SISTEMA AUTÓNOMO PARA LA DETECCIÓN DE INTRUSIÓN EN UN ENTORNO DISTRIBUIDO.

Fernando Miguélez Palomo, Armando Ferro Vázquez

Dpto. de Electrónica y Telecomunicaciones. Grupo de Ingeniería Telemática (G.I.T.)

E.T.S.I.I. y de I.T. de Bilbao – Alda. Urquijo s/n 48013-Bilbao
jtbmipaf@bipt106.bi.ehu.es, jtpfevaa@bi.ehu.es

ABSTRACT

Most of intrusion detection systems rely on a single entity, usually a process inside a monitored system, which lack flexibility to adapt themselves to other configurations and systems. This paper introduces an autonomous agent that not only can be easily ported to different platforms but also may work in a distributed environment with other agents under the co-ordination of a central server station, rather than just in a monolithic way.

1. INTRODUCCIÓN

Existen varias definiciones sobre lo que es la detección de intrusión. Una de ellas es la propuesta por Edward Amoroso [1]:

"La detección de intrusión es el proceso de identificar y responder a una actividad maliciosa dirigida a recursos de computación y de red".

Aunque existen muchas clasificaciones, y algunas no muy claras, habitualmente los sistemas de detección de intrusión (SDI) se suelen clasificar en dos grandes categorías:

- **Basados en red**. Funcionan capturando todos los paquetes que viajan por la red a la que están conectados y buscar en ellos ciertas palabras sospechosas o firmas de comportamiento anormal (*signatures of abnormal behaviour)*. Este tipo de sistemas también se puede conocer como SDIs *on-the-fly* [1], porque suelen realizar el procesado según van recibiendo la información.
- **Basados en host**. Son sistemas que basan su funcionamiento, normalmente, en la información de auditoria que proporciona el sistema operativo de la máquina en que residen, y cuyo acceso pretenden vigilar.

Este artículo presenta la arquitectura de un agente autónomo modular, orientado a la detección de intrusión basada en la utilización de firmas de comportamiento anormal y procesado on-the-fly.

El agente autónomo propuesto está basado en un diseño modular flexible que permite su portabilidad a distintos tipos de plataformas. Se pueden establecer configuraciones diversas para la detección de intrusión. El prototipo desarrollado pretende ayudar a analizar cual es la configuración más adecuada para obtener el mejor rendimiento en el análisis de tráfico intruso.

2. ENTORNO DE OPERACIÓN

La flexibilidad es un elemento clave en el agente autónomo propuesto, la cuál le permite adaptarlo de una forma relativamente sencilla a una serie de entornos de operación diferentes. Estos entornos pueden variar desde una configuración monolítica en forma de uno o varios procesos ejecutándose dentro de un ordenador personal, hasta una configuración distribuida, consistente en una serie de sistemas autónomos conectados y armonizados a través de un servidor central. Dentro de la opción distribuida las configuraciones también pueden ser múltiples, siendo una opción interesante aquella en la que los agentes se ejecutan en dispositivos hardware dedicados y utilizan kernels adaptables a las necesidades del agente [2].

3. ARQUITECTURA DEL SISTEMA

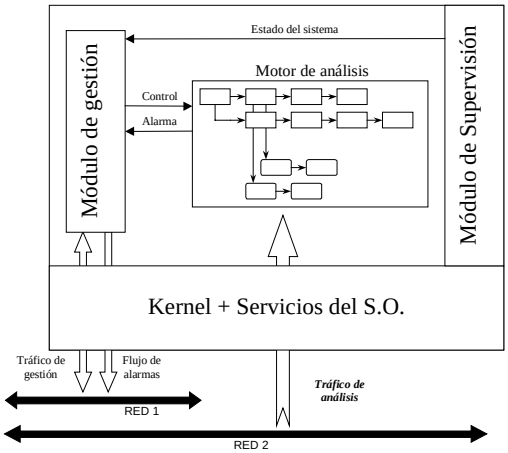

Fig. 1. *Arquitectura modular del sistema autónomo.*

3.1. Kernel y servicios del SO

Este módulo representa la plataforma software sobre la que se está ejecutando el agente. Puede tratarse de cualquier variante del S.O. UNIX. Para la construcción y validación del prototipo se ha elegido Linux. Una alternativa interesante al Linux convencional es RT-Linux que permite ejecutar procesos críticos en tiempo real conservando toda la funcionalidad del Linux

tradicional gracias a la utilización de un microkernel por debajo del kernel de éste [3]. Otra opción sería utilizar un kernel específico adaptado a las necesidades del agente [4]. En el prototipo de validación se ha desarrollado un kernel específico basado en el paquete OSKIT.

Independientemente de la implementación, este módulo debe proporcionar una serie de servicios comunes, como la captura de paquetes de red para su análisis, capacidad multitarea y mecanismo de registro de eventos (syslog), entre otros.

3.2. Motor de análisis

Es el corazón del sistema. Se basa en un sistema dinámico de listas cuyos nodos se asocian a funciones de análisis, las cuáles mediante una serie de algoritmos aplican una búsqueda de firmas de ataque descritas en un fichero de configuración. Este fichero puede residir localmente u obtenerse remotamente en el caso distribuido.

Básicamente el agente carga información sobre cómo debe analizar tráfico intruso en un sistema de listas dinámico que reflejarán la lógica de detección descrita en un conjunto de reglas proporcionadas. Este sistema de reglas es estándar de otros SDIs pero en nuestro caso se propone una lógica de seguimiento de los incidentes más precisa y con mejores rendimientos utilizando para ello una arquitectura adaptable. El sistema de listas permitirá adaptarse dinámicamente a las necesidades de detección necesarias en cada caso y generalizando estas listas para poder trabajar en memoria compartida se pueden implementar arquitecturas multiprocesador.

Es interés del trabajo de investigación el análisis de diferentes configuraciones para la arquitectura de detección de intrusión. Para ello se utilizará el prototipo desarrollado con configuraciones diversas. Por ejemplo se pretende analizar el impacto de arquitecturas multiprocesador en los rendimientos del sistema de detección de intrusión. Las configuraciones básicas a estudiar son:

- SDI basado en una arquitectura de listas estática.
- SDI de listas dinámicas compartidas.
- SDI de listas dinámicas con varios procesadores.
- SDI basado en servicios de kernel a medida.
- SDI basado en kernel a medida en un entorno multiprocesador.

De este estudio se espera poder obtener conclusiones respecto a las opciones de diseño más adecuadas para conseguir los mejores rendimientos en el análisis de tráfico intruso.

3.3. Módulo de supervisión

Se encarga de monitorizar el correcto funcionamiento del sistema. Verifica en todo momento que el motor de análisis puede hacerse cargo de todo el tráfico que se le suministra. Esto es de vital importancia en un análisis on-the-fly. También se encarga de comprobar que no se producen errores en el sistema. En caso de producirse una situación anómala (saturación del sistema o fallo) lo comunica al módulo de gestión.

3.4. Módulo de gestión.

Configura el resto de módulos del agente y controla su operación a partir de las indicaciones de estado procedentes del módulo de supervisión. En el caso distribuido hace de interfaz entre el agente y el servidor, ejecutando las tareas que éste le encarga, comunicándole las alarmas producidas como fruto del análisis del tráfico e informando sobre el estado del sistema autónomo.

Para la transmisión de las alarmas se utiliza el mecanismo syslog debido a que su utilización es relativamente sencilla y el procedimiento es el mismo para registrar eventos de forma local como de forma remota, lo que unifica el procedimiento de generación de alarmas para los casos distribuido y monolítico. Para la transmisión del resto de información de gestión, en el caso distribuido, se puede utilizar un protocolo estándar de Internet como el SNMP. Este protocolo implica que el agente tenga una MIB (Management Information Base) con todos los parámetros que son visualizables y/o configurables en la gestión del agente.

La información de gestión viaja por una red diferente (ver figura) a la información a analizar por motivos de seguridad.

4. CONCLUSIÓN

La división modular del agente es lo que permite una mayor flexibilidad y portabilidad del mismo a diversas configuraciones y plataformas. Dicha división se da no sólo en el código fuente sino también en la ejecución en varios procesos comunicados eficientemente entre sí por los mecanismos de comunicación interproceso que ofrece UNIX. Esta elección proporciona una mayor independencia y separabilidad entre los módulos.

De la arquitectura expuesta se puede deducir que para permitir un funcionamiento monolítico o distribuido el módulo que más modificaciones debe sufrir es el de gestión, permaneciendo el resto prácticamente inalterable.

5. AGRADECIMIENTOS

El trabajo aquí presentado está basado principalmente en los resultados obtenidos en el proyecto de investigación “Modelo distribuido para la detección de intrusión en grandes redes.” INTEK 2000 UNOD24 financiado por el Gobierno Vasco y la empresa I+D TOKI.

6. REFERENCIAS

[1] Edward Amoroso. “Intrusion Detection”. Intrusion.net books. 1999.
[2] Eduardo San Félix, Armando Ferro, Cristina Perfecto. “Arquitectura distribuida de agentes autónomos para detección de intrusión” *URSI 2000*, 2000.
[3] Barabanov M., Yodaiken V. “Real Time Linux”, *Real Time Linux Documentation Project*, Mar 1996.
[4] Armando Ferro et al. “Diseño de un sistema distribuido basado en un kernel adaptable para la implementación de servicios de comunicaciones”, JITEL 2001.