismo día. Una vez que se descargó el archivo tar.gz, se dispuso a destarearlo y ejecutar el programa "/tmp/super/init" a las 23:48:10. Esto último se observa en la figura 5.13 en el tercer subrayado de arriba hacia abajo. De igual forma se ve que son varios los archivos que contiene "so.tgz", estos se encuentran dentro del directorio "/tmp/super". Todo lo que ha hecho el intruso se efectuó en un lapso no mayor de dos minutos, por lo que se ve, es un intruso que ya tiene un mecanismo implementado, es decir, cada vez que accede a un sistema vulnerado, ingresa al sistema, descarga programas que son un rootkit, sniffer,

puertas traseras para asegurarse un regreso al sistema satisfactorio y posteriormente, la instalación de canales de chat para interactuar con otros intrusos en la red.

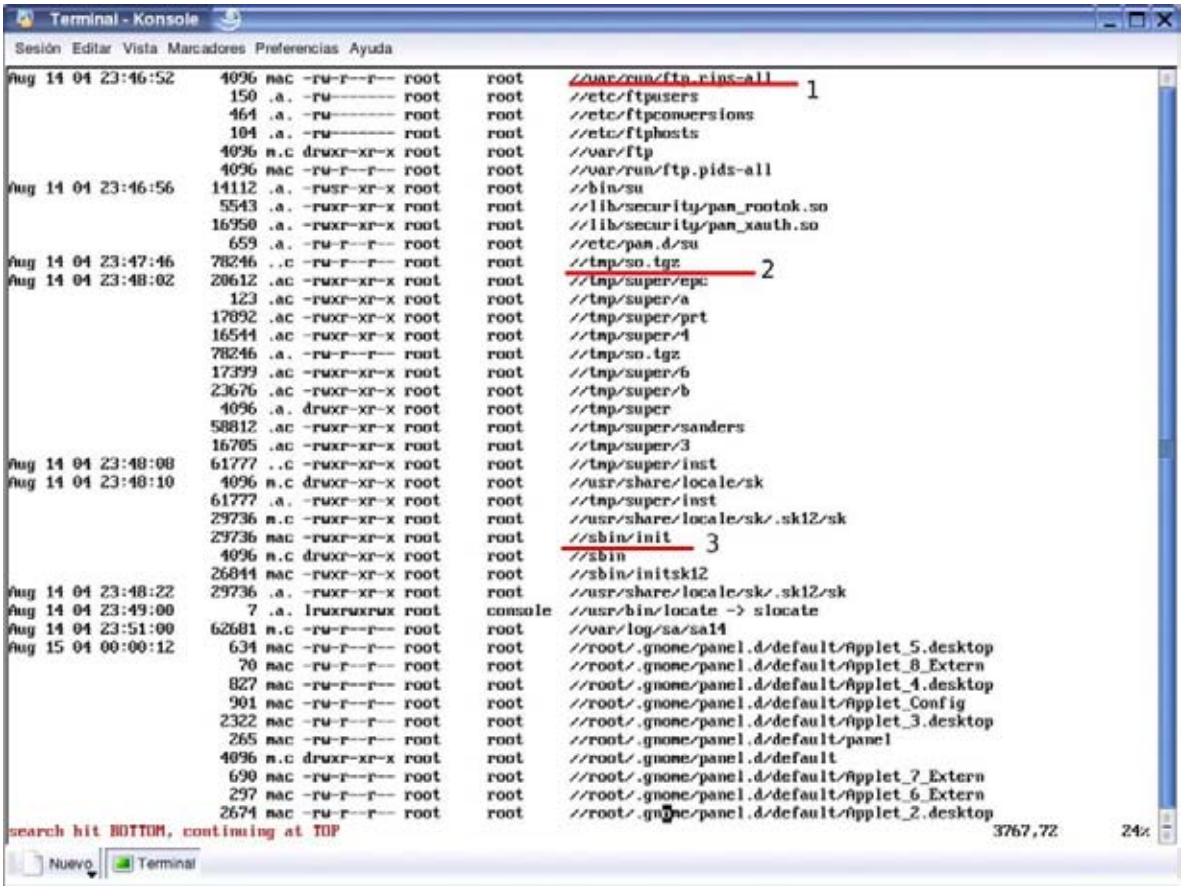


Fig 5.13 Tiempos mac-1 práctica 3

El archivo "/var/run/ftp-rips.all" que se encontró, fue analizado por nosotros descubriendo la dirección IP por la cual el intruso efectuó el ataque y el acceso a la Honeypot. A continuación mostramos el contenido de este archivo:

 # hexdump ftp.rips-all

0000000 0000 0000 0000 0000 0000 0000 0000 0000

*

0000ff0 0000 0000 0000 0000 0000 0000 e834 78d1

0001000

Para poder descifrar dicho contenido se utilizó el comando hexdump siendo el resultado de esto un archivo con números en hexadecimal, posteriormente tuvimos que

convertir los números hexadecimales a números decimales obteniéndose la IP xxx.xxx.xxx.xxx. Esta IP es la que utilizó el intruso para acceder a la Honeypot.

Al analizar lo capturado por el keylogger se muestra como el intruso va tecleando comandos dentro de la Honeypot lo cual permite observar a detalle los movimientos que realiza. Esta acción facilita la parte de seguimiento paso a paso para poder responder a la pregunta qué hizo en el sistema comprometido. En la figura 5.14 se muestra un pequeño fragmento de lo capturado por el keylogger, al momento de estar el intruso dentro del sistema, lo mostrado es un poco de los comandos que ejecutó esta persona. Por cuestiones obvias de seguridad no se muestra la IP de la Honeypot ni tampoco la IP del sistema de monitoreo. Como se observa en el recuadro, vemos que el UID es igual a cero, éste corresponde al uid que utiliza el administrador del sistema siendo ahora dueño el intruso. Además, en la parte subrayada, vemos como el intruso descarga de inmediato, haciendo uso del comando wget, sus programas para apoderarse por completo del sistema. De esta forma nosotros pudimos observar todos los movimientos realizados por el atacante sin perder detalle alguno de lo que hizo dentro de la honeypot.

Fig 5.14 Keylogger práctica 3

5.4.9 Análisis de resultados y reporte final

Para esta última práctica, buscamos determinar la forma en que el intruso pudo acceder al sistema y dar seguimiento a cada uno de los pasos que el atacante realizó dentro de la honeypot configurada por nosotros.

La honeypot tenía instalado el sistema operativo Red Hat 7.1 con el número de ip xxx.xxx.xxx.xxx sin que el sistema se le haya aplicado parche alguno. El sistema fue instalado el día 13 de agosto de 2004 contando con los servicios de: ssh, sendmail, web, X11, ftp, portmap, nfs, samba, syslog. El sistema Red Hat 7.1 se ejecutaba en el runlevel 5 con un kernel 2.4.20. La actividad por parte del intruso que ingreso al sistema inicio desde el día 14 de agosto 2004 a las 23:38 horas vía ftp desde la ip xxx.xxx.xxx.xxx. Siendo el sistema comprometido a las 23:46:52. Cabe hacer notar que no únicamente fue esa IP la que se registró en nuestro sistema de monitoreo, sino también las otros sistemas que intentaron ingresar sin tener éxito.

Una vez que el intruso tuvo éxito para ingresar a la honeypot se dispuso a descargar herramientas para obtener un control total sobre el sistema. Primero el verificó que no existiera ningún sniffer en la honeypot haciendo una búsqueda de ello, posteriormente como no encontró ninguno, se dispuso a descargar el archivo “so.tgz” a las 23:47:46. “destareo” el archivo descargado y se creo el directorio “super” a las 23:48:02 el cual contiene sus herramientas. Se movió dentro del directorio y ejecutó el script “inst” para poder instalar el rootkit “Suckit” en la honeypot. Una vez hecho esto ejecuta el sniffer y adicionalmente descarga otro archivo tar “w00t.tgz” el cual contiene un escaneador de puertos con un exploit para el servicio de “samba”.

Ya instalado el intruso, ocultando su presencia realiza una conexión a otro sistema con la ip xxx.xxx.xxx.xxx a las 00:25:42 del día 15 de agosto 2004. Posteriormente se descarga la herramienta “john the ripper” a las 00:30:26 del día 15 de agosto 2004 para crackear passwords dentro de la misma honeypot. Destareó el paquete y posteriormente lo ejecutó sobre el “/etc/shadow”.

A las 00:45:54 del día 15 de agosto 2004 descargó el archivo “psyBNC2.3.2-4.tar.gz” que es un proxy de IRC. Lo “destareo”, lo compiló y lo instaló dentro del sistema.

Una vez instalado el IRC lo dejó ejecutándose en el equipo. Posteriormente, el intruso dejó ejecutándose estos programas, por lo cual sólo hubo la modificación de archivos de registro durante el resto del día 15 de agosto 2004. El atacante regresó el día 16 de agosto 2004 a las 06:55:20 siendo esta conexión de entrada y salida únicamente. Nuevamente el intruso accedió a la honeypot a las 08:38:27 para agregar un usuario al sistema con el nombre de “cmi” teniendo por uid y gid igual a cero. Este usuario tenía como shell /bin/bash y como directorio personal “/home/cmi”. El intruso se logueó desde otra maquina con ese loguin para verificar que si podía entrar a la honeypot con esa cuenta con privilegios de administrador.

Por último, descargó el paquete “emech-2.8.tar.gz” el día 16 de agosto 2004 a las 08:09:44 que es un cliente IRC para conectarse a canales de chat, lo “destareo” y posteriormente lo compiló e instaló dentro del directorio “/usr/lib/.sysinit/” que es un directorio oculto creado por él mismo dentro del sistema.

Nosotros decidimos dar de baja el equipo a las 12:21 horas del día 16 de agosto 2004.

5.5 Discusión de los casos prácticos

Los resultados obtenidos en los casos prácticos corroboran la necesidad de realizar un análisis de la situación actual del incidente, antes de tomar alguna acción en la investigación, para poder capturar la mayor cantidad de evidencia tratando de alterarla lo menos posible y limitar los daños causados por la intrusión.

El proceso de captura de evidencia es sencilla cuando el sistema se encuentra apagado, pero cuando el sistema se encuentra en producción se complica la captura debido a la alteración que tanto el intruso como nosotros hacemos sobre el sistema, perdiendo calidad en la captura de información.

La selección de las herramientas forenses fue adecuada porque con ellas pudimos reconstruir de manera considerable los eventos ocurridos en el incidente, lo cual impactó en el reporte final de forma positiva.

CONCLUSIONES

- **Contribuciones del análisis forense a la seguridad informática**

Actualmente las intrusiones a sistemas de cómputo se están haciendo muy comunes y demasiado frecuentes, cuando analizamos el problema de la seguridad informática, vimos que el número de incidentes de seguridad con respecto al tiempo esta creciendo de manera exponencial, con un crecimiento cuatro veces más grande en el año 2000 a la fecha del que se ha registrado antes de dicho año y continúa creciendo. Un estudio realizado sobre un número significativo de sitios en Internet de diferentes sectores de la vida pública nacional nos revela que más de la mitad de ellos presentan vulnerabilidades de alto riesgo.

Conforme pasa el tiempo los atacantes y sus herramientas se hacen cada vez más peligrosas, complejas y sofisticadas así como también han incrementado en un gran número.

Por estas y otras razones el análisis forense es uno de los campos de la seguridad en cómputo que esta teniendo un gran auge ya que un sistema puede ser comprometido con relativa facilidad, dejando en su gran mayoría de los casos, a un lado las causas, las vulnerabilidades, los motivos de un ataque para enfocarse solo en la inmediata restauración del sistema atacado, sin ni siquiera hacer una mejoría de seguridad, ni la más remota idea de que fue lo que sucedió.

El análisis forense, se enfoca entre otras cosas en la detección de intrusos así como en el análisis de sistemas que se han sido comprometidos; ofreciendo protección de los datos y evidencias de los equipos afectados, por otra parte al conocer los tipos de ataques, los analistas forenses están en disposición de dar las medidas preventivas necesarias para evitar futuros incidentes en nuestros equipos.

El análisis forense no solo dará una respuesta a los incidentes de seguridad, si no que también fomentará la cultura de la seguridad de la información, mostrará las deficiencias en seguridad así como puntos débiles en nuestra institución, para que a futuro se maneje de una manera más experimentada cualquier incidente de seguridad, se tengan una mejor planeación de respuesta a incidentes así como también se corrijan y actualicen las políticas de seguridad, así como nos muestra el punto de vista de un intruso a nuestro sistemas.

Para finalizar esta parte, la investigación forense ha tratado de establecer una metodología formal para el análisis de incidentes de seguridad, como lo son la recolección de evidencias, las cadenas de custodia, etc. Para que de esta manera se pueda definir, el tamaño del daño, responder a las preguntas principales ¿Cuándo?, ¿Dónde?, ¿Cómo?, hacer una reconstrucción cronológica de los hechos y finalmente manejar la colección de las evidencias para poder demostrar los mismos. Todo con un enfoque metódico y basada en acciones premeditadas.

Esta es una de las aportaciones más importantes y, en particular, es una de las aportaciones que tiene el presente trabajo de tesis para la seguridad en cómputo.

- **El análisis forense y su aportación contra intrusos**

Las herramientas tecnológicas para el análisis forense son aplicaciones que tienen un papel muy importante para reunir pruebas e información necesaria.

Un IDS está continuamente monitorizando al sistema informático, pero con el fin de detectar algún intento de ataque, él le proveerá información de las tácticas y herramientas de los intrusos al análisis forense, por ejemplo: la forma en cómo llevan un ataque, cómo ocultan los datos, qué herramientas frecuentemente utilizan y finalmente detectar los pasos más comunes de un ataque, como lo son: reconocimiento, explotación, operaciones ocultas.

De esta manera se podrán optimizar y mejorar los IDS en su labor de detectar algún intento de ataque, ya que con esta información se mejora el análisis de patrones anómalos reduciendo el número de falsos positivos porque se mejoran las reglas que contienen los datos y patrones para detectar anomalías de seguridad en el sistema, también se mejorarán los filtros que comparan los datos espiados de la red o de logs con los patrones almacenados en las reglas, los detectores de eventos anormales en el tráfico de red, se podrían mejorar la tolerancia a fallos o capacidad de respuestas a situaciones inesperadas, el dispositivo generador de informes y alarmas.

Se puede optimizar la base de conocimientos de un IDS basado en un sistema experto o se puede también profundizar, actualizar e incrementar la base de datos de ataques que almacenan algunos IDS's.

Un IDS podría saber cuándo el intruso está usando un método nuevo, gracias a la información que proporciona el analista forense al inspeccionar un equipo comprometido. También cabe destacar que un IDS puede recibir nuevas reglas de los propios sniffers que se tengan instalados debido a la captura de paquetes y el análisis de los mismos.

Uno de los objetivos del análisis forense es hacer que la organización aprenda de sus errores en la seguridad informática, mejorando sus dispositivos, prevenirse de futuros ataques, mejorar sus políticas seguridad, etc. Un gran aporte que hace el análisis forense es en el mejoramiento de los IDS, ya que, como se ha mencionado anteriormente, dichos sistemas pueden basarse de la experiencia que almacenan los analistas forenses de hechos reales para prevenir los posibles ataques.

- **Limitaciones del análisis forense**

Dentro de las limitaciones para el analista forense son entre otras los privilegios con los que no dispone para analizar determinada información restringida para ciertos tipos de usuarios, pero esto no es todo ya que las implicaciones éticas y legales de no respetar las políticas de seguridad de la organización, pueden ser muy serias y complejas.

Se debe tener en cuenta que nunca bajo ninguna circunstancia debe usarse la información surgida de la investigación para beneficio personal, esto reflejaría un comportamiento no ético, que estaría en contra del orden y bien moral que debe regir en la actitud de cada individuo, esto tiene un impacto social muy fuerte dentro de la tecnología de la informática, es ir en contra de cualquier formulación de políticas para el uso de la información impuestas por las empresas, la ética informática es parte de la formación profesional que representa un activo muy valioso para las organizaciones, estableciendo la base de la confianza y la seguridad en todos los aspectos laborales, además es el pilar principal que nos da valor como seres humanos. Además que las políticas de la Organización comprometida pueden realizar actividades que eliminen parte de la información del incidente.

El manejo de información es extremadamente volátil por lo que se debe de contar con el equipo necesario, ya que sin éste se puede estropear la investigación.

Los analistas forenses necesitan adquirir una serie de conocimientos y habilidades que por lo general son muy complicadas así como franquear limitaciones como pueden ser: la inexistencia de software especializado, dificultad para encontrar información

relevante, existe una gran dificultad para poder contactar a las personas involucradas sobre todo en una organización con grados y jerarquías, entre otras.

Todo analista forense debe tener experiencia en el trabajo a desempeñar porque cualquier paso en falso implica pérdida importante de evidencia. Un analista inexperto puede fracasar de inmediato por no saber que hacer en una situación complicada. El nerviosismo, la presión y la falta de conocimientos pueden perjudicar el análisis de un sistema comprometido entorpeciendo la labor.

Uno de los grandes problemas de los profesionales de seguridad en cómputo consiste en buscar a través de giga bytes de información para encontrar el dato necesario. Cuando un sistema es atacado y es necesario realizar un análisis forense, se tienen demasiados problemas para el analista forense porque se trata de sistemas de producción que mantienen servicios de misión crítica y arrojan gran cantidad de información a los mecanismos de registro de actividad del sistema, por eso localizar los datos exactos de cómo fue vulnerado el sistema se convierte en un reto que puede llevar mucho tiempo y puede afectar a una rápida respuesta al incidente, un honeypot es un sistema en un ambiente controlado, de esta forma se pueden analizar los ataques sin afectar los procesos crípticos de las organizaciones.

- **Análisis de resultados de la metodología**

Al desarrollar nuestros casos prácticos los principales objetivos, como se mencionaron en el capítulo II, fueron los siguientes: Encontrar señales de que el sistema fue comprometido, determinar el daño, responder a las preguntas ¿Qué?, ¿Cuándo?, ¿Dónde? y ¿Cómo?, reconstruir la cronología de los hechos, manejar y reconstruir la evidencia de tal forma que pueda servir para demostrar los hechos.

De manera más específica nos basaremos en los siguientes puntos para realizar una evaluación más minuciosa: La rapidez y el cuidado del análisis, la manera de recolectar la evidencia, la integridad de las herramientas, si se minimizó la pérdida de datos, finalmente si se registró todo.

Con base en estos objetivos evaluaremos nuestra metodología y los algoritmos definidos en el Capítulo II, considerando las limitaciones que puede llegar a tener el análisis, los recursos con los que se contaron para estas prácticas y las tecnologías

aplicadas, evaluando los dos casos más generales, es decir, cuando el sistema está vivo y cuando el sistema se encuentra muerto.

La manera de encontrar señales de que el equipo fue comprometido fue eficiente, ya que en los ataques realizados la información que se trató de ocultar pudo ser recuperada en su mayor parte, perdiéndose muy poca información, esto debido al tiempo de respuesta; al determinar el daño, encontramos que los cuestionarios y los reportes finales describen por completo y de forma resumida lo ocurrido en el incidente, detectando la procedencia del ataque y la actividad desarrollada por los intrusos dentro de los sistemas, además de que la metodología planteada cubrió todos los aspectos del análisis, pudimos determinar qué fue lo que paso y cuándo paso gracias a que la metodología cubre varios entornos de sistemas en producción porque su enfoque es global sin particularizar en una sola situación.

Con la cadena de custodia pudimos dar una evidencia formal, segura y controlada que nos facilitó reconstruir los hechos ocurridos de forma acertada; la rapidez con la que se ejecutó al análisis fue muy lento debido a que se necesita una mayor capacidad de procesamiento que el de cualquier computadora convencional del mercado, los análisis se realizaron bastante bien en lo que respecta al almacenamiento y captura de evidencia ya que no se alteró en ningún sentido gracias en gran medida a la cadena de custodia.

Las herramientas utilizadas son de las versiones más actuales ya que cubren bugs encontrados anteriormente, por lo que confiamos que la integridad de los datos recolectados es muy bueno, con esto pudimos determinar qué fue lo que ocurrió de una manera aceptable.

Quizá un factor que influenció en los reportes de los resultados fue nuestra propia experiencia, como ya se ha mencionado, el analista forense depende mucho de su experiencia para poder realizar el análisis forense de la manera más óptima, pero a pesar de esto, nuestra metodología pudo subsanar esta falta de experiencia pudiendo dar un reporte razonablemente aceptable.

- **El futuro de los IDS**

Aparentemente con esta definición podríamos deducir que los IDS son una fuerte herramienta que encara los problemas anteriormente descritos pero, como toda tecnología joven, aún se tiene que madurar en varios aspectos y tiene ciertas deficiencias

como lo son: los falsos positivos, no se tiene ningún estándar en diversos aspectos de su metodología, todavía se necesita mejorar la seguridad de los enlaces entre los elementos de IDS, se puede tener sobrecarga de alarmas, la detección de anomalías es poco común en la práctica.

En el futuro se deberán mejorar y corregir las deficiencias mencionadas así como se deberán ajustar a los futuros protocolos de red, también se tendrán una escalabilidad tolerable para que puedan seguir existiendo; conforme las redes y los equipos se hagan más complejos los IDS también, por lo que a futuro se optimizarán los recursos que consumirán los IDS, la siguientes generaciones de IDS comenzaran a utilizar de manera más frecuente el Data Minig[*]; seguramente también a futuro se inventarán nuevas técnicas y algoritmos para hacer más eficientes a los IDS.

Otro aspecto que tendrá un gran auge serán los IDS que puedan recolectar información de diversas fuentes, relacionando la información de gran parte de elementos de una infraestructura para poder tener una visión global de las actividades que se realizan, los detectores de intrusos tendrán un enfoque distribuido, muchos algoritmos utilizados en la detección de anomalías tendrán un papel muy importante en el desarrollo de este aspecto, así como establecer estándares relativos a los protocolos y un lenguaje de alarmas.

Finalmente todos los sistemas detectores de intrusos tenderán a reunirse en un solo dispositivo con todas las características de seguridad actuales, este dispositivo podría coordinar la labor de diferentes IDSs, por otra parte una serie de elementos distribuidos colaborarán con el dispositivo central, monitorizando la actividad, realizando tareas específicas y enviando sus resultados.

- **El futuro del análisis forense**

El análisis forense es una de las áreas que se desarrollará con gran auge en el futuro, así como también en que se tendrá más experiencia y madurez. Una de las áreas que presenta un gran campo de desarrollo lo es en la detección de intrusos, ya que estudios actuales demuestran que estas pasan desapercibidas en gran medida.

[*] El Data Minig, es una técnica por la cual se permite elaborar sus propios modelos estadísticos de forma automática a partir de los analizados pudiendo describir pautas de conducta.

Por otra parte los analistas forenses utilizan en gran medida herramientas de software que automatizan y aceleran el análisis computacional, en el futuro estas herramientas deberán hacerse más sofisticadas debido a:

La gran cantidad de datos que se podrán almacenar en una computadora, ya que la capacidad de la PC se esta incrementado en gran medida día con día.

La variedad de formatos de archivos, actualmente se esta investigando y desarrollando mucho sobre esta área principalmente en las técnicas de compresión de archivos y formatos más pequeños permitiendo su portabilidad, las herramientas forenses se deberán ajustar a la par.

Las herramientas del futuro deberán ejecutarse de una manera más rápida, para tratar de solventar las limitaciones de tiempo del analista forense, así como también deberán ser cada vez más precisas al recopilar información. Soportarán y elaborarán nuevos mecanismos de encriptación más robustos para manipular la evidencia obtenida.

También las siguientes herramientas son solo una parte que tendrá un significativo uso en el futuro tanto como en la investigación como en la práctica: Las herramientas para el Monitoreo y/o Control de Computadoras; herramientas de Marcado de documentos, para casos de robo de información. Hay que tener presente que la elección de las herramientas impactará directamente a la metodología cuando se responde a un incidente. La respuesta dependerá de las herramientas que se tengan disponibles.

La amenaza que es y seguirá siendo muy común serán los rootkits, se especializarán cada vez más, generando nuevos y más complejos archivos troyanos, el analista forense del futuro deberá estar preparado para enfrentar a esta amenaza, pero esto no es todo, también se deberá estar preparado para sniffers, crackeadores de passwords, programas de Irc, caballos de troya, exploits, gusanos, etc. En la situación actual los servidores se están perfilando a la red mundial. Los ataques que son y seguirán siendo más comunes son los que utilizan estos medios para afectar a sus objetivos, casi cualquier servidor esta conectado en la red, ya sea porque envía o recibe información, correo, chats, etc. A pesar de que los protocolos de red se están haciendo más robustos, el analista forense puede encontrar y ha encontrado un gran número de ataques por este medio.

La evasión forense será uno de los objetivos a minimizar en los siguientes años, ya que las herramientas de los hackers se hacen más sofisticadas y están mejorando sus técnicas para borrar huellas. El estudio forense con el paso del tiempo será menos vulnerable a la subversión de pruebas, pero no solo los intrusos ocasionan este problema sino también las herramientas forenses actuales, en el futuro éste será un campo en el que se invertirá un gran esfuerzo a fin de mejorar las herramientas forenses en este aspecto.

Hasta aquí hemos analizado los aspectos técnicos del análisis forense, pero las tácticas y metodologías de la investigación forenses son un tanto más importantes, en el futuro las organizaciones deberían modificar sus políticas de seguridad, de tal forma que la información en el momento del incidente no sufra grandes cambios y, de esta manera, los analistas forenses puedan reconstruir mejor los hechos. También las organizaciones tenderán a formar dentro de su propia organización sus propios equipos forenses, con la preparación suficiente como para llevar un caso.

Así como los CERTS se asociaron para formar el foro internacional llamado FIRST, los analistas forenses comienzan a formar sus propios foros donde se experimentan y se comparten casos interesantes con el fin de mejorar sus herramientas y técnicas, de esta manera estarán al tanto de las vulnerabilidades que existen para el sistema operativo en cuestión, saber de qué forma puede ser explotada una vulnerabilidad y cómo impacta al sistema. Los lasos entre los CERTS y los analistas forenses se estrecharán y trabajarán de una manera mejor coordinada al momento de resolver algún incidente de seguridad.

Por último, todavía falta mucho en lo que se refiere a la metodología forense, ya que hasta la fecha no se ha tenido un método estándar o una manera normalizada de realizar las técnicas y métodos de la investigación forense, todo lo que se ha estado realizando se basa en la propia metodología y la experiencia del analista pero no se tiene nada formalizado o estandarizado, de manera que entre las aportaciones de este trabajo de tesis está precisamente el dar un paso adelante en esta complicada y gran labor desarrollando una metodología de análisis forense que permita ayudar a las nuevas generaciones de Ingenieros en Computación a caminar en esta nueva y creciente área, aunada a la práctica que para ello se ha desarrollado y que los alumnos de la carrera podrán recrear en el laboratorio. En el futuro se tendrá más robustez y madurez en este

campo, se podrá hasta cierto punto normalizar y formalizar en la medida de lo que sea posible la metodología forense, en esta área hay mucha investigación por delante.

Actualmente la Facultad de Ingeniería no cuenta con el análisis forense para dar solución a los problemas antes descritos, al contar con una metodología forense informática, se tendría un mayor control de los ataques que se le hagan a la Facultad de Ingeniería, además de que se contaría con un historial para concienciar a todo el personal y administradores de equipos informáticos, sobre las deficiencias y vulnerabilidades que se tienen, para que de esta forma se puedan prevenir en el futuro y no ser víctima dos veces del mismo ataque.

APÉNDICE A
CONFIGURACIÓN DE HERRAMIENTAS DE SEGURIDAD INFORMÁTICA

Configuración de la herramienta The Coroner toolkit

Es un conjunto de herramientas de dominio público para analizar un sistema, fue realizado por Dan Farmer y Wietse Venema y funciona solamente en Unix o Linux.

The Coroner toolkit (TCT) no analiza los datos, solamente obtiene información relevante para el análisis. Las herramientas que incluye TCT son: Grave robber, que recolecta información sobre el equipo incluyendo los tiempos MAC de cada archivo; ils e icat permiten el listado y copia de archivos a nivel de nodos-i; unrm y lazarus recuperan información borrada del disco duro, clasificándola en función del tipo de archivo; finalmente mactime emplea la información recogida por grave robber para listar los tiempos de acceso a los archivos.

La instalación de TCT se realiza de la siguiente manera:

```
#tar -zxvf tct-1.13.tar.gz
```

Se descomprime el archive tct-1.13.tar.gz

```
#cd tct-1.13
```

Finalmente dentro del directorio se ejecuta la instalación con el comando

```
#make
```

Configuración de la herramienta SNORT

Snort es un Sniffer/logger de paquetes flexible que detecta ataques. Snort está basado en la biblioteca `libpcap' y puede ser usado como un "sistema de detección de intrusiones" (IDS). Posee un registro basado en reglas y puede buscar e identificar contenido de la información que viaja en la red, además de poder ser usado para detectar una gran variedad de otros ataques e investigaciones, como buffer overflows, barridos de puertos indetectables, ataques CGI, etc. Otra característica importante de Snort es la capacidad de alertar en tiempo real, siendo estas alertas enviadas a syslog, un archivo de alerta separado.

Una vez descargado el snort de la página http://www.snort.org se ejecuta lo siguiente:

```
# cp snort-2.1.3.tar.gz  /usr/src/redhat/SOURCES
# cd /usr/src/redhat/SOURCES
# tar –zxvf snort-2.1.3.tar.gz
# cd /usr/src/redhat/SOURCES/snort-2.1.3
# ./configure
# make
# make install
```

Posteriormente se especifican las reglas con las cuales se va a indicar cuando el programa se encuentre auditado, estas reglas se pueden descargar de la página http://www.snort.org/dl/snapshots/ y se pueden ir ajustando de acuerdo a las necesidades del equipo. Para instalarlo se siguen los siguientes pasos:

```
$ mkdir /etc/snort
$ cp snortrules.tar.gz to /etc/snort
$ tar –zxvf snortrules.tar.gz
$ cd /etc/snort/rules
$ mv * ../
$ cd ..
$ rmdir rules
$ vi snort.conf
$ mkdir /var/log/snort
```

Existe un archivo llamado snort.conf, que sirve para configurar el lugar en el cual snort va buscar las reglas, en dónde guardará los archivos de salida o si estos archivos estarán almacenados en una base de datos. Las líneas a modificar para este archivo son las siguientes:

var RULE_PATH ../rules (se cambia por): var RULE_PATH /etc/snort

Posteriormente se ejecuta el siguiente comando:

```
# /usr/local/bin/snort  -i eth1 –D -c /etc/snort/snort.conf
```

Donde:

-i se selecciona la interfaz sobre la cual se va correr el programa.

-D para correr como demonio y que ponga las alertas en el archivo /var/log/snort/alert.

-c se selecciona el archivo del cual va tomar la configuración.

Para ejecutar este comando de manera automática se debe colocar un script en /etc/rc.d/init.d

Configuración PORTSENTRY

PortSentry es una herramienta que permite la fácil detección de barridos de puertos y tiene la capacidad de actuar de inmediato al detectarlos.

A continuación mostramos los pasos que se deben seguir para instalar Portsentry en Linux.

Primero se debe descargar el programa de Internet de la dirección siguiente: "http://www.psionic.com/abacus/portsentry/portsentry-1.1.tar.gz"

Después se procede a descomprimir y desempaquetar el archivo portsentry-1.1.tar.gz.

```
# cd /usr/local
# tar -zxvf /root/portsentry/portsentry-1.1.tar.gz
```

Cambiarse al directorio portsentry-1.1.

```
# cd portsentry-1.1
```

Posteriormente se debe editar el archivo portsentry.conf.

```
# vi portsentry.conf
```

```
# Use these if you just want to be aware:
TCP_PORTS="1,11,15,79,111,119,143,540,635,1080,1524,2000,5742,6667,12345,12346,20034,276
65,31337,32771,32772,32773,32774,40421,49724,54320"
UDP_PORTS="1,7,9,69,161,162,513,635,640,641,700,37444,34555,31335,32770,32771,32772,3277
3,32774,31337,54321"

# On many Linux systems you cannot bind above port 61000. This is because
# these ports are used as part of IP masquerading. I don't recommend you
# bind over this number of ports. Realistically: I DON'T RECOMMEND YOU MONITOR
# OVER 1024 PORTS AS YOUR FALSE ALARM RATE WILL ALMOST CERTAINLY RISE.
You've been
# warned! Don't write me if you have have a problem because I'll only tell
# you to RTFM and don't run above the first 1024 ports.
```

```
#
#
ADVANCED_PORTS_TCP="1024"
ADVANCED_PORTS_UDP="1024"
```

Estas líneas no se deben alterar.

```
##################
# Ignore Options #
##################
# These options allow you to enable automatic response
# options for UDP/TCP. This is useful if you just want
# warnings for connections, but don't want to react for
# a particular protocol (i.e. you want to block TCP, but
# not UDP). To prevent a possible Denial of service attack
# against UDP and stealth scan detection for TCP, you may
# want to disable blocking, but leave the warning enabled.
# I personally would wait for this to become a problem before
# doing though as most attackers really aren't doing this.
# The third option allows you to run just the external command
# in case of a scan to have a pager script or such execute
# but not drop the route. This may be useful for some admins
# who want to block TCP, but only want pager/e-mail warnings
# on UDP, etc.
#
#
# 0 = Do not block UDP/TCP scans.
# 1 = Block UDP/TCP scans.
# 2 = Run external command only (KILL_RUN_CMD)

BLOCK_UDP="1"
BLOCK_TCP="1"
```

Por último se debe descomentar lo siguiente:

KILL_ROUTE="/sbin/route add -host $TARGET$ reject"

Se debe compilar portsentry eligiendo el sistema operativo correspondiente

```
# make linux
SYSTYPE=linux
Making
cc -O -Wall -DLINUX -DSUPPORT_STEALTH -o ./portsentry ./portsentry.c \
./portsentry_io.c ./portsentry_util.c
```

Lo siguiente ejecutará la instalación de portsentry

```
# make install
```

Finalmente se guardó en el direcotorio /usr/local/psionic/portsentry, que contiene los siguientes archivos: portsentry portsentry.conf portsentry.ignore

Una vez instalado se debe configurar de modo avanzado para que el Kernel notifique si llega una petición hacia algún puerto que sea menor al que se especifica en ADVANCED_PORTS_TCP y ADVANCED_PORTS_UDP con el fin de que los puertos no se encuentren realmente abiertos, esto se realiza con las siguientes líneas de comando:

```
# /usr/local/psionic/portsentry/portsentry –atcp

# /usr/local/psionic/portsentry/portsentry –audp
```

Configuración de la herramienta TRIPWIRE

Esta herramienta es un comprobador de integridad para archivos y directorios de sistemas Unix: tripwire compara un conjunto de estos objetos con la información sobre los mismos almacenada previamente en una base de datos, y alerta al administrador en caso de que algo haya cambiado. La idea es simple: se crea un resumen de cada archivo o directorio importante para nuestra seguridad nada más instalar el sistema, y esos resúmenes se almacenan en un medio seguro (un CD-ROM o un disco protegido contra escritura), de forma que si alguno de los archivos es modificado t*ripwire* nos alertará la próxima vez que realicemos la comprobación. Para generar esos resúmenes se utilizan funciones *hash*, de forma que es casi imposible que dos archivos generen el mismo resumen; concretamente *Tripwire* implementa MD2, MD4, MD5.

Tripwire se instaló utilizando el RPM de Tripwire como usuario root de la siguiente forma:

```
# rpm –ivh  tripwire-2.3.1-17.i386.rpm
```

Esta herramienta se configura de la siguiente manera:

Modificando el archivo “/etc/tripwire/twcfg.txt” aquí se declara la localización de tripwire y se definen algunas variables de entorno, también se tiene que editar el siguiente archivo “/etc/tripwire/twpol.txt”. Este archivo de política permite darse cuenta de las aplicaciones específicas, de los archivos y de los directorios del sistema con el fin de optimizar los informes de Tripwire minimizando los falsos positivos.

El siguiente script en su ejecución pedirá contraseñas del sitio y local para generar llaves de protección de los archivos Tripwire, finalmente crea y firma los archivos de configuración, de política, la base de datos y los archivos de informe, protegiéndolos de los intrusos que no conocen las contraseñas locales ni de los sitios.

/etc/tripwire/twinstall.sh

El resultado de este paso es el siguiente:

Creating key files...

(When selecting a passphrase, keep in mind that good passphrases typically have upper and lower case letters, digits and punctuation marks, and are at least 8 characters in length.)

Enter the site keyfile passphrase:
Verify the site keyfile passphrase:
Generating key (this may take several minutes)...Key generation complete.
(When selecting a passphrase, keep in mind that good passphrases typically have upper and lower case letters, digits and punctuation marks, and are at least 8 characters in length.)

Enter the local keyfile passphrase:
Verify the local keyfile passphrase:
Generating key (this may take several minutes)...Key generation complete.

Signing configuration file...
Please enter your site passphrase:
Wrote configuration file: /etc/tripwire/tw.cfg
A clear-text version of the Tripwire configuration file
/etc/tripwire/twcfg.txt
has been preserved for your inspection. It is recommended
that you delete this file manually after you have examined it.

Signing policy file...
Please enter your site passphrase:
Wrote policy file: /etc/tripwire/tw.pol
A clear-text version of the Tripwire policy file /etc/tripwire/twpol.txt
has been preserved for your inspection. This implements a minimal policy, intended only to test essential
Tripwire functionality. You should edit the policy file to describe your system, and then use twadmin to generate a new signed copy of the Tripwire policy.

Una vez encriptados y firmados, no se puede cambiar el nombre ni mover los archivos. Posteriormente se procede a inicializar los archivos de base de datos, de la siguiente manera:

```
#/usr/sbin/tripwire –init

Please enter your local passphrase:

Parsing policy file: /etc/tripwire/tw.pol

Generating the database...

*** Processing Unix File System ***

Wrote database file: /var/lib/tripwire/xelha.twd

The database was successfully generated.
```

El script de shell tripwire-check en el directorio /etc/cron.daily/ ejecuta automáticamente un control de integridad una vez al día. Pudiendo realizar un control de integridad mediante el comando:

```
# /usr/sbin/tripwire --check

Parsing policy file: /etc/tripwire/tw.pol

*** Processing Unix File System ***

Performing integrity check...

Wrote report file: /var/lib/tripwire/report/maquina-20040622-103505.twr

Tripwire(R) 2.3.0 Integrity Check Report

Report generated by:        root

Report created on:          lun 22 jun 2004 10:35:05 CDT

Database last updated on:   Never


===============================================================
Report Summary:
===============================================================
Host name:                  servidor
Host IP address:            127.0.0.1
Host ID:                    None
Policy file used:           /etc/tripwire/tw.pol
Configuration file used:    /etc/tripwire/tw.cfg
Database file used:         /var/lib/tripwire/servidor.twd
Command line used:          /usr/sbin/tripwire --check
```

Con el comando twprint -m r se genera un informe Tripwire en modo texto al cual se le debe especificar el archivo de informe a mostrar, como se ilustra a continuación:

```
# /usr/sbin/twprint -m r --twrfile /var/lib/tripwire/report/<name>.twr
```

Confguración de SCANLOGD

Scanlogd es un demonio escrito por 'Solar Designer' para detectar barridos de puertos a un sistema en producción.

Para poderlo instalar, primero se debe descargar el programa de Internet de la dirección http://www.rpmfind.net. Posteriormente se instala:

```
# rpm –ivh /root/scanlogd/scanlogd-2.2-1.5.i386.rpm
```

Después se edita el archivo /etc/syslog.conf:

```
# vi /etc/syslog.conf
```

Agregándole las siguientes líneas:

```
# scanlogd
daemon.alert /var/log/escaneos
```

Una vez realizado lo anterior se debe editar el archivo /etc/rc.d/rc.local, con la siguiente línea:

```
/etc/init.d/scanlogd &
```

El archivo escaneos que se encuentra dentro de /var/log registra cuándo se realiza un escaneo, al borrar dicho archivo es necesario reiniciar el demonio, como se muestra a continuación:

```
# /etc/init.d/syslog restart
```

APÉNDICE B
GRABACIÓN EN MÉDIOS MAGNÉTICOS

APÉNDICE B. GRABACIÓN EN MEDIOS MAGNÉTICOS

Grabación en Medios Magnéticos: Principios Físicos

En general, los medios de almacenamiento magnético se basan directamente en cuatro fenómenos físicos:

A. Una corriente eléctrica produce un campo magnético

B. Algunos materiales se magnetizan con facilidad cuando son expuestos a un campo magnético débil. Cuando el campo se apaga, el material se desmagnetiza rápidamente. Se conocen como *Materiales Magnéticos Suaves*.

C. En algunos materiales magnéticos suaves, la resistencia eléctrica cambia cuando el material es magnetizado. La resistencia regresa a su valor original cuando el campo magnetizante es apagado. Esto se llama *Magneto-Resistencia*, o efecto MR. La Magneto-Resistencia Gigante, o efecto GMR, es mucho mayor que el efecto MR y se encuentra en sistemas específicos de materiales de películas delgadas.

D. Otros materiales se magnetizan con dificultad (es decir, requieren de un campo magnético fuerte), pero una vez que se magnetizan, mantienen su magnetización cuando el campo se apaga. Se llaman *Materiales Magnéticos Duros*, o *Magnetos Permanentes*.

Estos cuatro fenómenos son explotados por los fabricantes de cabezas grabadoras magnéticas, que leen y escriben datos, para almacenar y recuperar información en unidades de disco, de cinta y otros dispositivos de almacenamiento magnético.

Aplicaciones en almacenamiento de datos:

- Cabezas de Escritura: Cabezas usadas para escribir bits de información en un disco magnético giratorio, dependen de los fenómenos A y B para producir y controlar campos magnéticos fuertes.

- Cabezas de lectura: Éstas dependen de los fenómenos A, B y C y son sensibles a los campos magnéticos residuales de los medios de almacenamiento magnetizados (D).

-. Medios de Almacenamiento: Los medios de almacenamiento magnético son magnetizados de manera permanente en una dirección (Norte o Sur) determinada por el campo de escritura. Estos medios explotan el fenómeno D.

Escribiendo Datos Magnéticos

En la figura A.1 se muestra un esquema simplificado de una cabeza de escritura. La vista superior de una cabeza de escritura (izquierda) muestra un rollo espiral, envuelto entre dos capas de material magnético suave; a la derecha está un corte transversal de esta cabeza, vista de lado. Nótense dos detalles sobre esta figura: En el extremo inferior, hay un espacio entre las capas, y en el extremo superior, las capas están unidas. Las capas superior e inferior de material magnético se magnetizan con facilidad cuando fluye una corriente eléctrica en el rollo espiral, de tal forma que estas capas se vuelven los polos Norte y Sur magnéticos de un pequeño electro-magneto (En una cabeza real, la distancia desde el espacio hasta la parte superior del rollo es de aproximadamente 30 mm).

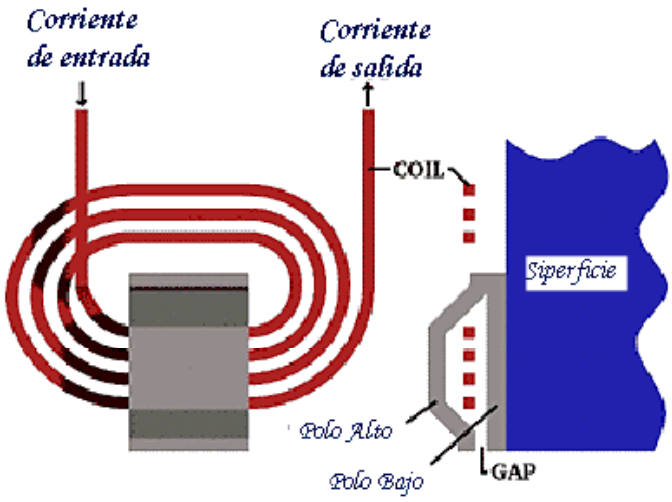


Fig. B.1 Una cabeza de escritura

Los polos N-S en el extremo de la separación de la cabeza de escritura concentran el campo para hacer de esta región el “extremo de negociación”, que es el área en donde el campo de escritura sale al espacio por fuera de la cabeza. Cuando un medio de almacenamiento magnético es ubicado muy cerca de la cabeza de escritura, el material magnético duro en la superficie del disco queda magnetizado de manera permanente (escrito) con una polaridad que corresponde a la del campo de escritura. Si la polaridad

de la corriente eléctrica se invierte, la polaridad magnética en la separación también se invierte.

Las computadoras almacenan datos en un disco giratorio en la forma de dígitos binarios, o bits transmitidos a la unidad de disco en una secuencia de tiempo correspondiente a los dígitos binarios (*bits*) uno y cero. Estos bits son convertidos en una onda de corriente eléctrica que es transmitida por medio de cables al rollo de la cabeza de escritura. Este proceso se esquematiza en la figura A.2. En su forma más simple, un *bit* uno corresponde a un cambio en la polaridad de la corriente, mientras que un *bit* cero corresponde a una ausencia de cambio en la polaridad de la corriente de escritura. Entonces, un disco en movimiento es magnetizado en la dirección positiva (Norte) para una corriente positiva y es magnetizado en la dirección negativa (Sur) para un flujo de corriente negativo. En otras palabras, los unos almacenados aparecen en donde ocurre una inversión en la dirección magnética en el disco, y los ceros residen entre los unos.

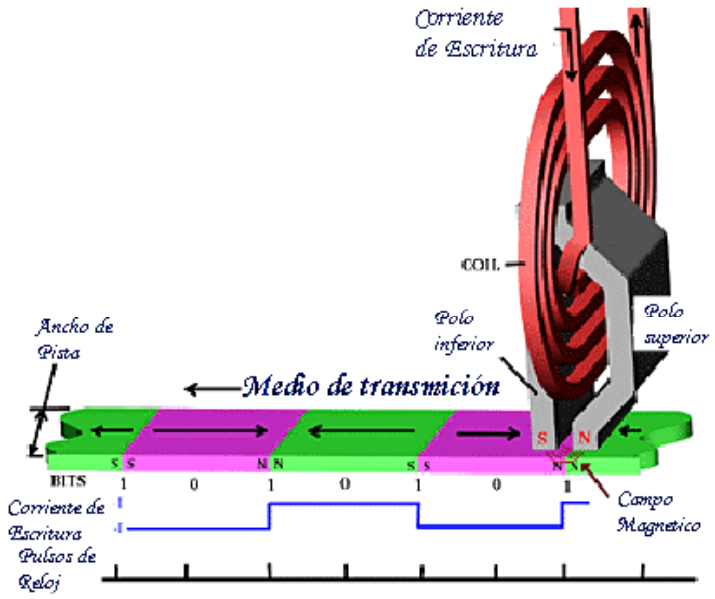


Fig. B.2 Escribiendo datos en un medio de almacenamiento

Un reloj de regulación está sincronizado con la rotación del disco y existen *celdas de* un *bit* para cada pulso del reloj; algunas de estas celdas de bits representarán un *uno* (una inversión en la dirección magnética, tal como N cambiando a S o S cambiando a N) y otras representarán *ceros* (polaridad N constante o S constante). Una vez escritos, los bits en la superficie del disco quedan magnetizados permanentemente en una dirección o la otra, hasta que nuevos patrones sean escritos sobre los viejos.

Existe un campo magnético relativamente fuerte directamente sobre la localización de los *unos* y su fuerza se desvanece rápidamente a medida que la cabeza de grabación se aleja. Un movimiento significativo en cualquier dirección que se aleje de un *uno,* causa una dramática pérdida en la fuerza del campo magnético, lo que implica que para detectar bits de datos de manera confiable, es extremadamente importante que las cabezas de lectura vuelen muy cerca de la superficie del disco magnetizado.

Leyendo Datos Magnéticos

En la actualidad, las cabezas de lectura leen datos magnéticos mediante resistores magnéticamente sensitivos llamados *Válvulas Spin* que explotan el efecto GMR. Estas cabezas *GMR/Válvula Spin* son situadas muy cerca del disco de almacenamiento magnético rotatorio, exponiendo el elemento GMR a los campos magnéticos de *bit* previamente escritos en la superficie del disco. Si la cabeza GMR se aleja ligeramente del disco (2 o 3 millonésimas de pulgada) la intensidad del campo cae por fuera de un nivel útil, y los datos magnéticos no pueden ser recuperados fielmente.

Cuando una corriente atraviesa el elemento GMR, los cambios en la resistencia (correspondientes a los cambios en los estados magnéticos que surgen de los bits escritos N y S) son detectados como cambios en el voltaje. Estas fluctuaciones de voltaje –es decir, la señal- son conducidas a las terminales sensoras del GMR. Sin embargo, el ruido eléctrico está presente en todos los circuitos eléctricos (las cabezas GMR no son la excepción), por lo que la señal combinada con el ruido de un lector GMR son enviados por medio de cables a los circuitos electrónicos de la unidad de disco, para decodificar la secuencia de tiempo de los impulsos (y los espacios entre los impulsos) en unos y ceros binarios. El proceso de lectura, incluyendo el indeseable pero siempre presente ruido, se esquematiza en la figura A.3.

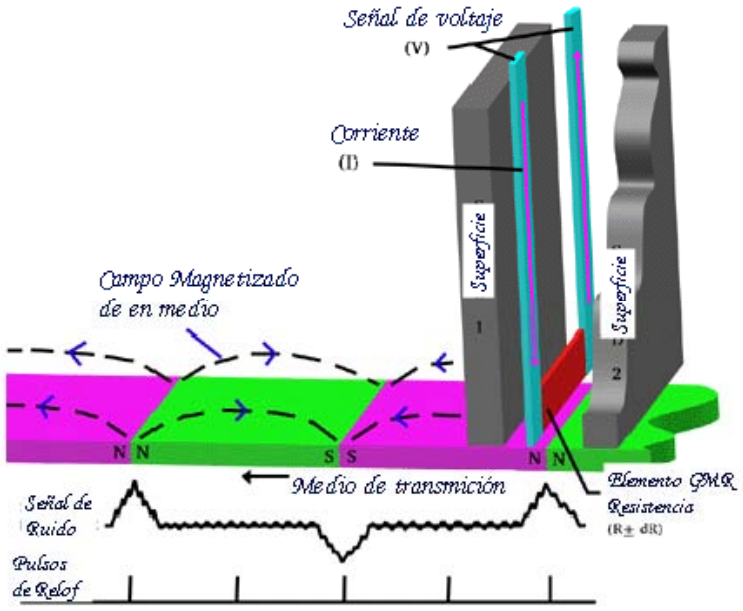


Fig. B.3 Leyendo datos desde un medio de almacenamiento

Eliminación de Datos en un Medio Magnético

Borrar de manera definitiva los datos en un medio magnético tiene toda una problemática asociada. Como se vio anteriormente, la información es escrita y leída aprovechando las características de magnetización de un material determinado. Sin embargo, y dependiendo del medio usado (unidades de disco, cintas, diskettes, etc.), el proceso de eliminación total de los datos se ve afectado por diversos factores.

Degaussing de Medios Magnéticos

La Matriz de Limpieza y Sanitización es una acumulación de métodos conocidos y aprobados para limpiar y/o sanitizar diversos medios y equipo. Cuando NISPOM fue publicado, el Rango Extendido Tipo II, Tipo III y los *degaussers* de propósito especial no existían. Esto resultaba en la necesidad de destruir todos los medios con un factor de coercividad (cantidad de fuerza eléctrica requerida para reducir la fuerza magnética grabada a cero) mayor que 750 oersteds (unidad que mide la fuerza magnetizante necesaria para producir una fuerza magnética deseada a lo largo de una superficie) y la mayoría de discos magnéticos cuando ya no fueran necesarios como soporte para una misión clasificada. Ahora, la “National Security Agency norteamericana” (NSA) ha

evaluado *degaussers* de cinta magnética que satisfacen los requerimientos del gobierno para sanitizar cintas magnéticas de hasta 1700 oersteds.

Las cintas magnéticas se encuentran divididas en Tipos. La cinta magnética de Tipo I tiene un factor de coercividad que no excede los 350 oersteds y puede ser usada para sanitizar (*degauss*) todos los medios de Tipo I .La cinta magnética de Tipo II tiene un factor de coercividad entre 350 y 750 oersteds y puede ser usada para sanitizar todos los medios Tipo I y II. La cinta magnética Tipo II de Rango Extendido tiene un factor de coercividad entre 750 y 900 oersteds y puede ser usada para sanitizar todos los medios Tipo I, Tipo II y Rango Extendido. Finalmente, las cintas magnéticas Tipo III, comúnmente conocidas como cintas de alta energía (por ejemplo, cintas de 4 ó 8mm) tiene un factor de coercividad actualmente identificado como entre 750 y 1700 oersteds y puede ser usada para sanitizar todos los tipos de cintas magnéticas.

Para sanitizar (*degauss*) todos los medios de disco, rígidos o flexibles (por ej., *diskettes*, Bernoulli, Syquest y unidades de Disco Duro) se deben usar *degaussers* de Unidad de Disco. Para este tipo de dispositivos la NSA tiene una nueva categoría de *degaussers*, conocida como *Degaussers* de Propósito Especial. DSS, como todas las agencias del DoD, referencia el “Information Systems Security Products and Services Catalog” como guía de sanitización de memoria y medios. NSA publica el “Information Systems Security Products and Services Catalog” entre sus productos y servicios de seguridad para sistemas de información. La lista de productos *degausser* (DPL) está dedica a los *degaussers* de discos y cintas magnéticas. La DPL hace un excelente trabajo identificando los fabricantes de *degaussers* y los diferentes tipos de éstos.

Eliminación de Datos en CDs

Los datos de un CD están almacenados en la parte superior del CD por medio de una capa reflectiva que es leída por un láser. Los CDs ofrecen buenas alternativas para almacenar información por largos periodos de tiempo, pero puede ser necesario destruirlos. Se mencionan algunos medios para hacer esto:

1. Retiro de la lámina reflectiva: Se puede retirar la lámina con algún elemento cortante, sin embargo se debe destruir la lámina reflectiva, y aún así pueden quedar algunos rastros de datos en el policarbonato.

2. Cortar en pedazos: Con una cortadora industrial de papel, el CD podría ser destruido, sin embargo, la lámina reflectiva podría separarse del CD y no ser cortada correctamente.

3. Destruir el CD por medios químicos: Una posible alternativa es introducir el CD en Acetona, lo cual dejaría la lámina superior inservible, sin embargo es posible que la lámina de policarbonato aún contenga algunos rastros de información.

4. Destrucción por Incineración: Probablemente es el método más rápido y eficiente, pero es realmente nocivo para el medio ambiente. El humo del policarbonato puede ser perjudicial para la salud de las personas.

5. Destrucción por medio de un horno microondas: Introduciendo el CD en un microondas por unos 3 segundos puede destruir gran parte del CD, sin embargo no todas las partes serán destruidas. Este método no se recomienda, especialmente porque puede dañar el horno debido a los campos magnéticos que usa el horno y que pueden causar un cortocircuito ya que el CD contiene metales.

6. Reescritura: Para los CDs re-escribibles, es posible volverlos a escribir de tal forma que el proceso dañe los datos. Sin embargo, no se sabe si por mecanismos especiales sea posible recuperar la información.

7. Rayado Simple: A menos que uno quiera ser realmente precavido, la forma mas fácil de destruir un CD es rayando la parte superior. La razón por la que se debe rayar la parte superior es porque es esta la que mantiene los datos. Si es rayada la parte inferior sería fácil recuperar la capa y corregir el problema, utilizando productos comerciales para recuperar CDs.

APÉNDICE C
CUESTIONARIO, CADENA DE CUSTODIA Y REPORTE FINAL

No. FOLIO _____
Fecha: _____

CUESTIONARIO PARA EL ANÁLISIS FORENSE

Información del encuestado:

Nombre completo:

_____
_____

Puesto que desempeña:

_____

Preguntas:

1.- ¿Cuántas personas cuentan con el password de administrador?

_____

2.- ¿Usted realizó o participó en la última instalación del sistema?

_____

5.- ¿Cuánto tiempo llevan los administradores del sistema a cargo del mismo?

_____

6.- ¿Hubo algún despido recientemente de algún encargado?

_____

7.- ¿El sistema qué servicios ofrece?

_____
_____

8.- ¿Cuál es el número de usuarios que tienen cuenta en el sistema?

_____

9.- ¿Qué distribución de Linux está instalado en el equipo?

_____

10.- ¿Cuándo fue la fecha en que aplicó usted el último parche o actualización al sistema y de qué fue?

_____

11.- ¿Cuándo fue la última vez que se realizó una auditoria al sistema?

_____

12.- ¿Cuáles son los motivos que le hacen pensar que tiene un problema de seguridad?

_____
_____

13.- ¿Cuál es la fecha exacta y hora aproximada en que se dieron cuenta de que se tenía un incidente de seguridad?

_____

14.- ¿Anterior a la fecha dada en la pregunta anterior, ¿había detectado algo anormal en el sistema?

_____

Si es afirmativo, ¿Qué fue?

_____
_____

15 ¿Cuenta con políticas de respuesta a incidentes dentro de la institución?

_____

Si es afirmativo, ¿Se aplicaron las políticas, y qué se obtuvo de la aplicación de las mismas?

_____

16.- ¿Tiene el sistema involucrado en el incidente alguna relación de confianza con otro(s) servidores?

_____

17.- ¿Realizan tareas administrativas remotamente?

_____

Si es afirmativo, ¿Cuáles y con qué herramientas?

_____
_____
_____

18.- ¿Cuenta con un Server log, cuál es la IP del mismo?

_____

19.- ¿Algún servicio o servicios se vieron afectados por el incidente?

_____
_____

20.- ¿Qué herramientas de seguridad tiene su sistema? (Mencione todas)

_____
_____

21.- ¿Cuenta con respaldos de su sistema?

_____

Si la pregunta anterior es afirmativa, ¿Cuándo fueron realizados?_____
_____

22.- ¿Cuenta con un IDS o con alguna herramienta implementada para analizar su red?

_____

Si es afirmativo. ¿Tiene registros guardados a partir de la fecha en que se sospecha la intrusión?

_____

23.- ¿En estos momentos el sistema se encuentra en producción o apagado?

_____

24.- Si está apagado, ¿De qué forma fue apagado?

_____

25.- Si está en producción, ¿Se ha reiniciado el sistema desde que ocurrió el incidente hasta ahora?

_____

26.- Si está en producción, ¿El equipo sigue conectado a la red?

_____

27.- ¿Qué se detectó en bitácoras?

_____

28.- ¿Desde el momento en que se detectó el incidente se tiene resguardado lo que se detectó?

_____

29.- ¿Puede explicar a grandes rasgos lo que ha hecho por controlar el incidente, la forma en que se detectó, si ustedes han borrado algún archivo importante a partir del incidente, qué archivos modificó el intruso, etc.?

_____
_____
_____

30.- ¿Qué se pretende indagar o perseguir?

_____

31.- ¿En qué runlevel se encuentra el sistema?

_____

32.- ¿Cuándo fue la fecha en que se instaló el sistema?

_____

_____

CASO NÚMERO: ____

ETIQUETA: _____

FECHA: _____

HORA: _____

CADENA DE CUSTODIA DE LA EVIDENCIA.

Número de personas que recolectaron la evidencia: _____

Nombre completo de quién recibe la evidencia: _____

_____

_____

Datos del equipo revisado:

Arquitectura: _____

Modelo: _____

Número de serie: _____

Datos Técnicos: _____

_____

_____

Datos de la evidencia:

Ubicación donde se encontró la evidencia:

_____

_____

_____

De quién se obtuvo la evidencia:

_____

_____

Descripción de la evidencia:

_____

_____

_____

_____

_____

Valores hash (cuando sea el caso):

_____

_____

_____

_____

_____

_____

REPORTE FINAL

CASO NÚMERO: ___

Fecha: _____

Responsables:

Analista 1:

_____

Analista 2:

_____

Organización Responsable:

_____

_____

Dirección:

_____

_____

_____

Ubicación del equipo comprometido:

_____

_____

_____

Responsable(s) del equipo comprometido:

_____

_____

_____

_____

_____

Descripción del caso:

¿Donde se ha producido la intrusión?

¿Qué hizo el atacante en la máquina?

Recomendaciones sobre la intrusión

CASO PRÁCTICO 1

No. FOLIO__0001
Fecha: __01 Julio

2004

CUESTIONARIO PARA EL ANÁLISIS FORENSE

Información del encuestado:

Nombre completo:
_____Jorge Acevedo Cañedo_____

Puesto que desempeña:

Administrador del equipo involucrado

Preguntas:

1.- ¿Cuántas personas cuentan con el password de administrador?
__una_____

2.- ¿Usted realizó o participó en la última instalación del sistema?
__Sí participé_____

5.- ¿Cuánto tiempo llevan los administradores del sistema a cargo del mismo?
_____un año_____

6.- ¿Hubo algún despido recientemente de algún encargado?
_____No_____

7.- ¿El sistema qué servicios ofrece?
_____FTP, Sendmail, WEB, Ssh y mysql_____

8.- ¿Cuál es el número de usuarios que tienen cuenta en el sistema?
_____tres usuarios_____

9.- ¿Qué distribución de Linux está instalado en el equipo?
_____Red Hat 6.2_____

10.- ¿Cuándo fue la fecha en que aplicó usted el último parche o actualización al sistema y de qué fue?
No se ha actualizado ni parchado nunca desde que se instaló el equipo a la fecha_____

11.- ¿Cuándo fue la última vez que se realizó una auditoria al sistema?
Nunca se ha realizado una auditoria

12.- ¿Cuáles son lo motivos que le hacen pensar que tiene un problema de seguridad?
Las bitácoras del sistema fueron limpiadas

13.- ¿Cuál es la fecha exacta y hora aproximada en que se dieron cuenta de que se tenía un incidente de seguridad?
El 15 de Junio del 2004 como a las 18:00

14.- ¿Anterior a la fecha dada en la pregunta anterior, ¿había detectado algo anormal en el sistema?
No

Si es afirmativo, ¿Qué fue?

15 ¿Cuenta con políticas de respuesta a incidentes dentro de la institución?
No

Si es afirmativo, ¿Se aplicaron las políticas y qué se obtuvo de la aplicación de las mismas?

16.- ¿Tiene el sistema involucrado en el incidente alguna relación de confianza con otro(s) servidores?
No

17.- ¿Realizan tareas administrativas remotamente?
No

Si es afirmativo, ¿Cuáles y con qué herramientas?

18.- ¿Cuenta con un Server log, cuál es el IP del mismo?
No

19.- ¿Algún servicio o servicios se vieron afectados por el incidente?
No

20.- ¿Qué herramientas de seguridad tiene su sistema? (Mencione todas)
Ninguna

21.- ¿Cuenta con respaldos de su sistema?
No

Si la pregunta anterior es afirmativa, ¿Cuándo fueron realizados?

22.- ¿Cuenta con un IDS o con alguna herramienta implementada para analizar su red?
No

Si es afirmativo. ¿Tiene registros guardados a partir de la fecha en que se sospecha la intrusión?

23.- ¿En estos momento el sistema se encuentra en producción o apagado?
Apagado

24.- Si está apagado, ¿De qué forma fue apagado?
Oprimiendo el botón de apagado

25.- Si está en producción, ¿Se ha reiniciado el sistema desde que ocurrió el incidente hasta ahora?

26.- Si está en producción, ¿El equipo sigue conectado a la red?

27.- ¿Qué se detectó en bitácoras?
No existen bitácoras antes de la fecha del incidente

28.- ¿Desde el momento en que se detectó el incidente se tiene resguardado lo que se detectó?
No

29.- ¿Puede explicar a grandes rasgos lo que ha hecho por controlar el incidente, la forma en que se detectó, si ustedes han borrado algún archivo importante a partir del incidente, qué archivos modificó el intruso, etc.?
Se siguieron revisando bitácoras

_____
_____
_____
_____

30.- ¿Qué se pretende indagar o perseguir?
Qué fue lo que pasó

31.- ¿En qué runlevel se encuentra el sistema?
En el 5

32.- ¿Cuándo fue la fecha en que se instaló el sistema?

Fue el 12 de Junio del 2004

CASO NÚMERO: 0001

ETIQUETA: EV01

FECHA:01 Julio 2004

HORA: 11:35 AM

CADENA DE CUSTODIA DE LA EVIDENCIA.

Número de personas que recolectaron la evidencia: 2

Nombre completo de quién recibe la evidencia:

Víctor Hugo Sánchez Quijada

Datos del equipo revisado:

Arquitectura: PC de escritorio en torre

Modelo: Armada

Número de serie: sin número

Datos Técnicos: Equipo LANIX con 32 Mb de RAM. Este equipo tiene un disco duro de 5.7 Gb, el microprocesador es Pentium I a 100 MHz, la tarjeta de red es Ethernet con sistema operativo Linux Red Hat 6.2. Tiene una IP no homologada 192.168.2.7. Este equipo es armado y dispone con una unidad de CD-ROM y otra unidad de floppy.

Datos de la evidencia:

Ubicación donde se encontró la evidencia:

Laboratorio de Redes y Seguridad de la División de Ingeniería Eléctrica de la Facultad de la Ingeniería

De quién se obtuvo la evidencia:

Del responsable del equipo comprometido, Jorge Acevedo Cañedo

Descripción de la evidencia:

____La evidencia consiste en un Disco Duro de 5.7 GB al que se le efectuó la firma md5 de todas las particiones para así poder llevar a cabo la copia de la evidencia sobre otro disco duro no contaminado.

Valores hash (cuando sea el caso):

/boot = 4f1fba9524374ab71300b25b09b96b60

/ = 39321d15766953cf2a33d2a6d66c5bd9

REPORTE FINAL

CASO NÚMERO: 0001

Fecha: 05 Julio 2004

Analistas:

Analista 1: Víctor Hugo Sánchez Quijada

Analista 2: Oswaldo Gómez Gallardo

Organización Responsable:

Área de Redes y Seguridad en Cómputo de la División de Ingeniería Eléctrica de la Facultad de Ingeniería

Dirección:

Ubicación del equipo comprometido:

Laboratorio de Redes y Seguridad en Cómputo, Primer Planta baja del Edificio Valdez Vallejo, División de Ingeniería Eléctrica

Responsable(s) del equipo comprometido:

Jorge Acebedo Cañedo

Descripción del Caso:

El equipo 192.168.2.7 fue instalado el día 12 de Junio del 2004. Se instaló el sistema operativo Red Hat 6.2 sin ninguna actualización; el sistema comprometido tenía los servicios de red activos, X11, portmap, mysql, httpd, wu-ftpd, samba, ssh, sendmail.

El sistema fue comprometido el día 15 de Junio del 2004 entre las 14:36 horas y las 15:26 horas.

¿Donde se ha producido la intrusión?

El equipo fue comprometido por un usuario del sistema con el login ‘usuario1’. Este usuario descargo varios tipos de exploits encontrando dos programas, uno que explota vulnerabilidades de sendmail escrito en C y otro en Shell Script que ataca al programa SuidPerl. Los servicios httpd, wu-ftpd, samba, sendmail tenían importantes problemas de seguridad.

¿Qué hizo el atacante en la máquina?

El ‘usuario1’ ingresó al sistema de forma legítima y descargó los exploits de sendmail y de SuidPerl para vulnerar el sistema y obtener una sesión de administrador, cuando lo logró, se dispuso a instalar un RootKit llamado Adore que reside en el núcleo como módulo, interceptando determinadas llamadas de sistema, de esta forma, Adore puede ocultar la presencia de un intruso en el equipo, tales como conexiones desde direcciones remotas desconocidas, intérpretes de comandos con privilegios de administrador que estén ejecutándose, o archivos con contraseñas capturadas.

Recomendaciones sobre la intrusión

El sistema debería contar sólo con los servicios activos más utilizados y tener actualizaciones o instalar un sistema operativo más reciente. Se debe mejorar la seguridad de los siguientes servicios activos: httpd, wu-ftpd, samba, sendmail y programas internos como lo es SuidPerl.

Se recomienda utilizar mecanismos de seguridad como lo son: un Firewall interno, programas que verifiquen la integridad del sistema, herramientas que detecten vulnerabilidades en el sistema.

CASO PRÁCTICO 2

No. FOLIO__0002
Fecha: 25 Julio 2004

CUESTIONARIO PARA EL ANÁLISIS FORENSE

Información del encuestado:

Nombre completo: _____Jorge Acebedo Cañedo_____

Puesto que desempeña: _Administrador del equipo involucrado_____

Preguntas:

1.- ¿Cuántas personas cuentan con el password de administrador?
_____1_____

2.- ¿Usted realizó o participó en la última instalación del sistema?
_____Si_____

5.- ¿Cuánto tiempo llevan los administradores del sistema a cargo del mismo?
_____Como 15 días_____

6.- ¿Hubo algún despido recientemente de algún encargado?
_____No_____

7.- ¿El sistema qué servicios ofrece?
_____FTP, NFS, Impresión, SSH y Correo_____

8.- ¿Cuál es el número de usuarios que tienen cuenta en el sistema?
_____1_____

9.- ¿Qué distribución de Linux está instalado en el equipo?
_____Red Hat 7.1_____

10.- ¿Cuándo fue la fecha en que aplicó usted el último parche o actualización al sistema y de qué fue?
_____Nunca se actualizó_____

11.- ¿Cuándo fue la última vez que se realizó una auditoria al sistema?
_____Nunca se ha auditado_____

12.- ¿Cuáles son lo motivos que le hacen pensar que tiene un problema de seguridad?
Registro en bitácoras de conexiones extrañas de FTP.

_____
_____

13.- ¿Cuál es la fecha exacta y hora aproximada en que se dieron cuenta de que se tenía un incidente de seguridad?
_____23 de Julio del 2004 a las 15:30 horas_____

14.- ¿Anterior a la fecha dada en la pregunta anterior, ¿había detectado algo anormal en el sistema?
_____No_____

Si es afirmativo, ¿Qué fue?
_____
_____

15 ¿Cuenta con políticas de respuesta a incidentes dentro de la institución?
_____No_____

Si es afirmativo, ¿Se aplicaron las políticas y qué se obtuvo de la aplicación de las mismas?
_____

16.- ¿Tiene el sistema involucrado en el incidente alguna relación de confianza con otro(s) servidores?
_____No_____

17.- ¿Realizan tareas administrativas remotamente?
_____No_____

Si es afirmativo, ¿Cuáles y con qué herramientas?
_____
_____
_____

18.- ¿Cuenta con un Server log, cuál es el IP del mismo?
_____No se cuenta con un Server log_____

19.- ¿Algún servicio o servicios se vieron afectados por el incidente?
_____Ningún servicio_____

20.- ¿Qué herramientas de seguridad tiene su sistema? (Mencione todas)
_____Ninguna_____
_____

21.- ¿Cuenta con respaldos de su sistema?
_____No_____

Si la pregunta anterior es afirmativa, ¿Cuándo fueron realizados?_____
_____

22.- ¿Cuenta con un IDS o con alguna herramienta implementada para analizar su red?
_____No_____

Si es afirmativo. ¿Tiene registros guardados a partir de la fecha en que se sospecha la intrusión?
_____

23.- ¿En estos momento el sistema se encuentra en producción o apagado?
_____Apagado_____

24.- Si está apagado, ¿De que forma fue apagado?
_____Apagado Normal_____

25.- Si está en producción, ¿Se ha reiniciado el sistema desde que ocurrió el incidente hasta ahora?
_____

26.- Si esta en producción, ¿El equipo sigue conectado a la red?
_____

27.- ¿Qué se detectó en bitácoras?
____Conexiones extrañas de FTP._____

28.- ¿Desde el momento en que se detectó el incidente se tiene resguardado lo que se detectó?
_____Si_____

29.- ¿Puede explicar a grandes rasgos lo que ha hecho por controlar el incidente, la forma en que se detectó, si ustedes han borrado algún archivo importante a partir del incidente, qué archivos modificó el intruso, etc.?
_____Se apagó el sistema_____
_____
_____

30.- ¿Qué se pretende indagar o perseguir?
____De donde provino el ataque_____

31.- ¿En qué runlevel se encuentra el sistema?
_____runlevel 3_____

22.- ¿Cuándo fue la fecha en que se instaló el sistema?
_____El 12 de Julio del 2004_____

CASO NÚMERO: 0002

ETIQUETA: V02

FECHA:25 Julio 2004

HORA: 15:00

CADENA DE CUSTODIA DE LA EVIDENCIA.

Número de personas que recolectaron la evidencia: 1

Nombre completo de quién recibe la evidencia:

Víctor Hugo Sánchez Quijada

Datos del equipo revisado:

Arquitectura: PC de escritorio en torre

Modelo:

Armada

Número de serie: sin número

Datos Técnicos: Equipo LANIX con 32 Mb de RAM. Este equipo tiene un disco duro de 5.7 Gb, el microprocesador es Pentium I a 100 MHz, la tarjeta de red es Ethernet con sistema operativo Linux Red Hat 6.2. Tiene una IP no homologada 192.168.2.7. Este equipo es armado y dispone con una unidad de CD-ROM y otra unidad de floppy.

Datos de la evidencia:

Ubicación donde se encontró la evidencia:

Laboratorio de Redes y seguridad de la División de Ingeniería Eléctrica de la Facultad de la Ingeniería

De quién se obtuvo la evidencia:

Del responsable del equipo comprometido, Jorge Acebedo Cañedo

Descripción de la evidencia:

__La evidencia es una imagen del disco duro de 5.7 GB al que se le efectuó la firma md5 de todas las particiones_____

Valores hash (cuando sea el caso):

b258f21b93e0eaa1f605dfd47d3f66f4 /boot_____

0b826f207f81bbc294bd059302a3558a /usr_____

87669b6e6939619cdbc8ff8dfba574c4 /home_____

29ac32347b0bdda0659c061b46dce9e4 /var_____

4eed1213ed2aaa48f26e3edffd8a888d /_____

c7da26612f6f7b318fb79064e2909500 swap_____

REPORTE FINAL

CASO NÚMERO: 0002

Fecha:28 Julio 2004

Analistas:

Analista 1:___Víctor Hugo Sánchez Quijada_____

Analista 2:___Oswaldo Gómez Gallardo_____

Organización Responsable:

___Área de Redes y Seguridad en Cómputo de la División de Ingeniería Eléctrica de la Facultad de Ingeniería_____

Dirección:

_____

_____

_____

Ubicación del equipo comprometido:

____Laboratorio de Redes y Seguridad en Cómputo, Primer Planta baja del Edificio Valdez Vallejo, División de Ingeniería Eléctrica_____

Responsable(s) del equipo comprometido:

____Jorge Acebedo Cañedo_____

Descripción del Caso:

El equipo comprometido tiene instalado el sistema operativo Red Hat 7.1 y cuenta con la IP 192.168.2.7; este sistema fue atacado varias veces hasta que se logró obtener el acceso no autorizado a las 09:22:47 del día 23 de Julio del 2004 vulnerando el servicio de wu-ftopd 2.6-1-16, el atacante utilizó herramientas programadas por terceros que incluyen un auto-rooter llamado autowu modificado.

La máquina fue comprometida por la IP 192.168.2.4 a las 09:22:47 horas del día 23 de Julio del 2004, también se transfirió el archivo /var/ftp/nerod.tar.gz para descomprimirlo a las 09:24:45 y se comenzó la instalación del rootkit incluido en este archivo a las 09:25:02 horas.

¿Donde se ha producido la intrusión?

El ataque se perpetuo en el laboratorio de Redes y seguridad de la División de Ingeniería eléctrica de la Facultad de Ingeniería, el equipo disponía de la IP 192.168.2.7, como se mencionó anteriormente, el intruso explotó la vulnerabilidad en el servidor wu-ftpd 2.6.1-16 con el kit de herramientas auto-rooter llamado autowu modificado; el objetivo principal de esta herramienta es localizar de forma automática servidores vulnerables wu-ftpd (<= 2.6.1), explotarlos y propagarse, utilizando cada máquina comprometida como base de operaciones.

¿Qué hizo el atacante en la máquina?

El atacante utilizó el rootKit auto-rooter para vulnerar el servicio de FTP, este rootkit tiene un archivo de instalación que deshabilita con la señal 15 el servicio de registro del syslogd/klogd a las 9:45, troyanizando algunos binarios y alterando el sistema de arranque y la configuración de los servicios de red

El intruso instaló un backdoor de SSH para crear una cuenta del mismo con un UID 0 y sin password para que el atacante pueda entrar al sistema sin necesidad de alguna clave, este backdoor más un linsniffer se gobiernan por la modificación del archivo /etc/rc.d/init.d/function hecha por el atacante al instalar el rootkit, por lo que tuvo que reiniciar el xinetd a las 09:25:15 horas. La información del /usr/local/games/tcp.log se encuentra contenido en linsniffer que está disfrasado de un archivo binario identd, además se envían de manera periódica el contenido de éste a una cuenta de correo que encontramos en el archivo /var/spool/cron/operador llamada sunsfant2x@yahoo.com, esto lo realizó el agresor creando una entrada en crontab a las 09:25:19 horas. Posteriormente este intruso crea la cuenta de usuario nerod a las 09:27:09 horas.

Dentro de las utilidades que tiene rootkit auto-rooter existe una que caducó la sesión creada en FTP a los 16 minutos, es decir a las 09:38:01.

También el atacante creó el directorio /root/., posteriormente trató de descargar un proxy para redes IRC y más archivos de auto-rooter lográndolo a la 10:16:55, un poco antes de realizar esto, el intruso deshabilita el acceso anónimo el servidor, esto lo hace a las 09:56:39

A las 10:16:55 horas con el programa wget descarga el archivo PsyBNC.tar.gz de Geocities, que descomprime y ejecuta a las 10:17:41 horas, posteriormente descarga con ese mismo programa el archivo awu.tgz de Geocites, este archivo contiene exploits para ssh y wu-ftpd, esto lo hace a las 10:41:51; al descomprimir este último archivo se crea el directorio /root/.,/aw/, ejecuta un barrido en la red a las 10:46:19 que detiene tres minutos después y finalmente limpia la pantalla con el comando clear.

El intruso ingresa al sistema a las 11:17:09 a través de su backdoor de SSHD, verifica si se encuentra alguien conectado y mueve el directorio /var/tmp a una carpeta "." creada previamente por él, posteriormente instala la aplicación IRC-BOT con el programa que ha venido usando (wget desde el servidor público de Geocities), este archivo contiene la versión 2.8 EneryMech.

Hasta ahora sospechamos que la versión de EnergyMech 2.8 contiene un puerta trasera porque al ejecutar el programa IRC-BOT aparecen los archivos /dev/hdx2 y /dev/hdx1 y los archivos binarios /bin/ping con los registros MAC-time alterados.
Detectamos que los binarios chgrp, chmod, chown, cp, cpio, dd, df, dnsdomainname, domainname, de, hostname, ln, ls, mail, mkdir, mknod, mktemp, mt, mv, netstat, nisdomainname, ping, red, setserial y ypdomainname, se encuentran infectados

El intruso borró el archivo /var/tmp/. y /manu.tgz, finalmente a las 15:36:30 horas el servidor se apaga por el administrador responsable.

Como se puede ver, el intruso utiliza herramientas creadas por terceros, tiene bases en línux y conocimientos de FTP y SSH; quizá se trata de un script Kinder que tiene tal vez intenciones de experimentar y probar ataques por curiosidad o diversión.

Recomendaciones sobre la intrusión

Lo más recomendable es actualizar la versión de Red Hat 7.1 a Rad Hat 9, también descargar e instalar actualizaciones de FTP y SSH, instalar un IDS que alerte a tiempo de posibles instrucciones o patrones anómalos y finalmente colocar un sniffer que permita monitorear el tráfico en la red.

CASO PRÁCTICO 3

CASO NÚMERO: 0003

ETIQUETA: EV03

FECHA: 16Agosto 2004

HORA: 13:00 Horas

CADENA DE CUSTODIA DE LA EVIDENCIA.

Número de personas que recolectaron la evidencia: 1

Nombre completo de quién recibe la evidencia:

Víctor Hugo Sánchez Quijada

Datos del equipo revisado:

Arquitectura: PC de escritorio en torre

Modelo: Hewlett Packard Vectra VL 400

Número de serie: MX10835425

Datos Técnicos: 128 MB en memoria RAM, Miocroprocesador Pentium III a 733 MHz, cd rom, floppy tarjeta de red Ethernet Tricom, disco duro de 5.7 GB

Datos de la evidencia:

Ubicación donde se encontró la evidencia:

Facultad de Ingeniería Unidad de Servicios de Cómputo Académico

De quién se obtuvo la evidencia:

Se obtuvo de la HoneyPot

Descripción de la evidencia:

La evidencia es el disco duro comprometido y sus imágenes de 5.7 GB al que se le efectuó la firma md5 de todas las particiones

Valores hash (cuando sea el caso):

7be7360a98acf50aa5bb7eac43cc7399 /dev/hdc7 /

4a3b6474a0db1270690e47884df8fffd /dev/hdc1 /boot

a5c2705b402cd28d1b8c8517d9274d13 /dev/hdc5 /swap

4d5c89f0eca5aa67b8a535fe9ab19ff9 /dev/hdc6 /swap

REPORTE FINAL

CASO NÚMERO: 0003

Fecha: Agosto 2004

Analistas:

Analista 1: Víctor Hugo Sánchez Quijada

Analista 2: Oswaldo Gómez Gallardo

Organización Responsable:

Área de Redes y Seguridad en Cómputo de la División de Ingeniería Eléctrica de la Facultad de Ingeniería

Dirección:

Ubicación del equipo comprometido:

Facultad de Ingeniería Unidad de Servicios de Cómputo Académico

Responsable(s) del equipo comprometido:

Víctor Hugo Sánchez Quijada

Oswaldo Gómez Gallardo

Descripción del Caso:

Se tuvo una honeypot exitosa que contaba con el sistema operativo Red Hat 7.1 (sin ninguna actualización) instalado el día 13 de agosto de 2004 el cual contaba con los siguientes servicios: ssh, sendmail, web, X11, ftp, portmap, nfs, samba, syslog. y disponía de una dirección IP. El equipo se ejecutaba en el runlevel 5 con un kernel 2.4.20.

¿Donde se ha producido la intrusión?

El 14 de agosto 2004 a las 23:38 horas vía ftp desde la ip xxx.xxx.xxx.xxx. se trató de infiltrar al equipo, siendo comprometido a las 23:46:52.

¿Qué hizo el atacante en la máquina?

El intruso verificó que no existiera ningún sniffer en la honeypot, al no encontrar ninguno, descargó el archivo “so.tgz” a las 23:47:46. Lo “destareo” creándose el directorio “super” a las 23:48:02 el cual contiene sus herramientas. Entró al directorio y ejecutó el script “inst” para instalar el rootkit “Suckit”. Posteriormente ejecutó el sniffer además de descargar otro archivo tar “w00t.tgz” que contiene un escaneador de puertos y un exploit para el servicio de “samba”. El intruso, ocultando su presencia realizó una conexión a otro sistema con la IP xxx.xxx.xxx.xxx a las 00:25:42 del día 15 de agosto 2004; después descargó la herramienta “john the ripper” a las 00:30:26 del día 15 de agosto 2004 para apoderarse de passwords. Este paquete lo “destareó” y posteriormente lo ejecutó sobre el “/etc/shadow” de la honeypot.

El día 15 de agosto 2004 a las 00:45:54 descargó el archivo “psyBNC2.3.2-4.tar.gz” que es un proxy de IRC; lo “destareo”, lo compiló y lo instaló dentro del sistema para dejarlo ejecutándose en el equipo. Después, sólo hubo modificaciones de archivos de registro durante el resto del día.

El día 16 de agosto 2004 a las 06:55:20, el atacante regresó haciendo una conexión de entrada y salida, posteriormente el intruso accedió a la honeypot a las 08:38:27 para crear el usuario “cmi” con uid y gid igual a cero. Este usuario tenía como shell /bin/bash y como directorio personal “/home/cmi”, después el intruso se logueó desde otra maquina con ese loguin para verificar que si podía entrar a la honeypot con esa cuenta con privilegios de administrador.

El día 16 de agosto 2004 a las 08:09:44, descargó el paquete “emech-2.8.tar.gz” que es un cliente IRC para conectarse a canales de chat, lo instaló dentro del directorio oculto creado por él mismo: “/usr/lib/.sysinit/”.

El día 16 de agosto del 2004 a las 12:21 horas, finalmente nosotros decidimos dar de baja el equipo.

Recomendaciones sobre la intrusión.

Nuestra HoneyPot nos reveló información interesante acerca de las medidas preventivas que se debe tener para un servidor de este tipo, creemos que las principales deficiencia de seguridad son: el no tener actualizado el sistema con los últimos parches, esto hubiera hecho al equipo menos vulnerable a exploits, también el tener varios puertos abiertos representa un riesgo, se debe hacer una evaluación profunda y exhaustiva de los servicios que se pretenden ofrecer, para que de esta manera sólo se dejen los que sí se utilizan, ya que el tener muchos puertos abiertos hace al sistema más susceptible a intrusiones.

APÉNDICE D
COMANDOS UTILIZADOS EN EL ANÁLISIS FORENSE

Tabla D.1 Comandos utilizados en el análisis forense

Comando	Descripción
ls	Lista el contenido de un directorio.
grep	Despliega líneas en base a un patrón dado.
sort	Ordena líneas de un archivo de texto.
more	Desplegar el contenido de un archivo en forma paginada.
dmesg	Muestra los mensajes acaecidos durante el proceso de arranque
lspci	Lista todos los dispositivos pci
rpm	Manejador de paquetes.
dd	Copia de archivos bit a bit
mount	Puede *montar* diferentes dispositivos de almacenamiento
md5sum	Obtiene firmas digitales de archivos por el método md5
tar	Agrupa uno o más archivos en un llamado *paquete*
gzip	Compríme archivos en un formato zip
netcat	Es una utilidad simple de Unix que lee y escribe datos a través de la red usando protocolos de conexión TCP o UDP
scp	Hace una copia segura a través de servidores
sftp	Ejecuta transferencias seguras de datos
ssh	Nos permite conectarnos a una computadora remota además de proveernos de una comunicación segura encriptada
ps	Muestra información sobre los procesos
lsof	Nos sirve para listar los procesos que se encuentran abriendo un determinado archivo, directorio, socket, etc
Top	Obtiene una lista rápida de todos los *Procesos* que se estén ejecutando
netstat	Muestra conexiones de red originadas o recibidas por el Host, tablas de ruteo y estadísticas de red
nmap	Realiza un mapeo de puertos
ifconfig	Permite configurar una interfase de Red y ver el "status" de ésta
route	Despliega la tabla de ruteo del Host
whois	Para obtener la información disponible en las bases de datos WHOIS de dominios de internet
iptables	Nos permite configurar un Firewall de forma que tengamos controlado quien entra, sale y/o enruta a través de nuestra maquina Linux
nslookup	Con esta herramienta puede dirigir preguntas al servicio de información correspondiente
ethereal	Es un comando que nos permite analizar el tráfico de red para sistemas operativos linux
find	Busca un archivo en el sistema
file	Muestra el tipo de un archivo
ldd	Imprime las dependencias de las librerías

debugfs	Es un depurador del sistema de archivos ext2
fsgrab	Obtiene bloques de un archivo de sistema ext2 en linux
e2recover	Herramienta automatizada para restauración en ext2
cat	Visualiza la información contienda en un archivo
vi	Abre un editor de texto
fdisk	Es un comando, capaz de trabajar y crear particiones tanto para Linux como otros sistemas operativos
du	Da el espacio total de almacenamiento utilizado por todos los archivos del directorio en el que nos encontremos
df	Da una estadística sobre el espacio total, el ocupado y el libre de todas las unidades de disco montadas
who	Muestra las sesiones de los usuarios que están conectados en un momento dado
finger	Muestra la información de un usuario del sistema
last	Muestra una lista de los últimos usuarios logueados
cp	Copia archivos
chmod	Cambia permisos a los archivos
echo	Despliega un mensaje en pantalla
env	Corre un programa que modifica las variables de entorno
uptime	Hora actual, tiempo que lleva el sistema corriendo desde su último reinicio.
adduser	Crea usuarios al sistema
umount	Con este comando se retira una unidad de disco del sistema de archivos
ping	Comprueba que las funciones básicas de una red TCP/IP funcionan correctamente
reboot	Reinicia el sistema
strace	Herramienta para la depuración de problemas de conexión
ltrace	Inspecciona las llamadas a bibliotecas en tiempo de ejecución en programas enlazados dinámicamente. ltrace es un programa de depuración que corre un comando especificado hasta que éste termina.
strings	Imprime las cadenas de la tabla de caracteres en archivos
gdb	Comando que permiten ejecutar paso a paso un código (debugger), es un comando de depuración
apptrace	Es el comando que inspecciona las llamadas a librerías

host	Comando que convierte nombres de IP a direcciones o viceversa.
grave-robber	Herramienta para captura de datos que automatiza la obtención de información del sistema.
lls	Lista la información de inodos. Es muy útil para recolectar inodos de archivos borrados.
pcat	Copia la memoria de procesos de un sistema vivo. Puede usarse para recolectar información sobre procesos en ejecución. Obtiene la identificación de un proceso sospechoso.
icat	Se usa sobre una imagen para recuperar archivos borrados.
unrm	Recolecta información en lugares no asignados del sistema de archivos.
lazarus	Analiza los datos crudos generados por unrm, espacio de swap. Intenta identificar bloques de disco.
mactime	Reporta fechas de MAC. Intenta determinar qué archivos han sido accedidos o modificados durante un periodo de tiempo dado.
Sleuthkit	Colección de herramientas y extensiones que utilizan the coroner toolkit proporcionando mayor funcionalidad.
Autopsy	Autopsy envuelve en una GUI a Sleuthkit basado en un navegador fácil de usar. Hace posible el análisis de la imagen a nivel de archivo, bloques e inodo.
dcat	Despliega el contenido de un bloque de disco.
dls	Lista el contenido de bloques de disco borrados.
dcalc	Mapea entre imágenes dd y los resultados de dls.
dstat	Lista estadísticas asociadas a bloques de disco específicos.
istat	Despliega información sobre un inodo.
icat	Despliega el contenido de bloques de disco asignados a un inodo.
ifind	Determina qué inodo tiene asignado un bloque en una imagen.
fls	Despliega entradas de archivo y de directorio en un inodo de directorio.
ffind	Determina qué archivo tiene asignado un inodo en una imagen.
fsstat	Despliega los detalles del sistema de archivo en ASCII.

APÉNDICE E
CRIPTOGRAFÍA

Encriptación

La encriptación es el método por el cual la información puede ser protegida a través de programas encargados de cifrarla, firmarla para identificar eficientemente al remitente, y cerrarla para que sólo pueda ser abierta nuevamente por quien tenga la clave apropiada, agregándose además métodos para corroborar la integridad de la información recibida, es decir, para validar que la información no haya sido modificada en el camino. Encriptación es solamente una parte del campo de la seguridad informática llamado "**criptología**". La criptología se conforma de dos amplias áreas: la "**criptografía**" que se dedica a la construcción y operación de sistemas de seguridad informática; el "**criptoanálisis**" cuyo fin es desencriptar los criptogramas y acceder de manera ilegítima a la información contenida en los mensajes mediante ciertas técnicas con las que no es preciso conocer la clave de cifrado.

Criptografía simétrica.

Los métodos criptográficos tradicionales operan a partir de una palabra o frase llave, que sirve para codificar y decodificar información, el conocido **password.** Esta llave debe ser conocida por los extremos de la comunicación, por lo que el punto débil de este método es justamente el proceso de difusión de la llave.

Criptografía asimétrica.

Por el contrario, la criptología de clave pública (Kpu) consiste en poner a cada extremo de la comunicación un par de llaves, una pública que cualquiera puede solicitar y conocer, y otra privada (Kpr), cuya seguridad es fundamental para el éxito de la codificación. Las llaves son una secuencia bastante larga de caracteres y números, generadas por un procedimiento matemático. Para enviar un mensaje a una persona, se codifica con la clave pública del destinatario. El sistema garantiza que el mensaje resultante sólo puede ser decodificado con la clave privada del destinatario (confidencialidad). Como se tiene la seguridad de la identidad del destinatario gracias a su clave pública, nos aseguramos que el mensaje va al sitio correcto (autenticación).

Para enviar un mensaje firmado, se genera una "firma digital" del mismo (con unos algoritmos matemáticos que proporcionan un resumen del mensaje), que se codifica

con la clave privada del remitente. Posteriormente, el receptor puede utilizar la clave pública del remitente para verificar su origen; de esta forma se garantiza que el mensaje sólo puede ser enviado por el remitente (no repudio), ya que él es el único que conoce su clave privada.

DES (Data Encryption Standar)

Este sistema fue desarrollado a principio de los años '70 por un grupo de trabajo de IBM. En 1981 la ANSI aprovó el DES como estándar, el X3.92. Por su parte, la ISO hizo lo mismo en 1987 dándole el nombre de DEA-1. Las principales características de este sistema de cifrado es que utiliza operaciones lógicas simples (transposiciones, desplacamientos y XOR's) sobre grupos reducidos de bits, lo que permite una fácil y eficiente implementación del algoritmo en hardware. El algoritmo toma la información en bloques de 64 bits produciendo un bloque de texto cifrado también de 64 bits. Las claves utilizadas por este sistema son de 56 bits, aunque se suelen distribuir en forma de un número de 64 bits, donde cada octavo bit (el lsb o less-significant bit) de cada uno de los ocho bytes de la calve es un bit de paridad.

RSA (Rivest, Shamir y Adleman)

La idea de un sistema de clave pública fue planteada por Diffie y Hellman en 1976 y realizada por primera vez en 1977 por Rivest, Shamir y Adleman quienes inventaron el criptosistema conocido RSA. Para describir la manera de trabajo del RSA, se requiere dos conceptos fundamentales, los cuales se plantean a continuación.

Uno de los algoritmos de clave publica mas ampliamente utilizado fue inventado por un grupo de investigadores del MIT, Rivest, Shamir y Adelmar, el nombre del algoritmo esta formado por sus iniciales RSA. Este criptosistema, realiza todos sus cálculos en n donde n=pq, con p, q primos distintos, y la relación siguiente se cumple: (n) = (p-1)(q-1)

Definición

Sea n = pq, donde p y q son primos.

Sea P = C = n y K se define como K = {(n,p,q,a,b) : n=pq, p, q primos, ab"1 (mod (n)) }

Para K=(n,p,q,a,b) se define las funciones de encriptación y decapitación como:

ek(x) = xb mod n

y dk(y) = ya mod n

donde x, y "n.

Los valores de n y b son públicos, y los valores de p, q, a son secretos.

Verifiquemos si las funciones de encriptacion y decriptación son inversos. Se tiene que:

ab " 1 (mod (n))

luego ab = t (n) + 1 para todo entero t " 1.

supongamos que x"n*, luego tenemos:

(xb)a " x t (n) + 1 (mod n)

" (x (n)) t x (mod n)

" 1t x (mod n)

" x (mod n)

DSA ('Digital Signature Algorithm'), DH ('Diffie-Hellman'), ElGamal y Nyberg-Rueppel

Basan su seguridad en el problema del logaritmo discreto del grupo multiplicativo de un campo finito. La última familia de criptografía asimétrica basa su seguridad también en el problema del logaritmo discreto, pero sobre el grupo de puntos racionales de una curva elíptica sobre un campo finito. Se trata de los mismos algoritmos que en el caso de la segunda familia, pero en versión elíptica. La criptografía asimétrica está dedicada principalmente a resolver el problema de autenticidad y no rechazo, y esto se logra con la firma digital. También es utilizada para el intercambio de claves simétricas.

GLOSARIO

ActiveX

ActiveX es una de las tecnologías más potentes que ha desarrollado Microsoft. Mediante ActiveX es posible reutilizar código, descargar código totalmente funcional de un sitio remoto, etc. Esta tecnología es considerada la respuesta de Microsoft a Java.

ActiveX soluciona los problemas de seguridad mediante certificados y firmas digitales. Una Autoridad Certificadora (AC) expende un certificado que acompaña a los controles activos y una firma digital del programador.

Cuando un usuario descarga una página con un control, se le preguntará si confía en la AC que expendió el certificado y/o en el control ActiveX. Si el usuario acepta el control, éste puede pasar a ejecutarse sin ningún tipo de restricciones (sólo las propias que tenga el usuario en el sistema operativo). Es decir, la responsabilidad de la seguridad del sistema deja en manos del usuario, ya sea este experto cibernauta consciente de los riesgos que puede acarrear la acción o un perfecto novato en la materia.

Esta última característica es el mayor punto débil de los controles ActiveX ya que la mayoría de los usuarios aceptan el certificado sin siquiera leerlo, pudiendo ser esta la fuente de un ataque con un control dañino.

La filosofía ActiveX es que las Autoridades de Certificación se fían de la palabra del programador del control. Es decir, el programador se compromete a firmar un documento que asegura que el control no es nocivo. Evidentemente siempre hay programadores con pocos escrúpulos o con ganas de experimentar.

Así, un conocido grupo de hackers alemanes, desarrolló un control ActiveX maligno que modificaba el programa de Gestión Bancaria Personal Quicken95 de tal manera que si un usuario acepta el control, éste realizaba la tarea que supuestamente tenía que hacer y además modificaba el Quiken, para que la próxima vez que la víctima se conectara a su banco, se iniciara automáticamente una transferencia a un cuenta del grupo alemán.

Otro control ActiveX muy especialmente malévolo es aquel que manipula el código de ciertos exploradores, para que éste no solicite confirmación al usuario a la hora de

descargar otro control activo de la Web. Es decir, deja totalmente al descubierto, el sistema de la víctima, a ataques con tecnología ActiveX.

Archivos ejecutables (Virus ExeVir)

El virus se adosa a un archivo ejecutable y desvía el flujo de ejecución a su código, para luego retornar al huésped y ejecutar las acciones esperadas por el usuario. Al realizarse esta acción el usuario no se percata de lo sucedido. Una vez que el virus es ejecutado se aloja en memoria y puede infectar otros archivos ejecutables que sean abiertos en esta máquina.

En este momento su dispersión se realiza en sistemas de 16 bits (DOS) y de 32 bits (Windows) indistintamente, atacando programas .COM, .EXE, .DLL, .SYS, .PIF, etc., según el sistema afectado.

Backdoors

Las puertas traseras con trozos de código en un programa que permiten a quien las conoce saltarse los métodos usuales de autenticación para realizar ciertas tareas. Habitualmente son insertados por los programadores del sistema para agilizar la tarea de probar código durante la fase de desarrollo.

Esta situación se convierte en una falla de seguridad si se mantiene, involuntaria o intencionalmente, una vez terminado el producto ya que cualquiera que conozca el agujero o lo encuentre en su código podrá saltarse los mecanismos de control normales.

Barreras Infrarrojas y de Microondas

Transmiten y reciben haces de luces infrarrojos y de microondas respectivamente. Se codifican por medio de pulsos con el fin de evadir los intentos de sabotaje. Estas barreras están compuestas por un transmisor y un receptor de igual tamaño. Cuando el haz es interrumpido, se activa el sistema de alarma y luego vuelve al estado de alerta. Estas barreras son inmunes a fenómenos aleatorios como calefacción, luz ambiental, vibraciones, movimiento de masas de aire, etc.

Borrado de Huellas

El borrado de huellas es una de las tareas más importantes que deben realizar el intruso después de ingresar en un sistema, ya que, si se detecta su ingreso, el administrador buscará como conseguir “tapar el hueco” de seguridad, evitar futuros ataques e incluso rastrear al atacante.

Las huellas son todas las tareas que realizó en intruso en el sistema y por lo general son almacenadas en Logs (archivo que guarda la información de lo que se realiza en el sistema) por el sistema operativo.

Los archivos Logs son una de las principales herramientas (y el principal enemigo del atacante) con las que cuenta un administrador para conocer los detalles de las tareas realizadas en el sistema y la detección de intrusos.

Bouncer

Es un programa que se ejecuta en una máquina de red y que sirve para hacer de puente o proxy entre el usuario y el servidor de IRC, haciendo uso de la IP del servidor donde se encuentra este bouncer. Es decir, que la IP del usuario se enmascara por la del servidor.

Bucaneros

Son comerciantes sucios que venden los productos Crackeados por otros, generalmente comercian con tarjetas de crédito y de acceso que compran a los copyHackers. Son personas sin ningún conocimiento de electrónica o informática.

Carding

Es el uso ilegítimo de las tarjetas de crédito pertenecientes a otras personas con el fin de obtener los bienes realizando fraude con ellas. Se relacionan mucho con el Hacking o el Cracking mediante los cuales consiguen los números de las tarjetas digitales.

Connection Flood

La mayoría de las empresas que brindan servicios de Internet (ISP) tienen un límite máximo en el número de conexiones simultáneas. Una vez que se alcanza ese límite, no se admitirán conexiones nuevas. Así, por ejemplo un servidor Web puede tener capacidad para atender a mil usuarios simultáneos. Si un atacante establece mil conexiones y no realiza ninguna petición sobre ellas, monopolizará la capacidad del servidor. Las conexiones van caducando por inactividad poco a poco, pero el atacante sólo necesita intentar nuevas conexiones, (como ocurre en el caso de SYN Flood) para mantener fuera de servicio el servidor.

Creadores de Virus

Personas que se encargan de hacer programas con el fin de dañar un sistema, su meta es infectar y dañar la mayor cantidad de equipos.

Decoy (Señuelos)

Los Señuelos son programas diseñados con la misma interfase que otro original. En ellos se imita la solicitud de un logueo y el usuario desprevenido lo hace. Luego, el programa guardará esa información y dejará paso a las actividades normales del sistema. La información recopilada será utilizada por el atacante para futuras visitas.

Una técnica semejante es aquella que, mediante un programa se guardan todas las teclas presionadas durante la sesión. Luego se estudia el archivo generado para conocer nombre de usuarios y claves.

Detector de ultrasonido

Este equipo utiliza ultrasonidos para crear un campo de ondas. De esta manera, cualquier movimiento realice un cuerpo dentro del espacio protegido, generará una perturbación en dicho campo que activará la alarma. Este sistema posee un circuito refinado que eliminará las falsas alarmas.

Detectores pasivos sin alimentación

Estos elementos no requieren alimentación extra de ningún tipo, solo van conectados a la central de control de alarmas para mandar la información de control.

Diccionarios

Los diccionarios son archivos con millones de palabras, las cuales pueden ser posibles passwords de los usuarios. Este archivo es usado para descubrir dicho password en pruebas de fuerza bruta.

El programa encargado de probar cada una de las palabras encripta cada una de ellas, mediante un algoritmo utilizado por el sistema atacado, y compara la palabra encriptada contra el archivo de passwords del sistema atacado (previamente obtenido). Si coinciden se ha encontrado la clave de acceso al sistema, mediante el usuario correspondiente la clave hallada.

Actualmente es posible encontrar diccionarios de gran tamaño orientados, incluso, a un área específico de acuerdo al tipo de organización que se este atacando.

E-mail Bombing-Spamming

El e-mail Bombing consiste en enviar muchas veces un mensaje idéntico a una misma dirección saturando así el mailbox del destinatario.

El spamming, en cambio se refiere a enviar un e-mail a miles de usuarios hayan estos solicitado el mensaje o no. Es muy utilizado por las empresas para publicar sus productos.

El Spamming esta siendo actualmente tratado por las leyes europeas como una violación a los derechos de privacidad del usuario.

Emisión de Calor

Se mide la emisión de calor del cuerpo (termograma), realizando un mapa de valores sobre la forma de cada persona.

Exploits

Es muy frecuente ingresar a un sistema explotando agujeros en los algoritmos de encriptación utilizados, en la administración de las claves por parte de la empresa o simplemente encontrando un error en los programas utilizados.

Los programas para explorar estos agujeros reciben el nombre de Exploits y lo que realizan es aprovechar la debilidad, fallo o error hallado en el sistema (hardware o software) para ingresar al mismo.

Nuevos Exploits (explotando nuevos errores en los sistemas), se publican cada día por lo que mantenerse informado de los mismos y de las herramientas para combatirlos es de vital importancia.

Fragmentation Scanning

Esta no es una técnica de scaneo como tal, si no una modificación de las anteriores. En lugar de enviar paquetes completos de sondeo los mismos se particionan en un par de pequeños fragmentos IP. Así se logra partir una cabecera IP en distintos paquetes para hacerlo más difícil de monitorizar por los filtros que puedan estar ejecutándose en la máquina objetivo.

Sin embargo, algunas implementaciones de estas técnicas tienen problemas con la gestión de este tipo de paquetes tan pequeños, causando una caída de rendimiento en el sistema del intruso o en el de la víctima. Problemas de esta índole convierten en detectables a este tipo de paquetes.

Ftp

(*File Transfer Protocol*, Protocolo de Transferencia de Archivos) Protocolo (parte de la arquitectura TCP/IP) utilizado para la transferencia de archivos.

Gurús

Son considerados como los maestros y los encargados de formar a los futuros Hackers. Generalmente no están activos y son reconocidos por la importancia de sus hackeos, de los cuales solo enseñan las técnicas básicas.

Hoax, Virus Fantasma

El auge del correo electrónico generó la posibilidad de transmitir mensajes de alerta de seguridad. Así comenzaron a circular mensajes de distinta índole (virus, cadenas solitarias, beneficios, catástrofes, etc.) de casos inexistentes. Los objetivos de estas alertas pueden causar alarma, la pérdida de tiempo, el robo de direcciones de correo y la saturación de los servidores con las consecuentes pérdidas que esto ocasiona.

Http

(*HyperText Transfer Protocol)* Protocolo utilizado por los servidores de Web para la visualización de páginas**.**

Huella digital

Basado en el principio de que no existen dos huellas dactilares iguales. Cada huella posee pequeños arcos, ángulos, bucles, remolinos, etc. (llamados minucias) características y la posición relativa de cada una de ellas es lo analizado para establecer la identificación de una persona. Esta aceptado que dos personas no tienen ocho minucias iguales y cada una posee más de 30, lo que hace al método sumamente confiable.

Imap

(*Internet Message Access Protocol*, Protocolo de Acceso a Mensajes de Internet) Protocolo diseñado para permitir la manipulación de mailboxes remotos como si fueran locales. IMAP requiere de un servidor que haga las funciones de oficina de correos pero en lugar de leer todo el mailbox y borrarlo, solicita sólo los encabezados de cada mensaje. Se pueden marcar mensajes como borrados sin suprimirlos completamente, pues estos permanecen en el mailbox hasta que el usuario confirma su eliminación.

Información

La información es el conjunto de datos que tiene un significado específico más allá de cada uno de éstos, y tendrá un sentido particular según cómo y quién la procese y la interprete. La información es cualquier mensaje (conjunto de datos) que le interese al receptor, entienda o lo ignore antes de recibirlo.

Insider

Pueden ser empleados disconformes o personas externas con acceso a sistemas dentro de la empresa u organización, los cuales utilizan sus permisos para alterar archivos o registros.

IRC-BOT

Es un programa que se conecta a IRC como cualquier usuario normal, la diferencia es que normalmente se ejecuta desde sistemas que están conectados a Internet las 24 horas del día y con un ancho de banda superior a la de conexión telefónica normal. Un IRC-BOT puede ayudar enormemente al control de uno o varios canales, protegiéndolos y manteniéndolos abiertos por más tiempo que lo que cualquier persona podría hacer. Un IRC-BOT puede concentrar la información de operadores y usuarios de canales y hacer gran cantidad de funciones, ya que se le pueden añadir scripts o rutinas pre-hechas. Definición: Equipo desarrollo EggDrop IRC-BOT.

IRC

El IRC, acrónimo de Internet Relay Chat, es un sistema de conversación en tiempo real para usuarios de Internet. Para poder participar solo es necesario disponer de un programa cliente de IRC y una conexión a Internet. El IRC permite que múltiples usuarios se reúnan simultáneamente en tertulias o debates, en los cuales cada uno va expresando sus opiniones de forma escrita y en tiempo real. Definición: Equipo IRC Hispano

Jamming o Flooding

Este tipo de ataques desactivan o saturan los recursos. Por ejemplo, un atacante puede consumir toda la memoria o espacio en disco disponible, así que como enviar tanto tráfico a la red que nadie más pueda utilizarla.

Aquí el atacante satura el sistema con mensajes que requieren establecer conexión. Sin embargo, en vez de proveer la dirección IP, el mensaje contiene falsas direcciones IP usando Spoofing y Looping. El sistema responde al mensaje, pero no recibe respuesta, acumula buffers con información de las conexiones abiertas, no dejando lugar a las conexiones legítimas.

Muchos ISPs (proveedores de Internet) han sufrido bajas temporales del servicio por ataques que explotan el protocolo TCP. Muchos host de Internet han sido dados de baja por “el ping de la muerte”.

Mientras que el ping normal simplemente verifica si un sistema esta enlazado a la red, el ping de la muerte causa bloqueo instantáneo del equipo. Esta vulnerabilidad ha sido ampliamente utilizada en el pasado pero, aún hoy pueden encontrarse sistemas vulnerables.

Otra acción común es la de enviar millares de e-mails sin sentido a todos los usuarios posibles en forma continua, saturando los sistemas destinos.

Java Applets

Java es un lenguaje de programación interpretado, desarrollado inicialmente por la empresa SUN. Su mayor popularidad la merece por su alto grado de seguridad. Los más usados navegadores actuales, implementan Máquinas Virtuales Java (MVJ) para ser capaces de ejecutar programas (Applets) de Java.

Estos Applets, al fin y al cabo, no son más que código ejecutable y como tal, susceptible de ser manipulado por intrusos. Sin embargo, partiendo del diseño, Java siempre ha pensado en la seguridad del sistema. Las restricciones a las que somete a los Applets son de tal envergadura (imposibilidad de trabajar con archivos a no ser que el usuario especifique lo contrario, imposibilidad de acceso a zonas de memoria y disco

directamente, firma digital, etc.) que es muy difícil de lanzar ataques. Sin embargo existe un grupo de expertos especializados en descubrir fallas de seguridad en las implementaciones de las MVJ.

JavaScript y VBScript

JavaScript (de la empresa Netscape) y VBScript (de Microsoft) son los lenguajes usados por los diseñadores de sitios Web para evitar el uso de Java. Los programas realizados son interpretados por el navegador.

Auque los fallos son muchos más numerosos en versiones antiguas de JavaScript, actualmente se utilizan para explorar vulnerabilidades específicas de navegadores y servidores de correo ya que no se realizan ninguna evaluación sobre el código.

Land Attack

Este ataque consiste en un Bug (error) en la implementación de la pila TCP/IP de las plataformas Windows.

El ataque consiste en mandar algún puerto abierto de un servidor (generalmente al NetBIOS 113 ó 139) un paquete, maliciosamente construido, con la dirección y puerto origen igual a que la dirección y puerto destino.

Por ejemplo se envía un mensaje desde al dirección 10.0.0.1:139 hacia ella misma. El resultado obtenido es que luego de cierta cantidad de mensajes enviados-recibidos la máquina termina colgándose.

Existen variantes a este método consistente, por ejemplo, enviar el mensaje a una dirección específica sin especificar el puerto.

Macro virus

Estos virus infectan archivos de información generados por aplicaciones de oficina que cuentan con lenguajes de programación macros. Últimamente son los más expandidos, ya que todos los usuarios necesitan hacer intercambio de documentos para realizar su trabajo. Los primeros antecedentes de ellos fueron con las macros de Lotus

123 que ya eran lo suficientemente poderosas como permitir este tipo de implementación. Pero los primeros de difusión masiva fueron desarrollados a principios de los 90's para el procesador de texto Microsoft Word, ya que este cuenta con el lenguaje de programación Word Basic.

Su principal punto fuerte fue que terminaron con un paradigma de la seguridad informática: "los únicos archivos que pueden infectarse son los ejecutables" y todas las tecnologías antivirus sucumbieron ante este nuevo ataque.

Su funcionamiento consiste en que si una aplicación abre un archivo infectado, la aplicación (o parte de ella) se infecta y cada vez que se genera un nuevo archivo o se modifique uno existente contendrá el macrovirus.

MD5 con clave

Dos usuarios se ponen de acuerdo para compartir una clave secreta k. El emisor entonces aplica MD5 a la concatenación del mensaje (m) con esta llave. La clave k es borrada del mensaje una vez MD5 ha finalizado. Lo que el emisor envía es:

$$m + MD5(m + k)$$

El receptor del mensaje aplica MD5 a la concatenación del mensaje con la clave secreta k. Si el resultado coincide con el checksum enviado con el mensaje, entonces el mensaje debe haber sido enviado por el usuario que tiene la clave. Este método utiliza una clave de encriptación simétrica.

Net Flood

En estos casos la red víctima no puede hacer nada. Aunque filtre el tráfico en sus sistemas, sus líneas estarán saturadas como tráfico malicioso, incapacitándolas para cursar tráfico útil.

Un ejemplo habitual es el de un teléfono: si alguien quiere molestar, solo tiene que llamar, de forma continua. Si se descuelga el teléfono (para que deje de molestar), tampoco se puede recibir llamadas de otras personas. Este problema es habitual, por

ejemplo cuando alguien intenta mandar un fax empleando el número de voz; el fax insiste durante horas, sin que el usuario llamado pueda hacer nada al respecto.

En el caso de Net Flooding ocurre algo similar. El atacante envía tantos paquetes de solicitud de conexión que las conexiones auténticas simplemente no pueden competir.

En casos así el primer paso a realizar es ponerse en contacto con el Proveedor del servicio para que intente determinar la fuente del ataque y, como medida provisional, filtre el ataque en su extremo de la línea.

El siguiente paso consiste en localizar las fuentes del ataque e informar a sus administradores, ya que seguramente se estará usando sus recursos sin su conocimiento y consentimiento. Si el atacante emplea IP Spoofing, el rastreo puede ser casi imposible, ya que en muchos casos la fuente del ataque es, a su vez, víctima y el origen último puede ser prácticamente imposible de determinar (looping).

Nfs

(Network FileSystem) Sistema de archivos en red.

OOB, SupernuKe o WinNuke

Un ataque característico o quizás el más común, de los equipos con Windows es el Nuke, que hace que los equipos que escuchan el puerto NetBIOS sobre TCP/UDP 137 a 139, queden fuera de servicio, o disminuyan su rendimiento al enviarle paquetes UDP manipulados.

Generalmente se envían fragmentos de paquetes Out Of Band, que la máquina víctima detecta como inválidos pasando a un estado inestable. OOB es el término normal, pero realmente consiste en configurar el bit Urgente (URG) en los indicadores del encabezamiento TCP, lo que significa que este bit es válido.

Este ataque puede prevenirse instalando los parches adecuados suministrados por el fabricante del sistema operativo afectado. Un filtro efectivo debería garantizar la detección de una inundación de bits Urgentes.

Outsider.

Son personas que atacan desde fuera de la ubicación física de la organización, estas personas ingresan a la red simplemente averiguando una contraseña válida.

Phreakers

El Phreaking es la actividad mediante la cual algunas personas con ciertos conocimientos de herramientas de hardware y software pueden engañar a las compañías telefónicas para que estas no cobren las llamadas que hacen. Los Phreakers son Crackers de redes de comunicación con amplios conocimientos en telefonía.

Proxy

Un proxy es una aplicación o un dispositivo hardware que hace de intermediario entre los usuarios, normalmente de una red local, e Internet. Lo que hace realmente un proxy es recibir peticiones de usuarios y redirigirlas a Internet.

Samba

Conjunto de programas que permite a UNIX y a los sistemas afines a UNIX comunicarse con host Microsoft.

Samurai

Son lo más parecido a una amenaza pura. Sabe lo que busca, donde encontrarlo y como lograrlo. Hacen su trabajo por encargo a cambio de dinero. Estos personajes a diferencia de los anteriores, no tienen conciencia de la comunidad y no forman parte de los clanes reconocidos por los hackers. Se basan en el principio de que cualquiera puede ser atacado y saboteado solo basta que alguien lo desee y tenga el dinero para pagarlo.

Scaning (Búsqueda)

El Scaneo como método de descubrir canales de comunicación susceptibles de ser explotados, lleva en uso mucho tiempo. La idea es recorrer (scanear) tantos puertos de escucha como sea posible, y guardar información de aquellos que sean receptivos o de

utilidad para cada necesidad en particular. Muchas utilidades de auditoria también se basan en este paradigma.

Scanear puertos implican las mismas técnicas de fuerza bruta. Se envía una seria de paquetes para varios protocolos y se deduce que servicios están escuchando por las respuestas recibidas y no recibidas. Existen diversos tipos de scanning según las técnicas, puertos y protocolos explotados.

Sendmail

Agente de transporte de correo electrónico.

Shoulder Surfing

Consiste en espiar físicamente a los usuarios para obtener el login y el password correspondiente. El Surfing explora el error de los usuarios de dejar su login y su password anotados cerca de la computadora.

Cualquier intruso puede pasar por ahí verlos y copiarlos para su posterior uso. Otra técnica relacionada al Surfing es aquella mediante la cual se ve, por encima del hombro, al usuario cuando teclea su login y su password.

Sistema Biométrico

Definimos a la biometría como la parte de la biología que estudia en forma cuantitativa la variabilidad individual de los seres vivos utilizando métodos estadísticos.

La Biometría es una tecnología que realiza mediciones en forma electrónica, guarda y compara características únicas para la identificación de personas.

La forma de identificación consiste en la comparación de características físicas de cada persona con un patrón conocido y almacenado en una base de datos. Los lectores biométricos identifican a la persona por lo que es.

Smurf o BroadCast Storm

Este ataque es bastante simple y a su vez devastador. Consiste en recolectar una serie de direcciones BroadCast para, a continuación, mandar una petición ICMP (simulando un Ping) a cada una de ellas en serie, varias veces, falsificando la dirección IP de origen (máquina víctima).

Este paquete maliciosamente manipulado, será repetido en difusión (BroadCast), y cientos o miles de host mandarán una respuesta a la víctima cuya dirección IP figura en el paquete ICMP.

Suponiendo que se considere una red tipo C la dirección Broad Cast sería .255; por lo que el simple envío de un paquete se convierte en un efecto multiplicador devastador.

Desgraciadamente la víctima no puede hacer nada para evitarlo. La solución está en manos de los administradores de red, los cuales deben configurar adecuadamente sus Routers para filtrar los paquetes ICMP de peticiones indeseados (BroadCast); o bien configurar sus máquinas para que no respondan a dichos paquetes. Es decir, que lo que se parchea son las máquinas/redes que puedan actuar como intermediarias (inocentes) en el ataque y no la víctima.

También se podría evitar el ataque si el Router/Firewall de salida del atacante estuviera convenientemente configurado para evitar Spoofing. Esto se haría filtrando todos los paquetes de salida que tuvieran una dirección de origen que no perteneciera a la red interna.

Sniffer

Un sniffer es un programa o dispositivo (puede ser un elemento de hardware, no necesariamente tiene que ser un programa), que analiza un determinado punto de la red con fines muy diversos. Un sniffer "olisquea", analiza el tráfico de datos que pasa por punto de la red en la que está instalado.

Socket

Todos hemos oído hablar de la palabra winsock, winsockets, o socket a secas. Tiene una mala traducción: al pie de la letra es "enchufe", y en algunos sitios, lo traducen por conexión, canal, o palabras similares....

Vamos a intentar explicar qué es un socket (vamos a respetar el término anglosajón). Un socket no es nada más que la combinación de una máquina y un puerto, con otra máquina (que puede ser incluso la misma) y otro puerto.

Sonorización de dispositivos luminosos

Dentro de los elementos de sonorización se encuentran las sirenas, campanas, timbre, etc. Algunos dispositivos luminosos son los faros rotativos, las balizas, las luces intermitentes, etc. Estos deben estar colocados de modo que sean efectivamente oídos o vistos por aquellos a quienes están dirigidos. Los elementos de sonorización deben estar bien identificados para poder identificar si el estado de alarma es de robo, intrusión, asalto o aviso de incendio.

Ssh

(*Secure Shell*) Es una aplicación de seguridad que permite la conexión entre computadoras de forma segura, a través de un demonio sshd.

SYN Flood

Es el más famoso de los ataques del tipo Denial of Service, se basa en un saludo incompleto entre dos host.

El Cliente envía un paquete SYN pero no responde al paquete ACK ocasionando que la pila TCP/IP espere cierta cantidad de tiempo a que el host hostil responda antes de cerrar la conexión. Si se crean muchas peticiones incompletas de conexión (no se responde a ninguna), el Servidor estará inactivo durante mucho tiempo esperando respuesta. Esto ocasiona la lentitud en los demás servicios.

SYN Flood aprovecha la mala implementación del protocolo TCP, funcionando de la siguiente manera:

Se envía al destino una serie de paquetes TCP con el bit SYN activo, (petición de conexión) desde una dirección IP Spoofeada. Esta última debe ser inexistente para que el destino no pueda completar el saludo con el cliente.

Aquí radica el fallo de TCP: ICMP reporta que el cliente es existente, pero TCP ignora el mensaje y sigue intentando terminar el saludo con el cliente de forma continua.

Cuando se realiza un Ping a una máquina, esta tiene que procesarlo. Y aunque se trate de un proceso sencillo, (no es más que ver la dirección de origen y enviarle un Reply), siempre consume recursos del sistema. Si no es un Ping, sino varios a la vez, la máquina se vuelve más lenta… Si lo que se recibe son miles de solicitudes, puede que el equipo deje de responder (Flood).

Es obligatorio que la IP origen sea inexistente, ya que de no ser así el objetivo, logrará responderle al cliente con un SYN/ACK, y como esa IP no pidió ninguna conexión, le va a responder al objetivo con un RST, y el ataque no tendría efecto.

El problema es que muchos sistemas operativos tienen un límite muy bajo en el número de conexiones semiabiertas que pueden manejar en un momento determinado (5 a 30). Si se supera este límite, el servidor sencillamente dejará de responder a las nuevas peticiones de conexión que le vayan llegando. Las conexiones semiabiertas van caducando tras un tiempo, liberando “huecos” para nuevas conexiones, pero mientras el atacante mantenga el SYN Flood, la probabilidad de que una conexión recién liberada sea computada por un nuevo SYN malicioso es muy alta.

Tampeing o Data Diddling

Esta categoría se refiere a la modificación desautorizada de los datos de software instalado en el sistema de la víctima, incluyendo borrado de archivos. Son particularmente serios cuando el que lo realiza ha obtenido derechos de administrador o Supervisor, con la capacidad de disparar cualquier comando y por ende alterar o borrar cualquier información que puede incluso terminar en la baja total del sistema.

Aún así, si no hubo intenciones de bajar el sistema por parte del atacante, el administrador posiblemente necesite darlo de baja por horas o días hasta checar y tratar de recuperar aquella información que ha sido alterada o borrada.

Como siempre, esto puede ser realizado por Insiders u Outsiders, generalmente con propósito de fraude o dejar fuera de servicio a un competidor.

Son innumerables los casos de este tipo: empleados bancarios (o externos) que crean falsas cuentas para derivar fondos de otras cuentas, estudiantes que modifican calificaciones de exámenes, o contribuyentes que pagan para que se les anule una deuda impositiva.

Múltiples Web Sites han sido víctimas del cambio en sus páginas por imágenes (o manifiestos) terroristas o humorísticos, como el ataque de The Mentor, ya visto, a la NASA; o la modificación del Web Site del CERT en mayo 2001.

Otras veces se reemplazan versiones de software por otros con el mismo nombre pero que incorporan código malicioso (virus, troyanos, etc). La utilización de programas troyanos y difusión del virus esta dentro de esta categoría.

TCP Connect Scanning

Esta es la forma básica del scaneo de puertos TCP. Si el puerto esta escuchando, devolverá una respuesta de éxito; cualquier otro caso significará que el puerto no esta abierto o que no se puede establecer comunicación con él.

Las ventajas que caracterizan a esta técnica es que no se necesita de privilegios especiales y su gran velocidad.

Su principal desventaja es que este método es fácilmente detectable por el administrador del sistema. Se verá un gran número de conexiones y mensajes de error para los servicios en los que se han conseguido conectar la máquina, que lanza el scanner, y también se verá su inmediata desconexión.

TCP SYN Scanning

Cuando dos procesos establecen comunicación usan el modelo de Cliente/Servidor para establecerla. La aplicación del servidor escucha todo lo que le ingresa por los puertos.

La identificación del servidor se efectúa a través de la dirección IP del sistema en el que se ejecuta y del número de puerto del que depende para la conexión. El cliente

establece la conexión con el servidor a través del puerto disponible para luego intercambiar los datos.

La información de control se llama handshake (saludo) se intercambia entre el cliente y el servidor para establecer un diálogo antes de transmitir datos. Los paquetes o segmentos TCP tienen banderas que indican el estado del mismo.

El protocolo TCP de Internet, sobre el que se basa la mayoría de los servicios (incluye el correo electrónico, web y el IRC) implica esta conexión entre dos máquinas. El establecimiento de dicha conexión se realiza mediante lo que se llama Three-Way HandShake ("conexión en tres pasos") ya que intercambia tres segmentos.

1. El programa cliente (C) pide una conexión al servidor (S) enviándole un segmento SYN. Este segmento le dice a S que C desea establecer una conexión.
2. Si S esta abierto y escuchando al recibir este segmento SYN activa el indicador y envía una autentificación ACK de manera de acuse de recibo a C. Si S está cerrado envía un indicador RST.
3. C entonces ACKea (autentifica) a S. Ahora ya pueden tener lugar la transferencia de datos.

Cuando las aplicaciones conectadas terminar la transferencia, realizarán otra negociación a tres bandas con segmento FIN en ves de SYN.

La técnica TCP SYN Scanning implementa un scaneo de media apertura dado que nunca se abre una sesión TCP completa.

Se envía un paquete SYN como si fuera a usar una conexión real y espera por la respuesta. Al recibir un SYN/ACK se envía, inmediatamente un RST para determinar la conexión y se registra ese puerto como abierto.

La principal ventaja de esta técnica de scaneo es que pocos sitios están preparados para registrarlos. La desventaja es que algunos sistemas Unix, necesitan privilegios de administrador para construir paquetes SYN.

TCP FIN Scanning Stealth Port Scanning

Hay veces en que incluso el scaneo SYN no es lo suficientemente clandestino o limpio. Algunos sistemas firewalls y filtros de paquetes monitorizan la red en busca de paquetes SYN a puertos restringidos.

Para subsanar este inconveniente los paquetes FIN, en cambio, podrían ser capaces de pasar sin ser advertidos. Este tipo de scaneo esta basado en la idea de que los puertos cerrados tienden a responder a paquetes FIN con el RST correspondiente. Los puertos abiertos suelen ignorar el paquete en cuestión.

Este es un comportamiento correcto del protocolo TCP, aunque algunos sistemas, entre ellos los de Microsoft no cumplen con este requerimiento, enviando paquetes RST siempre, independientemente de si el puerto esta abierto o cerrado. Como resultado no son vulnerables a este tipo de scaneo. Sin embargo es posible realizarlo en otros sistemas Unix.

Muchos de los problemas globales de vulnerabilidades son inherentes al diseño original de algunos protocolos.

Teardrop I y II-Newtear-Bonk-Boink

Al igual que Supernuke, los atacantes Teardrop I y Teardrop II afectan fragmentos de paquetes. Algunas implementaciones de colas IP no vuelven a armar correctamente los fragmentos que se superponen, haciendo que el sistema se cuelgue. Windows NT 4.0 de Microsoft es especialmente vulnerable a este ataque. Aunque existen parches que pueden aplicarse para solucionar este problema, muchas organizaciones no lo hacen, y las consecuencias pueden ser devastadoras.

Trashing (Cartoneo)

Generalmente el usuario anota su login y su password en un papel y luego, cuando lo recuerda, lo arroja a la basura. Este procedimiento por más inocente que parezca es el que puede aprovecharse el atacante para hacerse de una llave para entrar al sistema.

El trashing puede ser físico (como en el caso descrito) o lógico, como analizar buffer de impresora y memoria, bloques de discos, etc.

El trashing físico puede ser común en organizaciones que no disponen de alta confidencialidad, como colegios y universidades.

Verificación de voz

La dicción de una o más frases es grabada y en el acceso se compara la voz (entonación, diptongos, agudeza,etc).

Este sistema es muy sensible a factores externos como el ruido, el estado de ánimo, enfermedades de la persona, envejecimiento, etc.

Verificación de patrones oculares

Estos modelos pueden estar basados en los patrones del iris o de la retina y hasta el momento son los considerados los más efectivos (en 200 millones de personas la probabilidad de coincidencia es casi 0).

Virus en el sector de arranque

En los primeros 512 bytes de un disquete formateado se encuentra las rutinas necesarias para la carga y reconocimiento de dicho disquete. Entre ellas se encuentra la función invocada si no se encuentra el Sistema Operativo. Es decir estos 512 bytes se ejecutan cada vez que se intenta arrancar (bootear) desde un disquete (o si se dejó olvidado uno en la unidad y el orden de booteo de la PC es A: y luego C:). Esta área es el objetivo de un virus de booteo.

Se guarda en la zona de booteo original en otro sector del disco (generalmente uno muy cercano a los más altos). Luego el virus carga la antigua zona de booteo. Al arrancar el disquete se ejecuta el virus (que obligatoriamente debe tener 512 bytes o menos) quedando residente en memoria; luego ejecuta la zona de booteo original, salvada anteriormente. Una vez más el usuario no se percata de lo sucedido ya que la zona de booteo se ejecuta iniciando el sistema operativo (si existiera) o informando la falta del mismo.

BIBLIOGRAFÍA

1. Emilio del Peso Navarro Miguel A. Ramos, “LA INFORMACIÓN COMO ACTIVO ESTRATÉGICO: SEGURIDAD Y PROTECCIÓN”, Universidad Carlos III de Madrid

2. Emilio del Peso y Miguel A. Ramos, ”La Auditoría Informática: un enfoque práctico", Editorial RA-MA (Madrid).

3. Óscar López, Haver Amaya, Ricardo León, Beatriz Acosta “INFORMÁTICA FORENSE : GENERALIDADES, ASPECTOS TÉCNICOS Y HERRAMIENTAS”, Universidad de Los Andes Bogotá, Colombia

4. Francisco Jesús Monserrat Coll, “Incidentes de seguridad en equipos Linux”, Centro de Comunicaciones del CSIC RedIRIS

5. Brian Coswell, “Snort 2.0 Intrusion Detection”, SNORT.org Edit. Technical Editor

6. jay Beale, James C. Foster, Jeffrey Posluns, “Technical Advisor”, Edit. Syngress, 523 pp

7. Robert S. Sielken, “Application Intrusion Detection” , University of Virginia.

8. W. Lee, P. Chan, E. Eskin y M. Millar, “Real Time Data Mining-based Intrusion Detection” Columbia University.

9. J. Sundar, J. Omar, D. Isacoff, E. Spafford y D. Zamboni, “An Architecture for Intrusion Detection using Autonomous Agents”, urdue University.

10. S. Northcutt, D. McLachlan y J. Novak, “Network Intrusion Detection: An Analyst.s handbook”, New Riders Publishing, septiembre 2000.

11. Antonio Rincón, “Diccionario Conceptual de Informática y Comunicaciones”, España, Edit Paraninfo, 1998.

12. Gil Julio Pizarro, “Diccionario General de Informática”, Edit. ABETO, España, 1999.

13.O’Reilly, “Seguridad Práctica en Unix e Internet”, Edit. Mc Graw Hill, 1999

14. Deborah G. Jhonson y John M Mulvey, “Accountability and Computer Decision System”. Communications of the ACM, Vol38.

16. Karanjit Siyan and Chris Hare, “Internet y Seguridad en Redes”, Edit. Prentice Hall.

17. Línea de Especialización Análisis Forense e Implicaciones Legales, DGSCA-UNAM, 2003.

18. Línea de Especialización de Detección de Intrusos y Tecnologías HoneyPots. DGSCA-UNAM,2003.

19. Margarita Carrera Fournier, Roberto Carlos Zúñiga Ramírez, Yesenia Carrera Fournier, “Tesis: ESTRATEGIAS, PROCEDIMIENTOS Y POLÍTICAS PARA IMPLEMENTAR LA SEGUIRDAD INFORMÁTICA EN ORGANIZACIONES CON SISTEMAS LINUX RED HAT, CASO: UNIDAD DE SERVICIOS DE CÓMPUTO ACADÉMICO DE LA FACULTAD DE INGENIERÍA”(Licenciatura en Ingeniería en Computación), México, D.F. Universidad Nacional Autónoma de México, Facultad de Ingeniería.

20 Cinthia Reyes Quezada y Sergio Rodríguez Gutiérrez, “Tesis: FUNDAMENTOS DE SEGURIDAD DE LA INFORMACIÓN”(Licenciatura en Ingeniería en Computación). México, D.F., Universidad Nacional Autónoma de México, Facultad de Ingeniería, 2001.

21. Cristian F. Borhelio, Tesis “Seguridad Informática: Sus Implicaciones y Implementación”, 2001.

REFERENCIAS

1. http://www.nicatech.com.ni/r.htm

2. http://www.tuxteno.com/contents.php?cid=489

3. http://www.tripwire.org.

4. http://www.beeeeee.net/nautopia/linux/rootkit_deteccion.htm

5. http://www.rediris.es/cert/ped/reto/04/inf0_reto311203.txt

6. http://www.virusprot.com/Vi000092.html

7. http://es.tldp.org/Manuales-LuCAS/doc-unixsec/unixsec-html/node141.html

8. *www.first.org*

9. *www.gosci.gov*

10. http://www.enterate.unam.mx/Articulos/dos/noviembre/unamcert.htm

11. ftp://cert.org

12. http://www.super.unam.mx/seguridad/

13. http://escert.upc.es/castella/cert_faq.html

14. http://info.isoc.org:80/adopsec/index.html

15. http://www.first.org/team-info/

16. http://www.first.org/about/first-description.html

17. http://grupo.lugunar.com/crimendigital.php

18. http://grupo.lugunar.com/crimendigital.php

19. http://project.honeynet.org

20. http://www.hispasec.com/directorio/servicios/formacion/IDSanalisisforense

21. http://www.lafacu.com/apuntes/informatica/segu_prote/default.htm

22. http://www.portaley.com/pruebas-periciales/

23. http://www.bs.com.ar/bsweb/RevistaBSknow/Revistajunio03PDF/Principal.pdf.

24. http://personales.ciudad.com.ar/roble/seguridadinformatica.htm

25. http://www.gratisweb.com/nmrg291/ronaldmitre.htm

26. http://www.lasalle.edu.co/csi_cursos/informatica/termino/seguridadinformatica.htm

27. http://linux.ubiobio.cl/pasados/primero/documentacion/hvb/seguridad/x46.html

28. http://w4ww.cert.org/stats/cert_stats.html#incidents

29. www.aslan.es/boletin4/entrevista.shtm

30. http://loquefaltaba.com/documentacion/forense/index.html

31. http://www.robota.net/article?id=838&PHPSESSID=7431745d8e6333d4 5d49fa7a21eacc94

32. http://www.chkrootkit.org

33.http://www.rl.ac.uk/sysman/april2003/talks/SecurityIncidentReport.ppt+initsk12& hl=es

34.http://umeet.uninet.edu/umeet2003/spanish/talks/20031219.3.es.html+initsk12& hl=es&lr=lang_es

35. http://www.softwarelibre.cl/print.php?sid=222

36. http://mx.geocities.com/lemt78

37. http://www.linux.cu/manual/basico-html/node101.html

38. http://www.lcu.com.ar/faq.php