Framework para Detección de Intrusos usando DeLP

Luciano M. Guasco Javier Echaiz Jorge R. Ardenghi

Laboratorio de Investigación de Sistemas Distribuidos (LISiDi)
Departamento de Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur, Bahía Blanca (8000), Argentina
{lmg,je,jra}@cs.uns.edu.ar

Resumen

Uno de los principales problemas que se presentan en las redes de mediana y gran escala es la dificultad para monitorear y detectar vulnerabilidades que puedan ser explotadas. Tanto los recursos que se comparten en una red, como los principales hosts que intervienen en la misma quedan expuestos a posibles ataques que pueden causar pérdidas considerables en los datos.

En esta línea de investigación se pretende desarrollar un framework de detección de intrusos, donde todos los hosts que participan de una red puedan argumentar si una vulnerabilidad en algún host puede ser explotada.

Es necesario entonces un mecanismo para poder informar la configuración de cada host y de toda entidad que participe en la red, para luego razonar en forma argumentativa sobre un potencial ataque. Para esta representación de conocimiento e inferencia se utiliza programación en lógica rebatible, *Defeasible Logic Programming* (DeLP).

Además se tratan de analizar brevemente los problemas derivados de la estructura del framework y de la lógica. Dichos problemas impactan negativamente sobre la seguridad, performance y escalabilidad del sistema propuesto.

1 Introducción

En el transcurso de la última década se puede observar una tendencia a la protección y seguridad de las redes de área local (LAN). Es claro que este hecho no es casual, y se lo puede atribuir al crecimiento masivo de las redes y la tecnología, que han llevado a exponer recursos a la red Internet, dando la posibilidad a diversos ataques remotos.

Los Sistemas de Detección de Intrusos (IDS), tratan de establecer un mecanismo automático para la detección de atacantes a través de Internet o la red local, analizando el tráfico que fluye entre nuestra red y el mundo exterior, así como la disponibilidad, y distribución de los recursos de nuestros sistemas.

Lo que se está tratando de establecer en esta línea de investigación, es un framework de IDS en un entorno de red LAN inicialmente para luego extenderlo a un sistema distribuido (cluster o grid), basado en representación de conocimiento y razonamiento rebatible, a través de Defeasible Logic Programming (DeLP). Esta lógica provee un mecanismo para representar

información común entre nuestros hosts dentro de la red y aplicar un razonamiento rebatible donde cada host o entidad de la red pueda argumentar sobre un potencial ataque, a partir del conocimiento que se posee.

La idea es crear una herramienta para la seguridad y la administración de sistemas, inteligente e independiente, que sea una evolución de las existentes, teniendo la capacidad de razonamiento entre los hosts, desligando al administrador de sistemas de la rutina diaria de revisar cada host, analizar logs y tráfico de la red, y demás mecanismos de seguridad. Por medio de un portal web se puede crear una interfaz para que el administrador agregue un conjunto de reglas que implementen firewalls, ruteo, y demás políticas. Éstas reglas constituyen parte del razonamiento en conjunto, partiendo de los aportes de los nodos participantes del sistema.

2 Framework

Como se menciona en la sección anterior, necesitamos representar información de cada host en una base de conocimiento común, y de allí deducir si somos víctimas de un posible ataque. Para esto es necesario recolectar información de cada host referente a la configuración del sistema, versiones de software que tiene instalado, *daemons* que son accesibles desde otros hosts, analizar los logs locales, etc. También el administrador de sistemas podrá establecer políticas, reglas, y aspectos de los sistemas que puedan ser relevantes para el mecanismo de razonamiento.

Podemos utilizar herramientas complejas que puedan realizar la tarea de recolectar información local de cada nodo, entregando en un formato particular la salida de cada análisis del sistema. Una de las herramientas más reconocidas y prácticas de IDS local es SNORT [1].

Otras herramientas que presentan información y que podemos corresponder con la información de cada host, son las comunidades de reportes de bugs. Estas comunidades son bases de datos que presentan información estructurada sobre vulnerabilidades, detalles de las mismas, posibles ataques, y parches sobre un diverso conjunto de aplicaciones, y que tienen la particularidad de estar actualizadas con los nuevos problemas que las mismas comunidades van reportando.

Una vez que la información es procesada en cada host, podemos integrarla a la base de conocimientos de la lógica rebatible, sobre la cuál podremos hacer la inferencia. Notemos que aquí la información que provee cada host será fundamental para tomar decisiones sobre posibles ataques o vulnerabilidades en el sistema (Figura 1).

La base de conocimientos puede estar ubicada en un host, con lo cual mantenemos la consistencia de los datos de forma simple, o bien podemos mantenerla replicada en cada cada host, con un mecanismo que mantenga la consistencia de los datos, facilitando la tolerancia a fallas en la red y en cualquier nodo del sistema. Estos inconvenientes se pueden resolver analizando protocolos conocidos provenientes de sistemas distribuidos y algoritmos de replicación, y adaptándolos al contexto actual [2]. Dado que la información puede ser redundante, es necesario que sea optimizada antes de ser agregada a la base de conocimientos para evitar sobrecargas en los recursos del framework.

Este esquema de nodos que se comunican para inferir sobre posibles ataques nos da un panorama de una red de computadoras que cooperan en un entorno distribuido para extender la seguridad perimetral del sistema. Con esta visión podríamos extender el framework para trabajar sobre grids combinando esta tecnología propuesta con grid services.

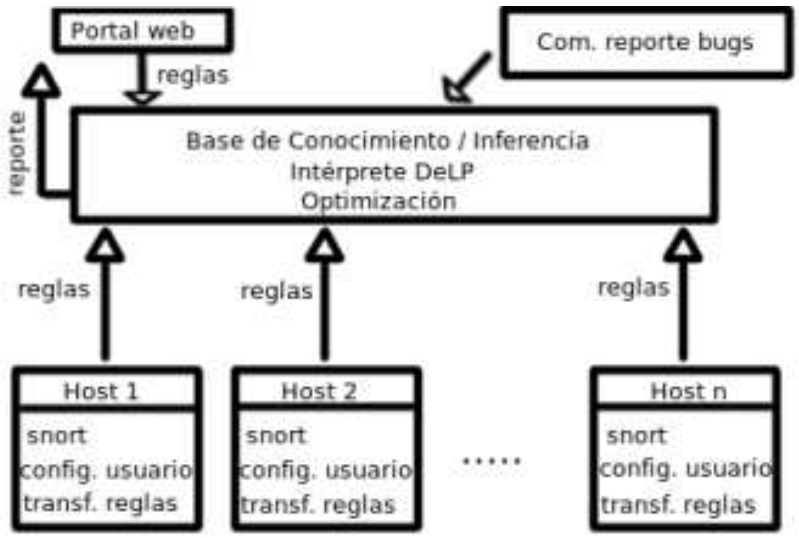

Figura 1: Framework IDS-DeLP

3 Inferencia

Uno de los aspectos más importantes de este framework es la posibilidad de inferir un posible ataque por medio de la argumentación. Este tipo de inferencia permite que las decisiones se tomen en conjunto con todos los nodos de la red, y que una posible determinación de ataque de un host sea refutada por un conjunto de otros nodos debido a que el conocimiento que estos tienen de la vulnerabilidad es mas específico. Si bien la idea de representar información de varios hosts para poder inferir posibles ataques, fue presentado en aplicaciones e investigaciones anteriores, e.g., [3], no existe una aplicación que pueda hacer inferencia utilizando argumentaciones contradictorias, sino que lo hacen por medio de la lógica de primer orden, que presenta falencias en el escenario propuesto, ya que tenemos una inconsistencia cuando un host argumenta un ataque por un razón, y otro host contradice esta suposición.

DeLP permite tener información que no es mutuamente consistente, pero que posibilita argumentar sobre un hecho o atacar un argumento con un contra-argumento. Así nuestros hosts podrán ir argumentando sobre un posible ataque, hasta que se llegue a un argumento que gane sobre los demás. DeLP presenta una formalización de la argumentación rebatible y un mecanismo de inferencia que utilizaremos en el framework [4]. DeLP cuenta con el criterio de especificidad para decidir si un argumento rebate a otro. Es decir, el argumento que presente mas información, pues tiene más conocimiento sobre el problema, será el que gane cuando existan argumentos contradictorios. Este criterio de especificidad, puede ser modificado o pertenecer a un conjunto de criterios que resulten adecuados al momento de adaptar esta lógica al framework.

La información contenida en la base de conocimiento común del framework puede ser ejecutada en un intérprete DeLP para realizar la inferencia, y decidir si un argumento es ganador y de allí deducir la validez de un posible ataque. Este intérprete puede estar ubicado en un host remoto o estar integrado en un nodo de la red.

Gracias a trabajos previos del Laboratorio de Investigación y Desarrollo en Inteligencia Artificial (LIDIA) de la Universidad Nacional del Sur [5], es factible contar con un intérprete DeLP que dada una base de conocimiento conteniendo argumentos, pueda inferir una consulta sobre un argumento, decidiendo si éste es un argumento ganador, o es derrotado por un contra-argumento.

4 Performance y Escalabilidad

La performance del sistema es crucial, y se puede mejorar en varios aspectos. Principalmente sobre la optimización de la información y eliminación de información redundante u obsoleta. Para esto, cada host puede plantear un mecanismo de optimización sobre la información que va a presentar a la base de conocimientos. Como la información es procesada en el host para luego ser incluida en la base de conocimientos común, argumentos para la lógica que estamos empleando, puede haber más de una regla que argumente lo mismo sin necesariamente ser mas específica. La base de conocimientos también puede ser optimizada en busca de argumentos obsoletos, por ejemplo que nunca ganarán, o que no van a ser aplicados en ninguna línea de argumentación. El sistema de comunicación también puede ser optimizado, y sin descartar principios de seguridad, tratar de utilizar protocolos de comunicación que mejoren la velocidad, por ejemplo UDP o protocolos especialmente optimizados dentro de la red local.

El framework se podría extender a grandes sistemas distribuidos, sin pérdida de escalabilidad, donde la seguridad es crucial y la detección de posibles vulnerabilidades es un requerimiento crítico debido a que los nodos participantes pueden encontrarse en distintos dominios administrativos y por ende bajo distintas políticas de seguridad. La adaptación deberá ser progresiva para poder solventar los problemas de la nueva topología que se intenta cubrir, primero con dominios pequeños, redes locales con pocos servidores y máquinas de escritorio, luego con clusters de mediana escala, y después escalando a sistemas distribuidos de mayor complejidad, conformados por distintos dominios administrativos.

5 Trabajos Futuros

Una vez que el framework esté adaptado a toda la red local para la detección de intrusos en un ambiente de servidores y máquinas de escritorio, será importante escalarlo a un sistema distribuido, como el cluster del Laboratorio de Investigación en Sistemas Distribuidos (LISiDi) de la Universidad Nacional del Sur [6], donde se presentarán nuevos protocolos y ambientes de trabajo [7].

Luego de llegar a integrarlo en el cluster, otro objetivo importante es adaptarlo como herramienta de seguridad entre los clusters que formen un grid. La seguridad de un grid es un factor crítico, ya que el medio de comunicación que interconecta los clusters que lo integran, suele ser la WWW, medio que representa la mayor amenaza en seguridad para los sistemas actuales.

Una idea para un próximo trabajo, relacionada con la seguridad de este sistema, es el desarrollo de un protocolo seguro que permita garantizar la autenticidad de un host que envía información a la base de conocimientos, así como la integridad misma de los datos enviados. Podremos así saber si la información que un host está tratando de actualizar en la base de conocimiento corresponde a un integrante legítimo del framework. Muchos protocolos de autenticidad existen, e integrar y adaptar uno puede ser un trabajo bastante complejo. Podríamos pensar en este esquema con Kerberos, como servidor de autenticación primario [8].

También podemos extender la seguridad del framework para los protocolos que integran la comunicación desde la base de conocimientos hacia el intérprete DeLP, cuando deseamos realizar la inferencia a partir de la información de la base de conocimiento. Esta comunicación es vulnerable a ataques sobre la red, ya que, como se mencionó en la sección anterior, el intérprete puede estar en un nodo fuera de la red. Podríamos entonces utilizar protocolos de autenticación, integrándolo a la propuesta anterior de Kerberos.

Una posible ampliación del framework puede realizarse mediante la creación e integración

de nuevas herramientas de seguridad que utilicen el mecanismo de inferencia presentado, y que se adapten en ciertos sectores del mismo. Una meta es poder agregar a cada host un mecanismo de seguridad de ejecución de comandos. Para lograrlo debemos razonar acerca del tipo de comando, y deducir si el usuario actual podría causar daño con la acción requerida. Este planteo presenta similitudes con un *shell* restringido, pero agrega inteligencia y aprendizaje a través del razonamiento planteado anteriormente.

Bibliografía

[1] M. Roesch, "Snort, intrusion detection system," *http://www.snort.org*.

[2] J. Echaiz and J. Ardenghi, "Extending an SSI Cluster for Resource Discovery in Grid Computing," in *GCC*, pp. 287–293, IEEE Computer Society, 2006.

[3] X. Ou, S. Govindavajhala, and A. W. Appel, "MulVAL: A Logic-based Network Security Analyzer," in *Proceedings of the 14th USENIX Security Symposium*, USENIX, Aug. 2005.

[4] A. J. García and G. R. Simari, "Defeasible Logic Programming: An Argumentative Approach," *TPLP*, vol. 4, no. 1-2, pp. 95–138, 2004.

[5] LIDIA, "Laboratorio de Investigación y Desarrollo en Inteligencia Artificial, Departamento de Ciencias e Ingeniería de la Computación de la Universidad Nacional del Sur." http://lidia.cs.uns.edu.ar.

[6] LISiDi, "Laboratorio de Investigación en Sistemas Distribuidos, Departamento de Ciencias e Ingeniería de la Computación de la Universidad Nacional del Sur." http://lisidi.cs.uns.edu.ar.

[7] S. Davicino, J. Echaiz, and J. Ardenghi, "Una Alternativa Económica para la Implementación de Servicios Web Localmente Distribuidos," *CACIC 2003*, pp. 459–470, Oct. 2003.

[8] A. Wachsmann, "Centralized authentication with Kerberos 5, Part I," *Linux Journal*, vol. 2005, pp. 6–6, Feb. 2005.