Diego Esteban Quintero Rey
Modelos estocásticos y simulación en computación y comunicaciones
ME01
Proyecto Final: Implementación de un enfoque basado en teoría de juegos para contrarrestar amenazas persistentes avanzadas

RESULTADOS

EXPERIMENTO 1:

Replicación del experimento de la sección 9 de [1].

Este experimento no depende de la topología del sistema, sino que se define una serie de exploits posibles en una infraestructura y los controles para mitigar las pérdidas.

Nota: Los controles son las estrategias del defensor y los exploits en este caso son las estrategias del atacante en el juego APT

Parámetros

- Estrategias del defensor: aplicación de parches, desactivación de servicios
- Estrategias del atacante: desbordamiento de búfer, acceso remoto
- Matriz de distribuciones de pérdida:

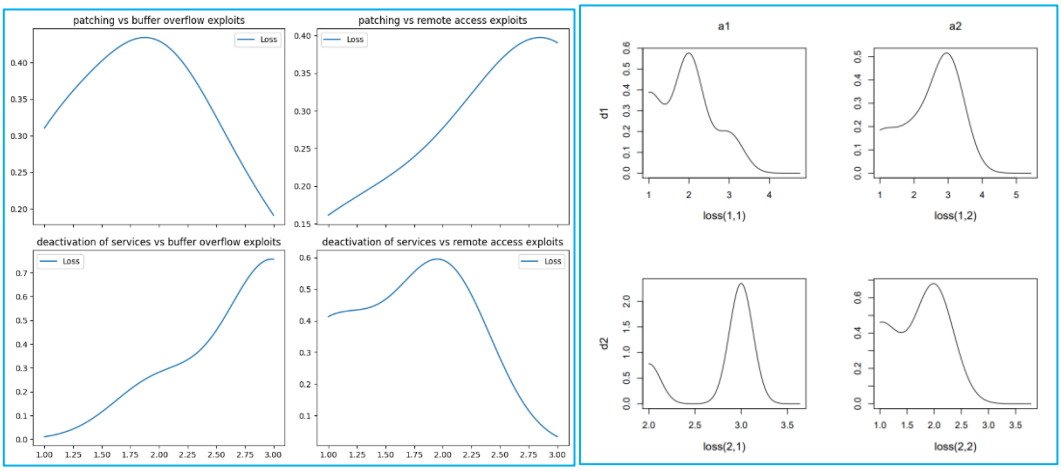
$$\begin{pmatrix} (2,3,1) & (1,1,3) \\ (0,1,3) & (2,3,0) \end{pmatrix}$$

Nota: Cada uno de los elementos de la matriz representa las frecuencias de una distribución categórica de tres categorías: baja pérdida, media pérdida y alta pérdida. Por ejemplo, la entrada $(2, 3, 1)$, significa que dos expertos dijeron que aplicar parches ante el desbordamiento de búfer es efectivo y reduce las pérdidas a un nivel bajo. Tres expertos dijeron que el efecto es medio y uno dijo que la estrategia de defensa no es muy efectiva y que las pérdidas son altas.

Resultados obtenidos

Matriz de estimaciones de densidad de kernel:

A la izquierda se encuentra las curvas generadas en la aplicación y a la derecha las que se presentan en el artículo [1] en la sección 9. Se observa una diferencia en la curva de la posición (2, 1).



Equilibrio lexicográfico de Nash:

Defensa óptima:

- Aplicación de parches el 78.4% del tiempo
- Desactivación de servicios el 21.6% del tiempo

Ataque (APT) esperado:

- Desbordamiento de búfer el 37.8% del tiempo
- Acceso remoto el 62.2% del tiempo

Estrategia	Resultado replicación	Resultado en el artículo
Aplicación de parches	78.4%	87.5%
Desactivación de servicios	21.6%	12.5%
Desbordamiento de búfer	37.8%	23.8%
Acceso remoto	62.2%	76.2%

EXPERIMENTO 2

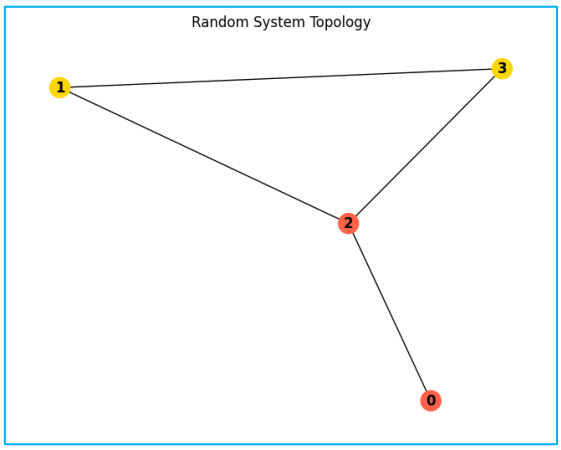
Este es un experimento dependiente de la topología.

Parámetros

- Tamaño de la topología del sistema: 4 nodos
- Nodo origen: 0
- Nodo objetivo: 1
- Número requerido de caminos de ataque: 4
- Número requerido de controles: 4
- Número de expertos: 10

Preparación de la topología y matriz de distribuciones

La topología resultante contiene dos enrutadores (nodos color rojo) y dos estaciones de trabajo (nodos color amarillo).



Los caminos de ataque seleccionados son los siguientes:

Camino de ataque 1:

- Filtrado de información con SSRF en el nodo 0
- Irrupción del cortafuegos en el nodo 2
- Escalado de privilegios en el nodo 3
- XSS en el nodo 1

Camino de ataque 2:

- Inyección de comandos en el nodo 0
- Irrupción del cortafuegos en el nodo 2
- XSS en el nodo 1

Camino de ataque 3

- Inyección de comandos en el nodo 0
- Irrupción del cortafuegos en el nodo 2
- Escalado de privilegios en el nodo 3
- Acceso no autorizado a datos sensibles en el nodo 1

Camino de ataque 4

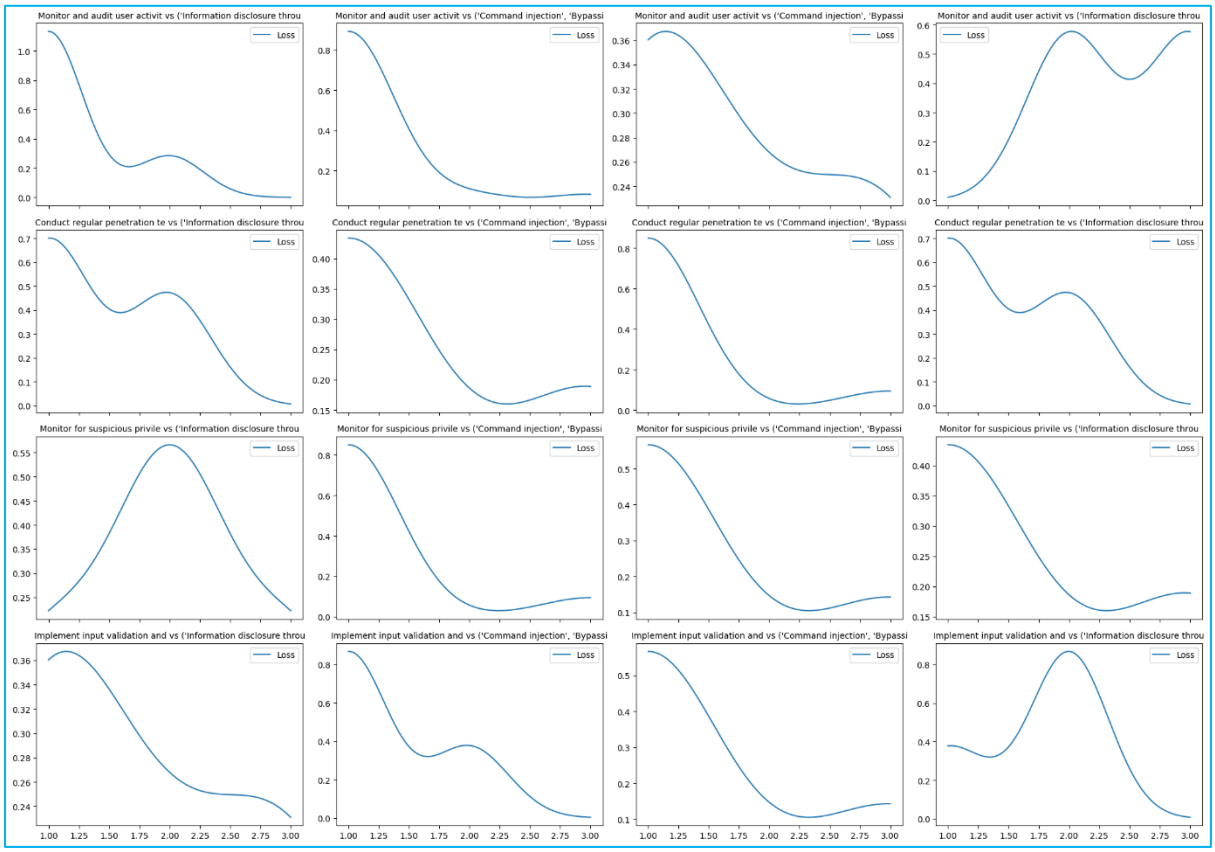
- Filtrado de información con SSRF en el nodo 0
- Irrupción del cortafuegos en el nodo 2
- Escalado de privilegios en el nodo 3
- Acceso no autorizado a datos sensibles en el nodo 1

Los controles seleccionados son:

- Monitoreo y auditoría de las actividades de los usuarios
- Conducción regular de pruebas de penetración
- Monitoreo de cambios de privilegio sospechosos
- Implementación de codificación de entradas y salidas

Resultados obtenidos

Matriz de estimaciones de densidad de kernel:



Equilibrio lexicográfico de Nash:

Defensa óptima:

- Monitoreo y auditoría de las actividades de los usuarios: 31.4% del tiempo
- Conducción regular de pruebas de penetración: 13.7% del tiempo
- Monitoreo de cambios de privilegio sospechosos: 0.1% del tiempo
- Implementación de codificación de entradas y salidas: 54.8% del tiempo

Ataque (APT) esperado:

- Camino 1 (SSRF, Cortafuegos, Privilegios, XSS): 73.9% del tiempo
- Camino 2 (Inyección, Cortafuegos, XSS): 5.7% del tiempo
- Camino 3 (Inyección, Cortafuegos, Privilegios, Acceso): 20.2% del tiempo
- Camino 4 (SSRF, Cortafuegos, Privilegios, Acceso): 0.2% del tiempo

EXPERIMENTO 3

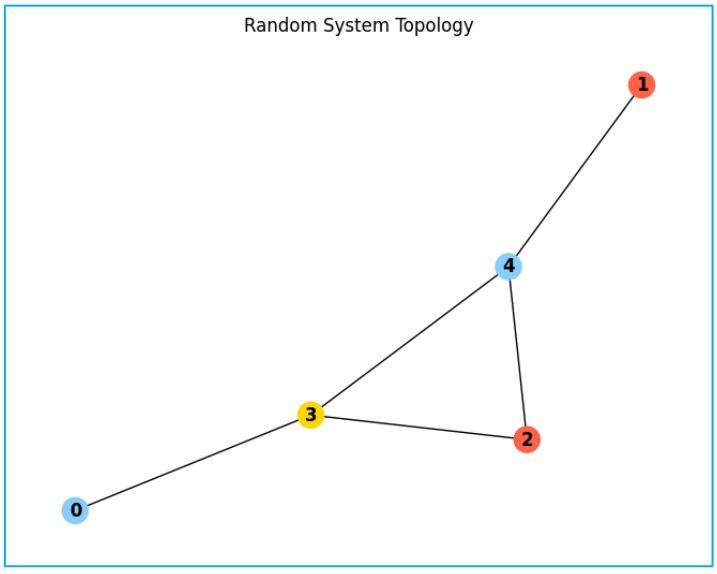
Este es un experimento dependiente de la topología.

Parámetros

- Tamaño de la topología del sistema: 5 nodos
- Nodo origen: 2
- Nodo objetivo: 4
- Número requerido de caminos de ataque: 3
- Número requerido de controles: 5
- Número de expertos: 10

Preparación de la topología y matriz de distribuciones

La topología resultante contiene dos enrutadores (nodos color rojo) y una estación de trabajo (nodos color amarillo) y dos servidores (nodos color azul).



Los caminos de ataque seleccionados son los siguientes:

Camino de ataque 1:

- Inyección SQL en el nodo 2
- Aprovechamiento de manejo inseguro de sesión en nodo 4

Camino de ataque 2:

- Inyección SQL en el nodo 2
- Ataque a componentes de terceros en el nodo 3
- Aprovechamiento de manejo inseguro de sesión en nodo 4

Camino de ataque 3:

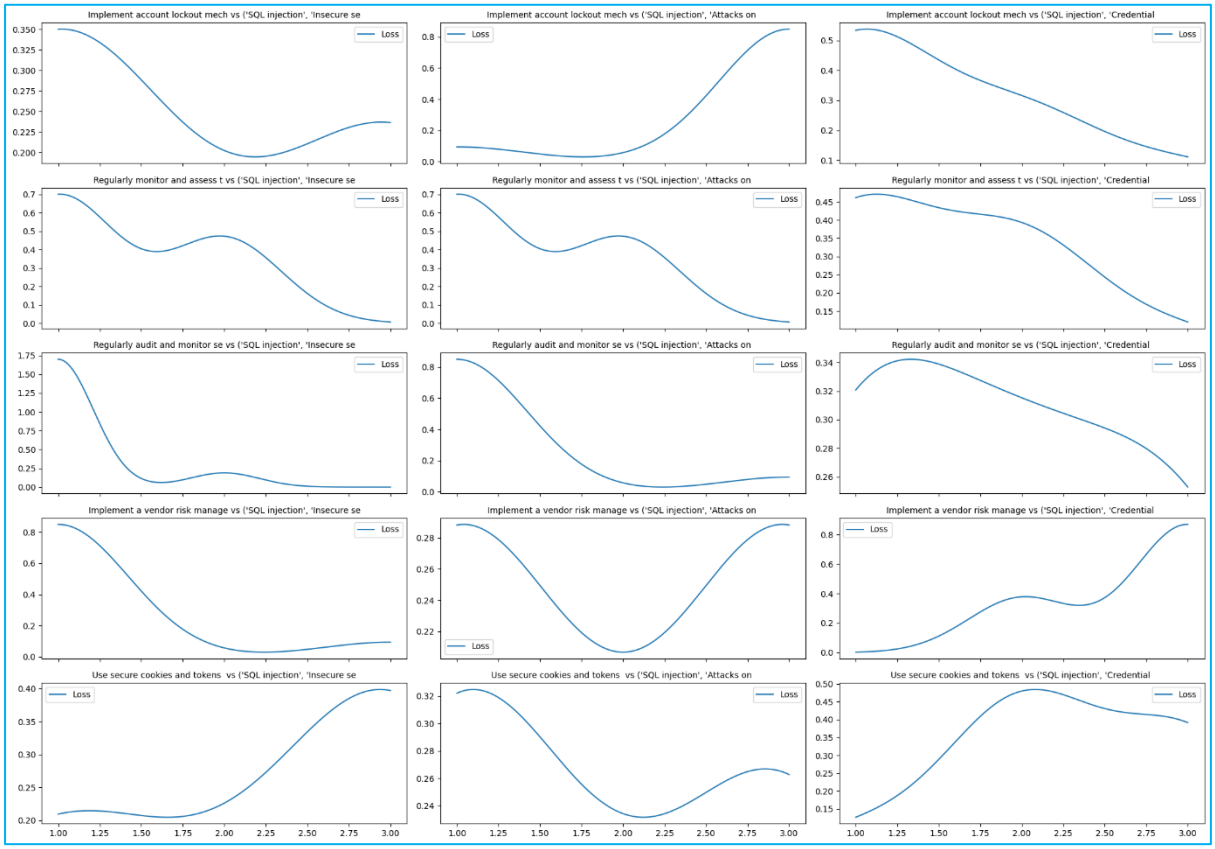
- Inyección SQL en el nodo 2
- Reutilización de credenciales robadas en el nodo 3
- Aprovechamiento de manejo inseguro de sesión en nodo 4

Los controles seleccionados son:

- Implementación de mecanismos de bloqueo de cuenta
- Monitoreo y evaluación de la seguridad de los componentes de terceros
- Monitoreo y auditoría de las actividades de los usuarios
- Implementación de un programa de gestión de riesgos de proveedores
- Utilización de cookies y tokens seguros para el manejo de sesiones

Resultados obtenidos

Matriz de estimaciones de densidad de kernel:



Equilibrio lexicográfico de Nash:

Defensa óptima:

- Implementación de mecanismos de bloqueo de cuenta: 16.1% del tiempo
- Monitoreo y evaluación de la seguridad de los componentes de terceros: 0.0% del tiempo
- Monitoreo y auditoría de las actividades de los usuarios: 0.0% del tiempo
- Implementación de un programa de gestión de riesgos de proveedores: 3.4% del tiempo
- Utilización de cookies y tokens seguros para el manejo de sesiones: 80.5% del tiempo

Ataque (APT) esperado:

- Camino 1 (Inyección SQL, Sesión Insegura): 47.8% del tiempo
- Camino 2 (Inyección SQL, Componentes de Terceros): 25.5% del tiempo
- Camino 3 (Inyección SQL, Credenciales Robadas, Sesión Insegura): 26.7% del tiempo

CONCLUSIONES

Para la implementación dependiente de la topología sería interesante implementar algún modelo de aprendizaje de máquina que haya sido entrenado sobre un conjunto de datos para predecir la pérdida que se espera obtener al implementar cierto control ante cierto vector de ataque. De esa manera se podría generar la distribución de la variable aleatoria de pérdida con ayuda de estos sistemas, permitiendo obtener soluciones más informadas. Puede ser posible incluso entrenar múltiples modelos sobre diferentes conjuntos de datos e implementar un mecanismo de votación similar a lo que harían los expertos, y de esta manera conformar las distribuciones.

Para el análisis topológico de vulnerabilidades puede ser buena implementar algún mecanismo basado en ciencia de redes para definir qué nodos serían más vulnerables ante cierto tipo de ataques y su grado de vulnerabilidad o de robustes. Esto sería particularmente útil en infraestructuras grandes.

Para infraestructuras grandes, es claro que resulta intratable implementar esta solución sobre todos los posibles caminos de ataque y para todos los pares de nodos de la red. Por lo tanto, utilizar herramientas de aprendizaje de máquina y ciencia de redes podría permitir acotar el espacio de análisis a solo aquellos nodos particularmente vulnerables (o incluso los más robustos, pues podrían ser útiles en términos de implementación de estrategias de defensa), y solo aquellos caminos y vectores de ataque que se consideren relevantes de acuerdo con las características de la infraestructura. Por otro lado, restringir la búsqueda a ataques generales que pueden ocurrir en cualquier parte del a infraestructura sin algún nodo objetivo puntual, también es una alternativa. Más conclusiones sobre esto pueden encontrarse en la sección de discusión de [1].

Así mismo como se menciona en la sección de generalizaciones de [1], vale la pena explorar la posibilidad de incluir otro tipo de variables en la implementación, como el costo, la disponibilidad de los sistemas, o la confidencialidad de la información, entre otros, para definir así la pérdida. Esto parece alcanzable si se tiene algún conjunto de datos con esas características y se entrena un modelo de aprendizaje de máquina que prediga la pérdida de acuerdo con esas variables. O se podría implementar un mecanismo de optimalidad de Pareto como se propone en [1].

En esta implementación, como punto a corregir, se incluirá en el futuro el nodo asociado a cada vector de ataque en los caminos de ataque, para facilitar la interpretación de los resultados. Esto es completamente necesario para grafos grandes.

REFERENCIAS

[1] Rass S, König S, Schauer S (2017) Defending Against Advanced Persistent Threats Using Game-Theory. PLOS ONE 12(1): e0168675. https://doi.org/10.1371/journal.pone.0168675