JNIC2016

II Jornadas Nacionales de Investigación en Ciberseguridad

Granada 15–17 de junio

GOBIERNO DE ESPAÑA
MINISTERIO DE INDUSTRIA, ENERGÍA Y TURISMO

10 incibe_
2006-2016
TRABAJANDO POR LA CONFIANZA DIGITAL

ugr Universidad de Granada

Sistema de detección de fases de ataque basado en Modelos Ocultos de Markov

Pilar Holgado
Departamento de Ingeniería y Sistemas Telemáticos.
Universidad Politécnica de Madrid Avenida Complutense, 30, 28040, Madrid.
pilarholgado@dit.upm.es

Víctor A. Villagrá
Departamento de Ingeniería y Sistemas Telemáticos.
Universidad Politécnica de Madrid Avenida Complutense, 30, 28040, Madrid.
villagra@dit.upm.es

Abstract- **La detección temprana de intrusiones es una tarea importante para anticiparnos a los pasos de los atacantes. En este artículo, se usan Modelos ocultos de Markov para detectar la fase de ataque de un ataque multi-paso utilizando distintas alertas reportadas por los IDSs. Para lograr este objetivo, hay que realizar el diseño del modelo y una fase de entrenamiento previa de manera totalmente off-line, basado en observaciones correspondientes a cada paso del ataque. El resultado final del modelo, podrá ser utilizado para la predicción de intrusiones con el objetivo de integrarlo en otros sistemas como Sistemas de Respuesta a Intrusiones o Sistemas de Gestión Dinámica del Riesgo.**

***Index Terms*- Aprendizaje Automático, Modelos Ocultos de Markov, Ataques multi-paso, Denegación de Servicio Distribuida.**

Tipo de contribución: *Investigación en desarrollo (Este artículo es un resumen de uno más extendido que se publicará en breve en una revista)*

I. Introducción

Actualmente todas las organizaciones están cada vez más preocupadas por posibles ataques de ciberseguridad en sus sistemas. Por ello, las empresas solicitan cada vez más sistemas basados en detección y análisis de estas amenazas. Una técnica que pude ayudar a estos sistemas a reaccionar antes de que la red sea comprometida es la predicción de intrusiones.

Un escenario de ataque o un ataque multi-paso [1] es un conjunto de actividades maliciosas llevadas a cabo por el mismo atacante para conseguir un objetivo específico. Estos pasos pueden ser usados para la predicción temprana de ataques. En este artículo se propone un método matemático para conseguir dicha predicción.

En concreto, proponemos el uso de Modelos Ocultos de Markov (HMM, Hidden Markov Model) [2] para la predicción de intrusiones. Un HMM está compuesto por dos procesos estocásticos, un proceso estocástico observable y otro no observable (oculto). El proceso estocástico oculto se puede observar a través del procesos estocástico que produce la secuencia de observaciones.

En este caso, una cadena compuesta por estados de ataque representa el proceso estocástico oculto y las observaciones se corresponden con las alertas producidas por los atacantes. Para obtener dichas alertas, usamos un sistema de detección de intrusiones (IDS) [3] que envía las alertas detectadas al sistema de predicción. Además, formateamos las alertas a estructura IDMEF (Intrusion Detection Message Exchange Format) [4]. Este formato proporciona un lenguaje común para la representación de las alertas.

Para conseguir las observaciones necesarias para el HMM es necesario realizar un proceso de clusterización de las alertas obtenidas del IDS. Las observaciones se componen de una etiqueta y una severidad. Para obtener las etiquetas hay que definir un conjunto de palabras obtenidas tras el proceso de clusterización de uno de los parámetros del informe de CVE (Common Vulnerabilities and Exposures) [5]. La severidad se coge directamente de la información proporcionada por la alerta.

Una vez definido el HMM, es necesario un proceso de entrenamiento para adaptarnos a los pasos de los distintos ataques. Tras esto, el HMM está preparado para predecir el estado del ataque multi-paso.

La predicción temprana de ataques puede ser utilizada en diferentes sistemas de seguridad, como por ejemplo, un Sistema de Gestión Dinámica del Riesgo o un Sistema de Respuesta a Intrusiones (IRS). En concreto, se puede incluir el método de predicción propuesto en un IRS Autónomo (AIRS) para ejecutar respuestas proactivas antes de que el ataque llegue a etapas finales que provocan un mayor peligro en la organización. En el caso de integrar el sistema predictivo a un Sistema de Gestión Dinámica del Riesgo, la salida del sistema se puede utilizar para la modificación del nivel de riesgo, dependiendo de la fase en la que se encuentre el ataque.

El artículo se organiza de la siguiente manera. La sección II trata sobre el estado del arte en detección y predicción de ataques multi-paso aplicando distintos modelos de Aprendizaje Automático. La sección III explica el Modelo Oculto de Markov y los parámetros necesarios para la detección de la fase de la intrusión. La detección de ataques multi-paso necesita una fase de entrenamiento previa comentada en la sección IV. Finalmente la sección V indica las tareas necesarias para conseguir el estado del ataque y en la sección VI se termina con conclusiones.

II. Trabajos relacionados

La predicción de intrusiones y las técnicas de correlación son un concepto clave para la prevención, detección y respuesta a intrusiones. Actualmente, se han realizado algunas investigaciones sobre este tema.

La predicción de intrusiones basada en HMM es realizada

en [6], [7], [8] y [9]. Estas investigaciones tienen en común que realizan una definición similar de los parámetros del HMM, siendo estos parámetros la clave para predicción. La principal diferencia con nuestra propuesta es esta definición del modelo. Ellos consideran los estados ocultos como diferentes estados de riesgo del sistema, en cambio, nosotros asignamos los estados ocultos a fases comunes de un determinado tipo de ataque, de esta manera conseguimos entrenar un modelo para cada tipo de intrusión y así poder ejecutar la respuesta que mejor se ajuste al ataque. Ajustarse tanto a la respuesta no es viable en el caso de asignar los estados de la cadena de Markov al riesgo ya que no es posible saber cuál es la intrusión que está afectando al sistema. En estos estudios previos, las observaciones del HMM se corresponden con el parámetro de severidad de las alertas obtenidas de un IDS o con una función que utilice dicho valor. Al igual que nosotros, utilizan las alertas obtenidas desde los IDSs como observación, pero en nuestro caso, ajustamos también el tipo de alerta clusterizando cada una de ellas a una etiqueta basada en el informe de vulnerabilidades teniendo en cuenta el CVE de la alerta, evitando al mismo tiempo el sobreajuste del HMM. En conclusión, nuestra metodología puede conseguir una mayor precisión que los trabajos previos porque predecimos los pasos de una intrusión específica mediante el entrenamiento del HMM con estados comunes de intrusión y observaciones basadas en etiquetas a partir del CVE de la alerta.

En [10] se propone el uso de HCPN (Hidden Colored Petri-Net) para predecir el siguiente objetivo del atacante. Usa un modelo matemático como en nuestro caso, pero su objetivo es la optimización y correlación de alertas de IDSs para reducir el ratio de falsos positivos y negativos, así que ellos no predicen intrusiones.

Usar procesos de Markov para realizar un AIRS se propone en [11]. En concreto ellos aplican un POMDP (Partially Observable Competitive Markov Decision Process) con un juego estocástico de Stackelberg de dos jugadores para decidir la mejor respuesta. A diferencia de nosotros, su propuesta no intenta predecir ataques antes de que sucedan, es decir, no realizan respuestas proactivas a intrusiones.

Detección de ataques de DDoS (Distributed Denial of Service) usando métodos matemáticos se propuso en [12]. En concreto, Katkar et al. utiliza el clasificador Bayesiano como modelo matemático.

Otros enfoques sobre predicción de escenarios de ataque son los propuestos en [13] y [14]. En este último caso, los autores usan una Máquina de Estados Finitos (FSM, Finite States Machine) para ataques multi-paso en un sistema de respuestas.

III. Definición del Modelo Oculto de Markov

Un HMM puede ser descrito como dos procesos estocásticos. El proceso oculto es representado por la variable aleatoria $x(t)$, y se corresponde con los estados de la cadena de Markov. El proceso observable, representado por la variable aleatoria $y(t)$, se corresponde con las alertas que llegan desde el IDS.

En este caso, t representa la llegada de las alertas desde IDSs en tiempo discreto.

Formalmente un HMM discreto de primer orden se define como $\lambda = (\sum, S, P, Q, \pi)$. Estos parámetros y su definición para nuestro objetivo se indican a continuación:

- Las observaciones, $\sum = \{v_1, v_2, \dots, v_M\}$, de la cadena de Markov en nuestro caso se basan en las distintas alertas de IDSs clusterizadas mas una severidad.
- Los estados, $S = \{s_1, s_2, \dots, s_N\}$, se corresponden con el conjunto de pasos comunes de un ataque concreto.
- La matriz de probabilidad de transición, $P = \{p_{ij}\}_{NxN}, p_{ij} = P(x_t = j | x_{t-1} = i)$, describe las transiciones entre los distintos estados de la cadena S de acuerdo con las alertas que llegan desde el IDS.
- Una matriz de probabilidad de observación tiene un vector de probabilidad para cada estado $Q = \{q_1, q_2, \dots, q_M\}$. Este vector indica la probabilidad de observar las diferentes alertas si el ataque se encuentra en un estado concreto.
- El vector de distribución inicial, $\pi = \{\pi_1, \pi_2, \dots, \pi_N\}$, indica la probabilidad de cada uno de los estados de ser el estado inicial cuando comienza la intrusión.

Las observaciones transformadas desde IDSs de un ataque multi-paso en particular son representadas mediante la secuencia de observación $O = \{o_1\ o_2\ \dots o_T\}$ donde cualquier o_t de la secuencia pertenece a un símbolo de la cadena $\sum$.

En nuestra propuesta, calcularemos una matriz de transición y de observación para cada uno de los tipos de intrusión en estudio y así ajustar las transiciones a cada paso de ataque. De esta manera conseguiremos anticiparnos al atacante ejecutando la mejor respuesta teniendo en cuenta el estado y el tipo de intrusión. Para ello, nosotros tendremos un modelo λ_k para cada tipo de ataque multi-paso.

IV. Entrenamiento del Modelo Oculto de Markov

Una de las tareas más importantes a la hora de utilizar un modelo de Markov es ajustar la probabilidad de las matrices P y Q para que se adecue al objetivo final del HMM de tal manera que maximice la probabilidad de la secuencia de observación dado el modelo concreto, $P(O|\lambda_k)$.

En nuestro caso, se necesita el entrenamiento de una matriz de transición, unos vectores de observación para cada estado y de un vector de distribución inicial para cada tipo de ataque.

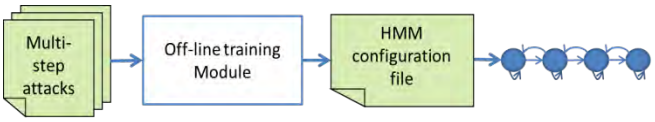

Fig. 1. Entrenamiento del HMM.

La Fig. 1 representa las etapas necesarias para el entrenamiento de un HMM. A continuación se explica en más detalle cada uno de estos módulos.

Multi-step Attack files almacena los distintos escenarios de ataque para cada una de las intrusiones en estudio. En concreto se corresponden con ficheros de trazas PCAP.

Off-line Training Module representa la fase de entrenamiento de los parámetros necesarios para el modelo. Para el aprendizaje hay diferentes algoritmos no supervisados como: Baum-Welch, EM (Expectation Maximization), GEM (Generalized EM) y el método del descenso del gradiente. Además es posible realizar el entrenamiento de una manera supervisada usando métodos estadísticos.

HMM configuration file es el fichero de configuración resultante para una intrusión concreta tras la aplicación de un algoritmo de entrenamiento. Almacena los parámetros

```
Hmm v1.0

NbStates 3
NObservations 18

State
Pi 0.7384105960264901
A 1.0 0.004484304932735426 0.004484304932735426
DiscreteOPDF 0.0134 4.4848 4.4843 4.4843 4.4843 4.4843 0.6457 4.4843 4.4843 4.4843 4.4843 4.4843 4.4234 0.3408 4.4843 4.4843 4.4843 4.4843
State
Pi 0.17880794701986755
A 0.018518518518518517 0.9629629629629629 0.037037037037037035
DiscreteOPDF 1.8518 1.8514 0.3333 1.8518 1.8518 1.8518 1.8518 1.8518 1.8518 1.8518 1.8518 1.8518 1.8518 0.6666 1.8518 1.8518 1.8518 1.8518
State
Pi 0.08278145695364239
A 0.04 0.04 0.92
DiscreteOPDF 4.0E-17 4.0E-17 4.0E-17 4.0E-17 4.0E-17 4.0E-17 4.0E-17 0.72 4.0E-17 4.0E-17 0.04 4.0E-17 4.0E-17 4.0E-17 0.2 4.0E-17 0.04 4.0E-17
```

Fig. 2. Fichero de configuración del HMM.

necesarios para la definición del HMM, como es el número de estados o las matrices de probabilidad.

La implementación de la arquitectura de entrenamiento del HMM es un programa en Java utilizando la librería jahmm [15]. Esta librería incluye distintos algoritmos utilizados por los HMMs y la estructura de datos para las matrices de probabilidad.

El formato del fichero de configuración del HMM está basado en el formato propuesto por esta librería de Java. La Fig. 2 muestra un ejemplo de un fichero de configuración con nuestro diseño de HMM. Los parámetros importantes y su correspondencia con la definición formal, $\lambda = (\sum, S, P, Q, \pi)$ se indica a continuación:

- NbStates: Número de estados de la cadena de Markov, S.
- Nobservations: número de observaciones posibles en cada estado para el HMM, $\sum$.
- P_i: Probabilidad de que ese estado sea inicial. El vector de distribución inicial (π) se obtiene juntando todos estos valores.
- A: Vector de probabilidad de transición de un estado. Hay un vector para cada uno de los estados del HMM, si juntamos todos ellos obtenemos la matriz de probabilidad de transición, P.
- DiscreteOPDF: Se corresponde con el tipo de dato de la librería jahmm utilizado para definir las observaciones. Esta librería tiene varios tipos de datos para la representación de observaciones (Enteros, Discretos, Reales, …). En nuestro caso, las observaciones están representadas por un tipo Discreto basado en la clusterización de las posibles alertas de un IDS y las severidad. Los valores que aparecen en esta misma línea representan el vector de probabilidad de observación. Juntando todos estos vectores, obtendremos la matriz de probabilidad de observación, Q.

V. Detección de la fase de ataque

El objetivo de esta propuesta es detectar la fase actual de un ataque multi-paso mientras la intrusión está en curso. Para esto necesitamos tener un conjunto de pasos de ataque para el desarrollo del sistema de predicción de ataques multi-paso.

El primer paso es por tanto seleccionar el número de estados del HMM. Estos estados se corresponden con las fases comunes de un ataque, como ya se ha mencionado en la Sección III. Como caso de estudio, utilizamos el escenario LLDDOS1.0 de DARPA 2000 [16]. Este escenario ha sido seleccionado ya que es el escenario que se utiliza en todas las publicaciones previas y porque necesitamos tener solo trazas de ataque individualizadas para cada uno de los estados, es decir un fichero PCAP para cada etapa de ataque. Los estados resultantes se muestran en la Fig. 3. Hay que tener en cuenta que las conexiones entre estados han sido simplificadas, ya que se utiliza un HMM totalmente conectado.

Estos estados se obtienen generalizando las distintas fases del ataque de DDoS basado en el escenario real publicado en DARPA 2000.

A continuación se indica a modo de ejemplo la correspondencia de cada una de estas fases para el escenario 1.0 de DARPA2000:

1. Step 1: IPsweep de la red desde un sitio remoto.
2. Step 2: Escaneo de IPs para buscar el demonio sadmind ejecutado en máquinas Solaris.
3. Step 3: Explotación de una vulnerabilidad del demonio sadmind, intentando tanto en todas las máquinas se tenga o no éxito.
4. Step 4: Instalación del trojano, mstream DDoS software, sobre tres máquinas de AFB
5. Step 5: Ejecución de DDoS.

El segundo paso es la selección de un conjunto de observaciones para la cadena de Markov. Estas observaciones son obtenidas desde las alertas reportadas por los IDSs como se comentó en la sección III. No es posible una observación por alerta debido a la gran cantidad de alertas distintas que puede enviar un IDS. Si incluyéramos todas ellas como observación en un HMM, tanto el tamaño de las matrices de probabilidad, como el tiempo de procesado de cada alerta con el sistema en producción, se incrementaría significativamente. Por ello proponemos la realización de una correspondencia de cada una de las alertas a una serie de etiquetas. Estas etiquetas serán obtenidas de un proceso de clusterización basado en el informe de CVE. Posteriormente, se combinará esta etiqueta con la severidad de la alerta para conformar la observación que servirá como entrada para el HMM. Con esta selección de observaciones conseguimos que el sistema sea totalmente genérico ante cualquier ataque.

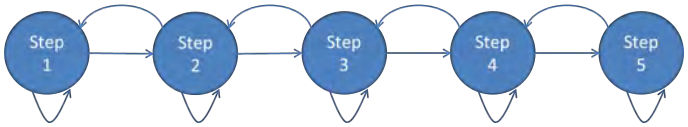

Fig. 3. Cadena de Markov.

La última tarea es entrenar el HMM teniendo en cuenta los estados y las observaciones definidas. Como resultado obtendremos las matrices de probabilidad de transición y observación, y el vector de distribución inicial. Para la realización de una comparativa de resultados, entrenaremos el HMM con un algoritmo supervisado y otro no supervisado. En este último caso, seleccionamos el algoritmo de Baum-

Welch [17] ya que es el algoritmo más utilizado para este método de aprendizaje automático.

Una vez entrenado el HMM está preparado para detectar las fases de una DDoS en progreso. El algoritmo de Viterbi [18] se encargará de obtener la secuencia más probable de estados cada vez que llega una alerta al sistema y por tanto sabremos la fase en la que se encuentra la intrusión. De esta manera, podemos identificar la intrusión final y saber cuántos pasos le quedan al atacante para finalizar el ataque. Con este conocimiento sistemas de respuesta o sistemas de evaluación del riesgo pueden anticiparse a la intrusión final evitando daños en los activos de la organización.

Por último, se realizarán medidas de rendimiento y eficiencia del sistema final y evaluaremos la viabilidad del uso de HMM en la predicción de ataques multi-paso en entornos en tiempo real.

VI. Conclusiones

El objetivo de este artículo es mostrar el diseño de un HMM para la predicción de la fase de ataque de la intrusión final. Para ello, se define el modelo matemático del HMM y la selección de los valores de los parámetros acorde a nuestro objetivo. Además, se determinan los mecanismos de entrenamiento basado en aprendizaje automático que se van a aplicar.

Este sistema puede ayudar a la detección temprana de intrusiones mediante la predicción de los pasos del atacante. En nuestra propuesta de HMM, un proceso estocástico no observable puede predecirse basado en distintas observaciones conseguidas desde distintos IDSs.

La primera tarea es seleccionar el número de estados para construir la cadena de Markov y almacenar los distintos tipos de observaciones posibles del sistema. A través de una etapa de entrenamiento se conseguirán las matrices de probabilidad que más se ajusten a la intrusión. El entrenamiento se realizará mediante dos algoritmos, supervisado y no supervisado, para comparar su efectividad.

Una vez que tenemos el HMM entrenado, el sistema está preparado para entrar en producción y aplicar el algoritmo de Viterbi cada vez que llegue una alerta desde un IDS. Como resultado obtendremos el estado más probable en el que se encuentra la intrusión y así podremos anticiparnos a la intrusión final. Estos cálculos se ejecutan en tiempo real, por lo que no afectan al tiempo de respuesta del sistema.

Una vez entrenado, el sistema de predicción puede ser desplegado para cualquier organización sin necesidad de ninguna modificación adicional. Y puede ser utilizado en muchos sistemas existentes como Sistemas de Respuesta a Intrusiones para la ejecutar respuestas proactivas, como en Sistemas de Evaluación Dinámica del Riesgo modificando el nivel de riesgo del entorno.

Una gran ventaja de esta propuesta con respecto a trabajos anteriores, es la capacidad de poder saber el tipo de intrusión y así poderse anticipar con una respuesta al ataque final. Como inconveniente, el diseño es más complejo e implica una mayor capacidad de almacenamiento y cómputo. El sistema debe tener la capacidad de almacenar cada una de las matrices de probabilidad para cada uno de los modelos. En cualquier caso, el número de modelos no se estima tan grande como para que no pueda ser almacenado en cualquier servidor sin necesidad de unos requerimientos especiales. En cuanto a la capacidad de cómputo, para garantizar unos niveles de rendimiento y prestaciones adecuados tras el entrenamiento, la solución ideal se basaría en paralelizar el cálculo de la predicción para cada uno de los tipos de intrusión a analizar de tal manera que la predicción no afectaría al tiempo real necesario para la respuesta.

Como trabajo futuro validaremos el sistema con varios ataques multi-paso, incluyendo también más escenarios de DDoS. Con todos estos escenarios podremos entrenar un HMM para cada una de las intrusiones consideradas. Además de calcular en qué fase de ataque nos encontramos, como trabajo futuro calcularemos la probabilidad de la predicción con el fin de mejorar la precisión del sistema.

Agradecimientos

Este trabajo ha sido financiado en parte con el apoyo del MINECO español (proyecto DHARMA, Dynamic Heterogeneous Threats Risk Management and Assessment, con código TIN2014-59023-C2-2-R) y por la comisión Europea (FEDER/ERDF).

Referencias

[1] Kristopher Kendall. “A Database of Computer Attacks for the Evaluation of Intrusion Detection Systems”. 1999 (phd)

[2] Lawrence R. Rabiner. “A tutorial on Hidden Markov Models and selected applications in speech recognition”. Proceedings of the IEEE, 1989

[3] S. Axelsson, Intrusion detection systems: A survey and taxonomy, Technical report Chalmers University of Technology, Goteborg, Sweden (2000).

[4] H. Debar, D. Curry, B. Feinstein. “The Intrusion Detection Message Exchange Format (IDMEF)”. IETF Request for Comments 4765, 2007.

[5] Mitre inc., [Online] http://www.cve.mitre.org/.

[6] Kajetil Haslum, Ajith Abraham, Svein Knapskog. “DIPS: A Framework for Distributed Intrusion Prediction and Prevention Using Hidden Markov Models and Online Fuzzy Risk”. IEEE, 2007

[7] Kjetil Haslum, Marie E. G. Moe, Svein J. Knapskog. “Real time intrusion prevention and Security Analysis of Networks using HMMs”. IEEE, 2008.

[8] Alireza Shameli Sendi, Michel Dagenais, Masoume Jabbarifar, Mario Couture. “Real time intrusion prediction based on Optimized Alerts with Hidden Markov Model. Journal of Networks”. Journal of Networks, 2012.

[9] H. A. Kholidy, A. Erradi, S. Abdelwahed, A. Azab, A finite state hidden markov model for predicting multistage attacks in cloud systems, in: In Dependable, Autonomic and Secure Computing (DASC). IEEE 12th International Conference on, 2014, pp. 14–19.

[10] Dong Yu, Deborah Frincke. “Improving the quality of alerts and predicting intruder’s next goal with Hidden colored Preti-Net”. Computer Networks, 2007.

[11] Saman A. Zonouz, Himanshu Khurana, William H. Sanders, Timothy M. Yardley. “RRE: A Game–Theoretic Intrusion Response and Recovery Engine.”. IEEE, 2013.

[12] V. Katkar, A. Zinjade, S. Dalvi, T. Bafna, R. Mahajan, Detection of dos/ddos attacks against http servers using naive bayesian, in: InComputing Communication Control and Automation (ICCUBEA), International Conference on. IEEE, 2015, pp. 280–285.

[13] Fayyad, Seraj, and Christoph Meinel. "Attack Scenario Prediction Methodology." Information Technology: New Generations (ITNG), 2013 Tenth International Conference on. IEEE, 2013.

[14] Alireza Shameli-Sendi, Julien Desfossez, Michel Dagenais, Masoume Jabbarifar. “A Retroactive-Burst Framework for Automated Intrusion Response System”. Journal of Computer Networks and communications, 2013.

[15] Jahmm - java library for hmm model and algorithms, [Online] https://code.google.com/p/jahmm/.

[16] Darpa - intrusion detection evaluation dataset, [Online] https://www.ll.mit.edu/ideval/data/2000/LLS_DDOS_1.0.html

[17] A. Dempster, N. M. Laird, D. Rubin, Maximum likelihood from incomplete data via the em algorithm, Journal of the royal statistical society. Series B (methodological) (1977) 1–38.

[18] G. D. Forney, The viterbi algorithm, in: Proceeding IEEE, 1973.