Esquema de integridad de bitácoras

Roberto Gómez Cárdenas, Ricardo C. Lira Plaza, Adolfo Grego
ITESM-CEM, Departamento de Ciencias Computacionales
Apdo. Postal 50, Módulo Servicio Postal, Atizapán Zaragoza, 52926, México
{rogomez,rlira,agrego}@campus.cem.itesm.mx

ABSTRACT

Computer security is defined as the mechanisms and policies set that assures confidentiality, integrity and availability of systems resources. In a forensics analysis is necessary that data logs remain unchanged. Our work proposes a schema, based in secret sharing theory that assures the integrity and confidentiality of log information. Furthermore our proposition allows fault tolerance, so the availability feature of the data log is also considered.

Keywords: crytology, computer security, networks, distributed systems

RESUMEN

La seguridad computacional se define como el conjunto de políticas y mecanismos que nos permiten garantizar la confidencialidad, la integridad y la disponibilidad de los recursos de un sistema. En un análisis forense es necesario que los datos contenidos en las bitácoras sean íntegros. Nuestro trabajo propone un esquema que involucra la teoría de secretos compartidos en un ambiente distribuido que permite salvaguardar la integridad así como la confidencialidad de la información para su futuro análisis. Además nuestra propuesta incorpora tolerancia a fallas por lo que el aspecto de la disponibilidad también es considerado.

Palabras clave: criptología, seguridad computacional, redes, sistemas distribuidos

1 INTRODUCCIÓN

Nuestro esquema se basa en aceptar la posibilidad de que un sistema puede ser penetrado, ya que nadie puede garantizar una seguridad perfecta. Con nuestra propuesta lo anterior no implica que la información sensitiva quede en manos del intruso. Lo que proponemos es un esquema en el que dicha información no esté físicamente en el sistema, sino distribuida en N equipos usando el concepto de secretos compartidos.

El problema principal en un análisis forense es el saber si el intruso ha modificado, borrado o alterado las bitácoras del sistema. Es necesario que la información que se va a analizar esté libre de cualquier distorsión o modificación ya que sólo así se podrán reconstruir los hechos, obtener evidencias reales, concluir de manera acertada y ejercer acción sobre los presuntos responsables. Es por esto que es necesario garantizar que el intruso no tiene acceso a dicha información. El esquema que nosotros proponemos permite salvaguardar la integridad y la confidencialidad de la información sensitiva dentro del sistema como son las bitácoras, las llaves privadas, respaldo del contenido público de un servidor de web, archivo de contraseñas, documentos confidenciales, bases de datos, etc. En este artículo nos basaremos en la integridad de las bitácoras.

La importancia de salvaguardar la integridad de las bitácoras parte del hecho de que un intruso después de modificar o robar la información del sistema, intentará borrar cualquier rastro de su presencia o huella que permita su futura identificación. Son precisamente las bitácoras en donde quedan plasmadas sus huellas y todo lo que hizo dentro del sistema.

La base de nuestro esquema es la teoría de Secretos Compartidos desarrollado por Adi Shamir [1]. Este esquema cuenta con características que lo hace muy adecuado para nuestra aplicación, como el no depender de una contraseña para brindar confidencialidad a la información así como tolerancia a fallas (dividiendo la información en N partes y poder reconstruirla con M partes donde M<N).

Nuestro trabajo propone crear un ambiente distribuido a través de la técnica de Secretos Compartidos para salvaguardar la integridad y brindar confidencialidad a las bitácoras. Lo anterior permitirá un acertado análisis de forensia de datos.

El trabajo se encuentra dividido de la siguiente forma: en la siguiente sección se presenta un panorama general de las bitácoras. La sección tres explica los conceptos fundamentales de los secretos compartidos. La sección cuatro da a conocer nuestra propuesta, mientras que la sección cinco la implementación de ésta. Por último se presentan las conclusiones y el trabajo a futuro.

2 LAS BITACORAS

El proceso de monitorear el comportamiento de un sistema es conocido como auditoría, dicho proceso sólo es posible si se cuenta con la información adecuada dentro de las bitácoras. Una vez que se han establecido los mecanismos de protección del sistema, el siguiente paso es monitorear dicho sistema, solo así podremos identificar comportamientos no deseados.

En realidad las bitácoras son las que nos muestran quien, como y cuando está usando el sistema, así como lo que se está haciendo durante una sesión. Es así como es posible identificar problemas y/o ataques, y los alcances de estos. En muchas ocasiones son las bitácoras los mecanismos que ayudan a reconstruir un sistema después de que éste ha sido atacado. Además son la única fuente de información que permite rastrear al intruso para ejercer acciones futuras.

Además de servir para la auditoría del sistema, las bitácoras también pueden mostrar actividades poco habituales como el intentar acceder a la cuenta del super usuario así como el intento de violar mecanismos de seguridad, el indagar por lugares del sistema que no son propios del usuario. Son una herramienta indispensable para el monitoreo de las actividades de nuestros usuarios y de intentos por acceder a él.

Una vez mencionada la importancia de las bitácoras debemos establecer ahora las consecuencias que puede representar el que ellas sean alteradas. Como dijimos con anterioridad, son la única fuente que nos pueden

llevar a rastrear a un intruso para ejercer alguna acción sobre él. Pero, ¿qué sucede si dicha información ha sido alterada? ¿qué sucede si no podemos garantizar su integridad?. La respuesta es sencilla: es información que no nos sirve para nada. Únicamente garantizando la integridad de la información contenida dentro de las bitácoras podremos realizar un análisis de detección de intrusos o de forensia de datos.

Una opción para salvaguardar la integridad de la bitácoras es enviarlas a otro sistema. Esta es una buena solución, sin embargo, se sigue corriendo el riesgo de que si un intruso logra penetrar el sistema seguramente podrá penetrar otro segundo sistema en el que se encuentra las bitácoras. En efecto, el grado de dificultad se aumentó, pero el riesgo de exponer información al intruso continúa latente. Es por esto que nuestra propuesta es distribuir las bitácoras a través de la técnica de secretos compartidos hacia N sistemas, por lo que ahora el intruso tendrá que penetrar M de N sistemas (donde M<N y M>1).

3 SECRETOS COMPARTIDOS

Adi Shamir en [1] propone una técnica para dividir un secreto en N partes, de tal manera que el tener una de estas partes no da ninguna información. Son necesarias M de N partes para poder recuperar el secreto. Se establece que esta técnica puede ser utilizada para brindar seguridad a las llaves, nosotros pensamos que tiene mayores alcances ya que puede servir además para darle seguridad, bajo ciertos parámetros, a cualquier tipo de información. Las mejores características de dicha teoría son que no depende de una contraseña y que necesita M de N partes para recuperar la información, donde M<N (figura 1).

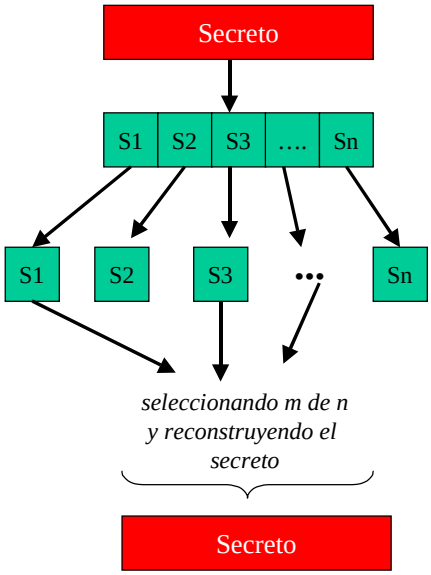

Figura 1. Esquema general de secretos compartidos

El almacenar la información en un solo lugar crea el problema del llamado punto único de falla, además si el sistema es penetrado se tiene acceso a toda la información. El posible mal funcionamiento del sistema compromete la integridad de la información. Es por esto que proponemos la técnica de secretos compartidos para dividir la información sensitiva, en nuestro caso las bitácoras, en N partes. Cada una de estas N partes será enviada a N sistemas (N1, N2, N3, …, Nn) dentro de una red privada. Al implementar esta técnica podemos garantizar la integridad de la información ya que no estará físicamente en el sistema original (expuesto a una red pública) sino distribuida en N sistemas dentro de una red privada. También aumentamos su disponibilidad ya que pueden corromperse hasta N-M partes y todavía puede recuperarse la información y finalmente obtenemos bajo un ambiente distribuido confidencialidad de la información, ya que el tener M-1 partes no me permite tener acceso a la información.

La teoría de los secretos compartidos cuenta con las siguientes propiedades:

- Para recuperar la información original es necesario tener un número significativo de las partes.
- La información original debe poder recuperarse aún cuando una porción significativa de las partes haya sido comprometida.
- El que recibe una de las partes debe poder verificar que la división de la información se ha realizado de manera correcta.

El esquema original propuesto por Adi Shamir sólo satisface las dos primeras propiedades, razón por la cual se introdujo un nuevo concepto basado en la misma teoría, llamado Secretos Compartidos Verificables. Con ésta se puede satisfacer la tercera propiedad. Cada una de las partes puede verificar que tiene una parte de la información que permitirá el reconstruir la información original, aunque la información que tiene no le permite conocer o intuir la información original. Esta teoría ha sido ampliamente estudiada por Rosario Gennaro en [2].

Uno de los problemas de la teoría de los secretos compartidos es que la información original debe ser recuperada en un cierto lugar [3], si dicho lugar es el sistema que está expuesto a la intrusión obviamente la integridad de la información se vería comprometida. Sin embargo, para nuestra aplicación no es necesario que la información original se recupere en dicho sistema, de hecho puede ser recuperada en cualquiera de los sistemas que cuenta con uno de los secretos compartidos (N1, N2, N3, …, Nn). Aún más, podemos respaldar cada uno de los secretos compartidos y procesarlos en un sistema físicamente desconectado de todo nuestro esquema.

4 NUESTRA PROPUESTA

Nuestra propuesta consiste en utilizar la teoría de secretos compartidos para distribuir las bitácoras entre N sistemas que residen en una red que consideramos privada (no está conectada físicamente al ambiente hostil). Mediante esta propuesta podemos garantizar la integridad de la bitácoras ya que el intruso tendría que penetrar M sistemas para poder tener acceso a dicha información. En un inicio proponemos el esquema (M,N) donde M igual con 2 y N igual con 3, es decir, las bitácoras se dividen en 3 partes y se requerien 2 partes para reconstruirlas. Desde otro punto de vista, el intruso requiere penetrar primero un sistema, conectado al ambiente hostil, y para tener acceso a las bitácoras de dicho sistema requiere penetrar otros dos dentro del ambiente privado. Es importante recordar que una característica esencial de los secretos compartidos es que una sola parte no proporciona información sobre la original, se requieren por lo menos M partes. Y, desde otro punto de vista, puede fallar uno de los tres sistemas y aún así podemos recuperar de manera íntegra las bitácoras, con esta propiedad aumentamos la disponibilidad de nuestra información.

Las principales ventajas de nuestra esquema son que se puede garantizar la integridad de la bitácoras ya que éstas ya no están dentro del sistema sino en N sistemas dentro de una red privada por lo que el intruso no podrá borrarlas, ni modificarlas, ni siquiera observarlas. El hecho de no estar físicamente presentes en el sistema conectado al ambiente hostil les proporciona confidencialidad. Por último aumentamos la disponibilidad de las bitácoras ya que puede fallar un sistema y aún así se puede acceder a la información original.

Dicho esquema es muy flexible ya que podemos aumentar tanto M como N de acuerdo a nuestros requerimientos. En un ambiente muy hostil podemos proponer M igual con 3 y N igual con 5, con lo cual necesitamos 3 partes de 5 para recuperar nuestras bitácoras. Este esquema también se puede implementar en un ambiente que requiera alta disponibilidad, para lo cual se requiere que M sea relativamente pequeño y N relativamente grande. En [11] se encuentra un estudio sobre el tamaño de M y N.

También proponemos el uso de controles de acceso entre el sistema A y la red segura. Dichos controles comprenden el filtrado de tráfico de A solo hacia ciertos sistemas y el tráfico de los sistemas de la red segura hacia A. Para complicar el esquema el puerto por el que se comunica A con los sistemas dentro de la red segura cambian con el tiempo. Es importante señalar que consideramos a la red interna (R1,R2,...,Rn) ya que es un segmento de red dedicado exclusivamente para nuestro esquema. Se tiene un riguroso control de acceso físico a dicho segmento y los sistemas solo pueden ser administrados por consola y nunca de manera remota. Todos los servicios de red están deshabilitados y solo cuentan con los programas “recibe y almacena”.

En primera instancia hemos elegido las bitácoras como elemento a proteger, sin embargo, este esquema también puede servir para salvaguardar la integridad y la confidencialidad de la información sensitiva dentro del sistema, por ejemplo, el archivo de contraseñas, el respaldo del contenido público de nuestro servidor de web (utilizado en esquemas de restablecimiento automático de información en caso de ser modificada, ej. Tripwire), información de los usuarios del sistema, huellas de los distintos programas utilizados (con lo cual podemos evitar caballos de troya y códigos maliciosos), bases de datos, llaves privadas y secretas, así como documentos con información clasificada.

5 IMPLEMENTACIÓN

Lo primero a definir es la información a almacenar dentro de las bitácoras generadas en el sistema A. En [4] se establecen los principales puntos que deben auditarse en un sistema basado en un extensa investigación para obtener información suficiente que nos lleve a conclusiones adecuadas sobre la intrusión a un sistema.

Entre las principales recomendaciones se establece que las acciones deben ser auditadas al principio de su ejecución y no al final como se acostumbra hacerse. Así mismo es necesario establecer los parámetros introducidos en ciertos programas para conocer las intenciones reales de un intruso, por ejemplo, el programa de búsqueda no manda a la bitácora lo que se está introduce en el campo de la búsqueda. Los mecanismos que auditan el uso y ejecución del comando *"su"* no están bien establecidos ya que pueden llegar a enmascararse con lo que un intruso podría no llegar a detectarse.

Además de estas consideraciones, se recomienda habilitar las bitácoras necesarias para que un sistema sea considerado con nivel de seguridad C2, avalado por el libro naranja (TCSEC, Trusted Computer System Evaluation Criteria). En éste se establece la necesidad de auditar: logins, logouts, accesos remotos al sistema, apertura, cerrado, renombrado y borrado de archivos, cambios en los privilegios y atributos de seguridad de los archivos y programas. Además se requiere capturar para cada uno de los eventos anteriores, la fecha y hora del evento, la identificación única del usuario que inició el evento, el tipo de evento, éxito o fallo del evento, el origen de la petición, el nombre del objeto involucrado, descripción de las modificaciones a las base de datos de seguridad.

Nuestro esquema requiere el establecer dos tipos de redes, la primera una red hostil o pública y la segunda una red segura o privada a la que se le pueden aplicar controles de acceso y un monitoreo muy cercano. El sistema A, conectado al ambiente hostil, fue implementado en una máquina que consta de dos tarjetas de red, una conectada al ambiente hostil y otra conectada a la red segura. Los sistemas de la red segura R1, R2, ..., Rn (que contienen los secretos compartidos) se implementaron en tres máquinas de la misma red. Las cuatro máquinas forman parte de una red local, formada por varias computadoras IBM Intel stations, pentium III a 500 mhz y con 128M en RAM. Cada computadora cuenta con el sistema operativo *"Security-Enhanced Linux*" desarrollado por la Agencia de Seguridad Nacional de los Estados Unidos. Si bien es cierto que las computadoras cuentan con una buena capacidad de computo, los sistemas R1, R2, ..., Rn no requieren gran poder de procesamiento ya que solo escuchan por un puerto para recibir la información enviada por A. Sin embargo, es aconsejable que tengan un espacio de almacenamiento adecuado al nivel de las bitácoras que se están procesando. Lo anterior se encuentra ilustrado en la figura 2.

El sistema A contiene el programa que toma cada una de las entradas a las bitácoras, les aplica el algoritmo de secretos compartidos y las N partes las envía a través de sockets por un puerto que cambia conforme al tiempo a R1, R2, ... Rn. Estos últimos cuentan con un programa que "recibe y almacena" la información enviada por A. La información de las bitácoras es recuperada, para su análisis ya sea forense o de detección de intrusos, en un equipo ajeno a todo el sistema. Una opción alterna es que la información se almacene en uno de los R1, R2, ..., Rn sistemas. No se recomienda hacer dicha operación en el sistema A, puede haber un intruso en el sistema, que podría comprometer el esquema.

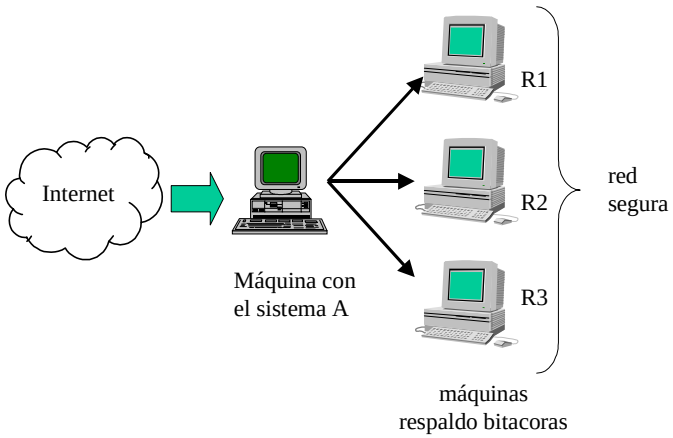

Figura 2. Esquema general del sistema.

A todos los sistemas (A, R1, R2, …, Rn) se les realizó una auditoría inicial de todos sus archivos y programas y se obtuvo una huella digital (MD5) de los mismos (a través del software Tripwire). Dichas huellas se guardaron en un medio de solo lectura para garantizar su integridad y poder cotejar en caso de un incidente. Además se realizaron monitoreos con herramientas como Nessus, Cops y Saint para detectar y corregir las vulnerabilidades más comunes. También se instalo en A un detector de intrusos, Snort, cuyas bitácoras se enviaron respetando el esquema de secretos compartidos a R1, R2 y R3. Se ha instaló Tcp Wrappers en los sistemas R1, R2 y R3 para que solo puedan ser accedidos bajo ciertas direcciones, protocolos y puertos. Se instaló Xinetd el cual es una extensión del demonio de internet presente en la mayoría de los sistemas Linux con la ventaja de que éste genera bitácoras.

El desempeño de las máquinas no se vio afectado por el sistema y la reconstrucción de las bitácoras no tomó mucho tiempo (unos 5 segundos). A través de las opciones de calendarización de linux (crontab) el sistema se activa en periodos determinados para aplicar el algoritmo de secretos compartidos y enviar la información de las bitácoras.

6 Conclusiones

Se presentó un esquema que permite garantizar la integridad de las bitácoras generadas en un sistema. El esquema se basa en la teoría de secretos compartidos propuesta por Adi Shamir y fue implementado en un ambiente distribuido.

Nuestro esquema permite una gran flexibilidad que permite adaptarse a distintos escenarios dependiendo el grado de hostilidad al cual nos estemos enfrentando, así como al grado de disponibilidad que requerimos. Este esquema no represento ninguna carga considerable al sistema.

Aun restan varias pruebas por hacer, tan sólo se hicieron pruebas a nivel reconstrucción con dos de tres máquinas y con información “ligera”. Lo próximo a realizar es probar el sistema con un conjunto más grande de computadoras, así como en un equipo de producción. Este tipo de pruebas nos llevará a definir parámetros que permitan establecer el valor correcto de los parámetros M y N de la teoría de secretos compartidos. También se esta explorando el utilizar el esquema de secretos compartidos con otro tipo de información crítica del sistema.

7 Referencias

[1] A. Shamir, “How to share a secret”, Comm. of the ACM, Vol. 22, 1979, pp 612-613.

[2] R. Gennaro, “Theory and Practice of Verifiable Secret Sharing”, Ph. D. thesis, Massachusetts Institute of Technology, 1996.
[3] T. Wu, M. Malkin, D. Boneh, “Building Intrusion Tolerant Applications”, In Proceedings of USENIX Security Simposium, August 1999.
[4] S. Axelsson, “An approach to Unix Security Logging”, In Proceedings of the 21st National Information Systems Security Conference, 62-75. EUA, octubre 5-8, 1998.
[5] B. Schneier, “Cryptographic Support for Secure Logs on Untrusted Machines”, The Seventh USENIX Security Simposium Proceedings, USENIX Press, Jan 1998, pp. 53-62.
[6] B. Schneier, J. Kelsey, “Minimizing Bandwith for Remote Access to Cryptographically Protected Audit Logs”,
[7] B. Schneier. “Applied Cryptography” . John Wiley and Son, Inc. 1996, ISBN 0-471-12845-7
[8] S. Garfinkel, G. Spafford. “Practical Unix and Internet Security”. O’Reilly & Associates Inc. 1996, ISBN 1-56592-148-8
[9] W. Stallings. “Cryptography and Network Security”. Prentice Hall Inc. ISBN 0-13-869017-0
[10] H. Tipton, M. Kruse. “Information Security Management Handbook”. Auerbach Publications. ISBN 0-8493-9829-0
[11]A. de Santis, L. Gargano, U. Vaccaro. “On the size of shares for Secret Sharing Schemes”, J. Cryptology 6 (1993), 157-167. [Preliminary version appeared in "Advances in Cryptology --CRYPTO '91", J. Feigenbaum, ed., Lecture Notes in Computer Science 576 (1992), 101-113.]