Universidad de Concepción
Dirección de Postgrado
Facultad de Ingeniería - Programa de Magíster en Ciencias de la Computación

MODELO DE SISTEMA DE DETECCIÓN DE INTRUSOS EN RED BASADO EN ESPECIES INDICADORAS ARTIFICIALES

Tesis para optar al grado de
MAGÍSTER EN CIENCIAS DE LA COMPUTACIÓN

POR
Matías Octavio Lermanda Sandoval
CONCEPCIÓN, CHILE
Marzo, 2021

Profesor guía: Pedro Pablo Pinacho Davidson
Departamento de Ingeniería Informática y Ciencias de la Computación
Facultad de Ingeniería
Universidad de Concepción

Urbi et Orbi

AGRADECIMIENTOS

A todos los que han sido parte de esto, ustedes saben quienes son.
To all who have been part of this, you know who you are.

Resumen

La cantidad de ataques informáticos a redes de computadores ha ido en aumento desde hace varios años, en este escenario se han desarrollado diversas soluciones al problema de la seguridad informática. Una de estas soluciones es la llamada ABS (*Artificial Bioindicators System*).

Este trabajo se basa en el uso y aplicación de ABS como herramienta para desarrollar un modelo que permita la detección rápida y confiable de amenazas informáticas.

La investigación se basa en una propuesta anterior del modelo de ABS, agregando las mejoras necesarias para superar los puntos débiles encontrados, en particular, la capacidad de evitar la contaminación de su conocimiento requerido para la detección y clasificación de amenazas, y la posibilidad de caracterizar nuevas amenazas que se presenten.

Para evaluar el desempeño comparativo de las mejoras propuestas, se realizaron pruebas con el dataset de DARPA .

Los resultados mostraron que en los experimentos realizados, el modelo ABS tarda 103.6 paquetes en promedio en reaccionar a una anomalía, mientras que el modelo propuesto en este trabajo tiene un tiempo de respuesta promedio de 1.8 paquetes, lo que representa menos de un 2 % del tiempo que tarda el modelo ABS, esto se debe a la adición de un mecanismo para caracterizar automáticamente las amenazas.

Índice general

Índice de figuras

Capítulo 1

Introducción

Cada año las empresas son objetivo de una gran cantidad de ataques informáticos. La cantidad de ataques de denegación de servicio con un ancho de banda superior a 1 Gbps aumentó en un 172 % en 2016 y se proyecta que aumente 2.5 veces para un total de 3.1 millones de ataques para el año 2021 [8]. Constantemente, a medida que se van descubriendo nuevas vulnerabilidades, se están inventando nuevos ataques de red, por lo que los sistemas de defensa deben mantenerse constantemente actualizados para que continúen siendo efectivos.

Primero, comenzando con los conceptos básicos, acerca del ABS [28] y los bioindicadores. En palabras simples, una especie indicadora (o bioindicador) no es más que aquella que puede definir una característica del medio a través de algún indicador, por ejemplo la presencia de ciertos organismos en cuerpos de agua puede ser un indicador de un tipo específico de contaminación. Esto ocurre de manera automática en la naturaleza mediante la selección natural, las especies se adaptan para sobrevivir a un medio ambiente cambiante y prosperar.

Un IDS (*Intrusion Detection System*) es aquel que se encarga de vigilar una red o conjunto de sistemas para detectar intrusiones o situaciones anormales, típicamente reportando estas situaciones a un administrador [18].

La mayoría de los sistemas IDS actuales están basados en soluciones estadísticas o de reconocimiento de patrones preestablecidos y, en general, estos comparten un problema en su desempeño respecto a nuevos ataques, los llamados Zero-day[1], un Zero-day usualmente exhibe patrones completamente nuevos, esto significa que cualquier sistema IDS que trate de compararlo con

[1] Ataque que permanece desconocido a las partes responsables de su posible mitigación.

una base de datos de ataques no va a poder reconocerlo como tal, por su elemento diferenciador. Este problema crítico es el que motiva la investigación aquí propuesta, mediante un modelo que sea capaz no solo de reconocer estas nuevas amenazas, si no que además, aprenderlas y caracterizarlas.

Para solucionar los problemas más comunes de los IDS, se propone la investigación y desarrollo de un modelo de IDS basado en ABS.

En el caso de un IDS, el medio ambiente corresponde al sistema monitoreado, tomando las características de la red y se simula la especie mediante agentes que, con un código genético y a través de procesos evolutivos como reproducción y mutación, pueden adaptarse al medio ambiente en un proceso que simula la selección natural. De esta forma, estos agentes se deberían adaptar, de manera natural, a las condiciones de la red, y cualquier cambio considerado significativo se reflejaría de alguna forma en la población de agentes. Este cambio en los agentes es lo que permite inferir ciertas características de la red.

Esta investigación se centrará específicamente en el uso de ABS como IDS para detectar y caracterizar tales anomalías. La motivación para elegir ABS está basada en las principales características positivas que presenta esta solución, las cuales serán resumidas a continuación.

Una de las características más importantes del ABS es su capacidad adaptativa. En contraste con soluciones enfocadas a patrones preestablecidos, el ABS tiene la facultad de adaptarse a condiciones cambiantes en el ambiente, esto significa que un medio ambiente que muestra condiciones de normalidad fluctuantes no presenta un riesgo importante para el correcto funcionamiento del sistema, un ABS es capaz de adaptarse a estas nuevas condiciones, sin marcarlas como una condición de anormalidad.

Otra característica positiva del ABS es que puede ser individualmente muy simple, pero presenta comportamientos complejos a nivel de la población completa. Esto permite que, mediante reglas de agentes muy simples, la población parezca obedecer a reglas complejas, la aparición de comportamientos complejos a partir de partes simples se conoce como *comportamiento emergente*.

En cuanto a características negativas del ABS, se puede decir que una desventaja que presenta es que requiere de un cierto período de adaptación inicial que debe estar desprovisto de ataque, esto es para que la población se acostumbre a las condiciones normales, de otro modo, se podría producir que la población termine caracterizando situaciones anómalas.

Si pensamos por ejemplo en un caso en que el período de adaptación ocurra durante la ejecución de un ataque, la población se adaptaría a esas condiciones, y cuando vuelva la normalidad no sabrían reconocerla como tal y la considerarían una anomalía, esto es un problema.

Otra desventaja es que los agentes se están adaptando constantemente, esto quiere decir que a medida que ocurre un ataque, la población se vuelve cada vez más insensible a este, a medida que la selección natural privilegia agentes que se puedan adaptar al ataque para sobrevivir.

Si se compara al ABS con un método tradicional de reconocimiento de patrones, se puede encontrar otra desventaja. Al ser un detector de anomalías, un ABS tradicional solo puede detectar aquello que no sea normal, todo lo que se salga de esta categoría recibe una etiqueta de ”anómalo”, mientras que un método que haga reconocimiento de patrones podría saber en detalle lo que está ocurriendo, siempre que esté entre los patrones almacenados.

Se puede decir que el modelo de IDS basado en ABS que se propone en este trabajo, funciona como un AIS (*Artificial Immune System*) [14]. Un AIS es un sistema adaptativo que simula el funcionamiento de un sistema inmune con el objetivo de resolver un problema particular [9]. El modelo ABS propuesto en este trabajo, como se verá más adelante, toma elementos y mecanismos del sistema inmune y los integra para obtener un sistema clasificador que funciona como un NIDS (*Network Intrusion Detection System*).

Para hacer un símil con el sistema inmune biológico [43], se puede decir que los agentes representan las diversas partes del sistema inmune y un ataque o situación anómala representa un antígeno o elemento malicioso que ha entrado al sistema.

Al igual que en un sistema biológico, lo esperable es que la primera incursión

de este elemento foráneo provoque un caos en el sistema, esto se puede ver reflejado en una disminución de la población de agentes, acompañado de una baja de energía de ellos.

Este caos provoca una respuesta innata del sistema, una segunda población se encarga de adaptarse al ataque, y una vez que esta segunda población es capaz de reconocer el ataque, se convierten en células de memoria permanente, que, al igual que el sistema inmune biológico, pueden responder de manera inmediata a un ataque conocido mitigando el caos resultante en el sistema y además otorgando la posibilidad de caracterizar individualmente al ataque y quizá incluso de mediar algún tipo de respuesta del sistema.

Este proceso puede continuar ocurriendo para todos los nuevos ataques que se vayan presentando en la red, creando múltiples poblaciones que reconocen, cada una, un ataque distinto.

El principal aporte de esta investigación está centrado en las capacidades de caracterización de anomalías antes mencionadas. Los modelos desarrollados con anterioridad sólo poseían la capacidad de caracterizar la normalidad y tenían la desventaja de que eventualmente se adaptaban a los ataques, es decir, que si el ataque era muy regular o de larga duración, la población terminaba aceptándolo como una situación normal.

En este caso, se pretende aprovechar la capacidad adaptativa natural de la población de agentes para generar estas poblaciones de reconocimiento de la anormalidad.

Como ya se mencionó antes, estas nuevas poblaciones tendrían la capacidad innata de reconocer los ataques y podrían provocar una respuesta secundaria extremadamente rápida en comparación con la respuesta primaria ante la primera instancia del ataque. Esta respuesta secundaria que se menciona, se debe a las capacidades de aprendizaje y caracterización de amenazas que poseería esta propuesta.

Capítulo 2

Estado del Arte

Desde hace varios años que se ha experimentado con la idea de usar análogos al sistema inmune para construir un IDS que sea capaz de detectar ataques en tiempo real [15, 20, 24], ya en el año 1999 Jungwon y Bentley [25] propusieron una estructura general para la construcción de un NIDS con 3 requerimientos: que sea un sistema distribuido, que se organice solo y que sea liviano en cuanto a recursos del sistema.

Investigaciones anteriores [6, 35, 28] ya han demostrado la factibilidad de utilizar ABS para crear un IDS. Se utilizó un modelo de ABS bidimensional que contenía agentes sensibles a cambios en el medio, estos agentes podían ser observados para determinar si había ocurrido alguna anomalía, pero esto presentaba algunos problemas, siendo la complejización del modelo uno de los principales. Los modelos presentados contenían demasiadas variables cuya modificación tenía un gran efecto en los resultados.

En el 2014, Yang et al. [44] elaboró, en detalle, acerca de los pasos necesarios para crear un IDS basado en AIS.

Como primer paso indica que se deben representar los elementos del sistema y sus interacciones en una forma que concuerde con el sistema inmune, de esta forma se pueden medir y cuantificar las ya mencionadas interacciones usando medidas de afinidad[1], con diferentes representaciones o implementaciones adoptando diferentes formas de medir la afinidad.

El segundo paso corresponde a la generación de la población de agentes, también llamado el algoritmo generacional.

Finalmente el tercer paso es descrito como la optimización del algoritmo, en

[1] Se entiende por afinidad, una medida de la adaptación de un agente a una cierta característica del entorno.

otras palabras, la evolución de la población mediante variados mecanismos.

De esta descripción, se desprenden 3 principales problemas que deben ser atacados:

2.1. Codificación del antígeno y anticuerpo

La codificación del antígeno y del anticuerpo es una decisión que es clave para una correcta representación de la situación y un correcto funcionamiento del sistema [14]. De esta manera, Yang [44] recomienda representar el código genético como un vector de características buscadas y cada antígeno como un dato que corresponde a algún segmento del código genético. La representación recomendada por Forrest et al. [15] es binaria, representación adoptada cuando desarrollaron el primer modelo de AIS, también adoptaron el uso de algoritmos de selección negativa (NSA) para evitar la auto-inmunidad, método también utilizado por otras investigaciones [40, 21, 4].

En cuanto a las medidas de afinidad de los agentes, definida en este caso como una medida del nivel de adaptación genética que tiene un agente a su medio ambiente, se han investigado diversas opciones, *r-contiguous bits matching* [15], *r-chunks matching* [5], *landscape-affinity matching* [17], distancia Hamming [20], entre otros [21, 29, 14]. Estos acercamientos presentan un problema en común, una baja cobertura del espacio solución [16] junto con un aumento exponencial del tiempo de cómputo requerido a medida que aumenta el tamaño de la población de agentes.

2.2. Algoritmo generacional

La mayoría de los métodos basados en NSA utilizan generación aleatoria de agentes detectores, de esta manera, por métodos de selección negativa se van eliminando aquellos que detectan el *self* quedando los que detectan el *non-self*, el llamado *self* corresponde al espacio o medio ambiente que es considerado normal o propio, mientras que el *non-self* corresponde a lo que es anormal o que no es propio del ambiente. Aunque este es el método más

utilizado, Stibor et al. [40] menciona que aumenta las posibilidades de generar detectores inválidos y que mientras más grande sea el *self*, más tardan los agentes en estar listos para poder clasificar.

Es por esto, que D'haeseleer et al. [13] idea 2 métodos para generar detectores, llamados:

- Linear Time Detector Generating Algorithm
- Greedy Detector Generating Algorithm

Ambos algoritmos buscan sólo generar detectores que sean válidos para las anomalías que se quieren clasificar, ya sea evitando que se realice una búsqueda en espacios que no contienen soluciones o eliminando detectores redundantes.

En cuanto a la relación con el sistema inmune humano, en 1959 Burnet [7] propuso mejorar la generación de agentes detectores mediante principios de selección clonal, esto fue posteriormente refinado y mejorado por de Castro y Von Zuben primero en el llamado CSA [10] y luego en su versión más conocida, CLONALG [11].

La habilidad de poder distinguir entre el *self* y el *non-self* es uno de los principios que guían el desarrollo de los AIS, por lo tanto es muy útil aprovecharse de la selección negativa. Respecto a esto, Matzinger propuso utilizar la teoría del peligro que dice que la respuesta inmune está mediada por señales de daños o de peligro, más que por la presencia de antígenos o partículas extrañas o en combinación con estos últimos [7, 30].

2.3. Evolución y Algoritmo Evolutivo

Para evitar el estancamiento genético de los agentes detectores, es decir, que alcancen un óptimo local, Hofmeyr sugirió hacer que estos fueran dinámicos [16], para esto les asigna una edad y una vez que ellos cumplen su ciclo de vida, pueden ser descartados y reemplazados. Como alternativas propuestas, se ha explorado la posibilidad de eliminar aquellos que no son muy efectivos para lo que están evaluando [27].

Otra posibilidad con la que se ha experimentado es usar algoritmos genéticos para estudiar el impacto de la evolución en los genes de los anticuerpos [19, 34, 32], esto permite utilizar algún tipo de retroalimentación de los anticuerpos previamente presentes para, en el futuro, generar mejores de éstos.

2.4. Funcionamiento Online

Entendemos como funcionamiento online a la capacidad del sistema de trabajar en tiempo real, procesando el tráfico entrante a medida que va pasando por la red y sin demora alguna. El funcionamiento online puede o no incluir el aprendizaje online.

Se han hecho algunas propuestas en términos de funcionamiento online de un IDS, principalmente con la introducción de 3 parámetros importantes para los detectores [26]:

- Período de tolerancia para un agente detector inmaduro
- Umbral de activación del agente
- Edad máxima del agente reactivo

La principal característica que debe mostrar un sistema que funciona online, es la capacidad de procesar los datos en tiempo real, esto para que no haya un retraso de los datos que está analizando respecto de los datos que se presentan en la red, esta característica puede ser observada en software tales como SNORT Y Suricata, ambos conocidos sistemas IDS [41].

Posteriormente, Yang et.al [45] presentaron un modelo que era capaz, mediante agentes distribuidos, de capturar tráfico en tiempo real. Su experimentación mostró que el modelo tiene todas las características necesarias para funcionar en línea.

2.5. Posibles variantes

Una posible variación de modelo de IDS que explica Snapp et al. [38], consiste en un sistema denominado *Distributed Intrusion Detection System* (DIDS), este modelo consta de una arquitectura distribuida que combina varias capas, la primera capa es el director del DIDS que se encarga principalmente del análisis de los datos obtenidos por las otras capas, la segunda capa es un monitor de LAN, de los cuales existe uno por cada subred que se quiera monitorear y el tercer elemento es un monitor de host, estos son generados individualmente para cada host.

Ambos monitores se encargan de la recolección de información para hacerla llegar al director. Este sistema demostró tener la capacidad para identificar individualmente a máquinas de una red y de poder seguir a un atacante que se mueva por la red, pero tiene la desventaja de que podría no ser escalable en redes más grandes debido a la cantidad de información que tendría que procesar el director.

Una opción descrita por Jabez y Muthukumar [23] es la de usar detección de anomalías enfocada en los llamados *outliers*, es decir, datos que se alejan substancialmente de la normalidad, ellos proponen una solución que buscan cumpla con 5 propiedades: precisión, que pueda lidiar con pequeñas variaciones en los datos, baja tasa de falsos positivos, adaptable y que funcione en tiempo real.

Ellos proponen utilizar datos acerca de una normalidad conocida para calcular la distancia que un dato presenta respecto a lo normal, de esta manera, se le asigna un puntaje de anormalidad a cada dato y si la cantidad de datos considerados anormales sobrepasan cierto nivel, se considera que se dispara una alerta.

En cuanto a resultados, se comparan favorablemente con las técnicas más comunes de *machine learning*, obteniendo mejor desempeño en cuanto a anomalías detectadas y porcentaje de uso de CPU.

Algunas opciones para sistemas IDS distribuidos han sido exploradas con

anterioridad [21]. Igbe et al. utilizaron un sistema distribuido con agentes separados por la red, cada uno de los cuales utilizaba algoritmos de selección negativa para mejorar la calidad de los detectores y separarlos en *self* y *non-self*. Estos agentes posteriormente comparan sus resultados para poder determinar si la detección tiene un cierto nivel de certeza y así poder disparar una alarma.

2.6. Dataset

Algunos de los datasets más utilizados en el ámbito de investigación de ataques a redes son DARPA'98/99, KDD'99 y NSL-KDD.

Ha sido ampliamente [37, 12, 22, 1, 33, 36, 46] recomendado el no utilizar el dataset DARPA, debido, principalmente, a que fue recopilado en una época en que las comunicaciones en redes eran distintas a lo que son ahora, es decir, ya no representa un conjunto de datos suficientemente actualizado.

Por otro lado, el dataset KDD'99 se desprende de DARPA, siendo un subconjunto de este último, por lo tanto no debería considerarse válido por las razones ya expuestas.

La tercera opción, NSL-KDD, es la mejor considerada de las 3, el NSL-KDD fue tomado del KDD'99, pero se proceso de manera de tratar de eliminar los problemas detectados en el set KDD'99 original [42], entre estos problemas, se encuentra el hecho de que el set original está sesgado a ciertos tipos de ataque que son más frecuentes en los datos, por lo tanto, esto usualmente resulta en un sobre-entrenamiento para esos tipos de datos mientras que los ataques menos comunes (R2L y U2R) no son detectados por los modelos. Por lo tanto, este set está balanceado respecto a los diversos tipos de ataque para tratar de evitar este problema, aún así, los creadores del dataset reconocen que no esperan que esto resuelva todos los problemas inherentes a KDD y simplemente lo recomiendan debido a la falta de datasets públicos de calidad.

2.7. Medidas de Desempeño

Según explica Sokolova et al. [39], las medidas clásicas de desempeño para *Machine Learning* tienen graves problemas. Entre éstos se destacan el hecho de que la medida más usada, *accuracy*, no distingue entre distintas clases. El uso de medidas como *precision* y *recall* que sólo se enfocan en la clase que se busca clasificar, debido a que no toman en cuenta los verdaderos negativos. Mientras que otra medida común, el *F-score* representa un balance entre *precision* y *recall* [14].

Por otro lado, la curva ROC presenta una evaluación mucho más completa acerca del desempeño de un clasificador al igual que el área bajo la curva *AUC*.

Tomando en cuenta lo dicho anteriormente, los autores definen las medidas más comunes de la siguiente forma:

- *Accuracy*: Indica la probabilidad de realizar una evaluación correcta, en otras palabras, la efectividad general del algoritmo.
- *Precision*: Estima el valor predictivo del algoritmo relativo a una cierta etiqueta, ya sea positiva o negativa.
- *Sensitivity (Specificity)*: Aproxima la probabilidad de que una etiqueta positiva (o negativa) haya sido aplicada correctamente, o la efectividad del algoritmo para una cierta clase.
- ROC: Muestra la relación entre *Sensitivity* y *Specificity* del algoritmo.
- *F-Score*: Una medida compuesta que beneficia a los algoritmos con una buena sensitividad, respecto a aquellos con una alta especificidad.

Para resolver los problemas planteados, se proponen las siguientes medidas, tomadas de métricas que se utilizan en diagnósticos médicos para analizar exámenes:

1. *Youden's Index*: Este índice califica la habilidad de un algoritmo para evitar

fallar, ponderando, con igual importancia, los positivos y negativos:

$$\gamma = sensitivity - (1 - specificity) \tag{2.1}$$

También se puede calcular el índice, relativo al área bajo la curva (*AUC*) ROC:

$$\gamma = 2AUC_b - 1 \tag{2.2}$$

En ambos casos, un mayor valor de γ representa una mejor habilidad para evitar fallar en la clasificación.

2. *Likelihood*: Esto combina las medidas de sensitividad y especificidad en 2 nuevas medidas definidas de la siguiente forma:

$$\rho+ = \frac{sensitivity}{1 - specificity} \tag{2.3}$$

$$\rho- = \frac{1 - sensitivity}{specificity} \tag{2.4}$$

Un $\rho+$ más alto y un $\rho-$ más bajo indican un mejor desempeño en las clases positivas y negativas respectivamente, de esta manera se puede priorizar una medida por sobre la otra al comparar distintos algoritmos.

Las posibles combinaciones de ρ_+ y ρ_- se pueden ver con más detalle en las siguientes cuatro ecuaciones:

$$\rho_+^A > \rho_+^B \wedge \rho_-^A < \rho_-^B \tag{2.5}$$

En este caso A es superior a B para todos los casos.

$$\rho_+^A < \rho_+^B \wedge \rho_-^A < \rho_-^B \tag{2.6}$$

En este caso A es superior para los verdaderos y falsos negativos.

$$\rho_+^A > \rho_+^B \wedge \rho_-^A > \rho_-^B \tag{2.7}$$

En este caso A es superior para los verdaderos y falsos positivos.

$$\rho_+^A < \rho_+^B \wedge \rho_-^A > \rho_-^B \tag{2.8}$$

En este caso A es inferior a B para todos los casos.

Los problemas que se verían beneficiados por el uso de alguna de estas métricas son aquellos que cumplan con las siguientes cualidades:

- Las clases tienen el mismo peso o valor
- Se deben comparar 2 o más clasificadores
- La obtención de datos es costosa (trabajo manual o poca disponibilidad)

2.8. Teoría del Peligro

La teoría del peligro fue propuesta por primera vez por Matzinger [31]. Su punto central dice que el sistema inmune no distingue entre el *self* y el *non-self*, si no que distingue entre lo que es seguro y lo que es dañino. Hace esta diferenciación reconociendo patógenos o señales de alarma de tejidos o células dañadas.

Aicklein et al. [3] realizaron un análisis de la teoría del peligro desde la perspectiva de un sistema inmune artificial. En esa investigación, ellos concluyen que el punto clave es la elección de la señal de peligro, llegando a mencionar que es una decisión tan crítica para el funcionamiento de un sistema evolutivo como lo es la elección del algoritmo de *fitness*.

En otra investigación, centrada en un posible uso de la teoría del peligro para sistemas inmunes artificiales [2], los autores enfatizan la elección de las señales de peligro, en este caso, la muerte de agentes, realizando una marcada diferenciación entre las señales apoptóticas y necróticas. Las señales apoptóticas

son aquellas correspondientes a la muerte "*normal*"de los agentes, esto puede ocurrir habitualmente por diversos motivos y no debe ser necesariamente una causa de alarma, sin embargo, una señal apoptótica puede ser precursora de una verdadera anomalía. Por otro lado, las señales necróticas son las que son causadas por daños en el sistema que se desea monitorear, estas son evidentes señales de peligro que son evidencia más clara de que hay una situación anómala, un ataque.

2.9. El ABS

El modelo ABS (*Artificial Bioindicator System*) de Blum et al. [6] representa la base de esta investigación, por lo tanto, se le ha de prestar una especial atención a su funcionamiento.

El modelo utiliza una población de agentes sensibles al ambiente, agentes que son usados para la detección de anomalías. Por esta razón, estos agentes son llamados agentes AB (*Artificial Bioindicators*)

El modelo es representado como un mundo bidimensional poblado por dos entidades, *agentes* y *partículas*. Los agentes actúan como bioindicadores. Cambios en el ambiente perturban a estos agentes. El ambiente corresponde a la red, o más específicamente, al tráfico que presenta esta red. Por lo tanto, las partículas son una representación de características del tráfico en este mundo bidimensional.

Los agentes son entidades inmóviles, creados con un código genético aleatorio. Estos agentes pasan por procesos evolutivos convencionales que les permiten adaptarse de tal forma que solo los más aptos logren sobrevivir a las condiciones. Cada agente tiene dos parámetros importantes:

- **Energía (E_x):** Indica el nivel energético del agente, esto se utiliza para su metabolismo y para su reproducción.
- **Código Genético (θ):** Este es un vector que determina las capacidades del agente, en otras palabras, esto representa la capacidad que tiene el agente para sobrevivir a distintas características del ambiente, cada $i \in$

θ representa un gen, y cada gen representa una de las características observadas en el tráfico.

Los agentes requieren de un cierto nivel energético para reproducirse (E_r). Los agentes se reproducen de manera asexuada, con un operador de mutación que actúa sobre sus genes, esto consiste simplemente en el intercambio aleatorio de dos elementos del vector θ.

Las partículas son creadas y entran al mundo cuando el modelo detecta una característica específica en el tráfico de la red. Estas partículas son insertadas en el mundo y se mueven a través de él, cruzándose con los agentes, hasta que salen del mundo o se quedan sin energía. Las partículas tienen dos componentes importantes:

- **Tipo:** Esto determina la característica que originó a la partícula, existen tantos tipos como características se busquen en el tráfico.
- **Energía:** Esto indica cuánta energía le resta a la partícula que incide sobre la cantidad de agentes que pueden obtener energía de ella al colisionar, en otras palabras, es la energía que aún tiene una partícula para entregar a los agentes del medio.

Cuando una partícula p impacta con un agente x, puede tener un efecto positivo o negativo sobre el nivel de energía de x, dependiendo de la siguiente ecuación:

$$N_x = \sum_p (\Phi - \epsilon * i_p) \tag{2.9}$$

Donde N_x es la energía que entrega la partícula p para al agente x, Φ es la máxima energía que una partícula puede otorgarle a un agente, i_p es el índice en el que ese tipo de partícula se encuentra en el código genético de x, y ϵ es una pérdida lineal de valor nutricional. Por lo tanto, según esta ecuación, una partícula puede afectar a un agente ya sea otorgándole energía o quitándole energía, dependiendo de la afinidad que tenga el agente por esta partícula.

Para ilustrar esto, considérese un sistema con cinco partículas distintas. Cada agente tiene una permutación de todas las partículas disponibles en su vector genético. De esta forma, si un agente A tiene el vector $A_v = (e, a, c, b, d)$, y el paquete de red entrante P produce la creación de las partículas e y b, el valor nutricional asociado a P para el agente A será: $N_A = (\Phi - \epsilon * i_e) + (\Phi - \epsilon * i_b)$, donde i_e es 0 y i_b es 3. En este escenario, la partícula e entrega la mayor cantidad de energía posible a A, la partícula b entrega un menor benéfico y podría ser tóxica (con un valor negativo de N) dependiendo del valor de ϵ.

2.10. Principales problemas encontrados

Tomando en cuenta todo lo expuesto anteriormente, podemos identificar los siguientes problemas principales que han presentado las propuestas existentes en el ámbito de los Sistemas Inmunes Artificiales, en particular Blum et al. [6], que representa la idea de base de esta investigación:

1. **Falta de capacidad reactiva:** El modelo ABS original carece de cualquier capacidad para reaccionar ante amenazas. Para notificar la detección, el modelo requiere de un observador externo que realice alguna acción para bloquear la amenaza. El modelo simplemente reporta la situación actual de la red.

2. **Problema de readaptación de los agentes:** Cuando un ataque persiste en el tiempo, los agentes tienden a adaptarse a él, de esta forma pierden sensibilidad a tal ataque, y finalmente se termina clasificando como normalidad. Esto contamina el *pool* genético de los agentes de la normalidad.

3. **Tiempo de respuesta:** El modelo requiere de una perturbación significativa y sostenida de los parámetros observados en la población, por lo tanto, la reacción del modelo puede ser relativamente lenta.

4. **Carencia de capacidad de aprender ataques:** El modelo no cuenta con

ningún mecanismo para aprender a caracterizar distintos tipos de ataques, cada vez que un ataque es encontrado, la reacción es la misma, y tarda lo mismo.

Capítulo 3

Modelo RABS

La solución propuesta con este modelo, llamado RABS (*Reactive Artificial Bioindicators System*), surge como una solución a los problemas ya mencionados. Por lo tanto, el nuevo modelo ha de tener los siguientes requerimientos:

- Las reacciones deber ser mediadas por el modelo mismo, no por herramientas u observadores externos, manteniendo una funcionalidad otorgada por el comportamiento emergente.
- Proteger la *pureza* de la normalidad, implementando un mecanismo que inhibe la adaptación de la normalidad a situaciones anómalas.
- Caracterizar, aprender y memorizar ataques que ya han sido vistos en la red, de esta forma, es posible aprovechar una respuesta secundaria la siguiente vez que el mismo ataque se presente en la red, esta respuesta secundaria sería prácticamente instantánea, atacando el problema de la lentitud y la falta de aprendizaje del modelo original.

En la Figura 3.1 se puede ver el funcionamiento general del modelo RABS. El tráfico de la red es capturado por medio de un sniffer para luego ser insertado en el sistema compuesto por distintas poblaciones de agentes.

El modelo RABS tiene dos tipos de agentes, el primer tipo de agente es similar a la definición original encontrada en el modelo ABS, y se denominan *n-agents*. El propósito de estos *n-agents* es el de caracterizar la normalidad del sistema que está siendo monitoreando (también llamado *Self*). El segundo tipo de agente corresponde a los *r-agents*; cuyo objetivo es caracterizar anomalías en el sistema.

El modelo RABS puede crear una o más poblaciones de *r-agents*, existiendo una población por cada amenaza distinta encontrada. Las dos poblaciones se

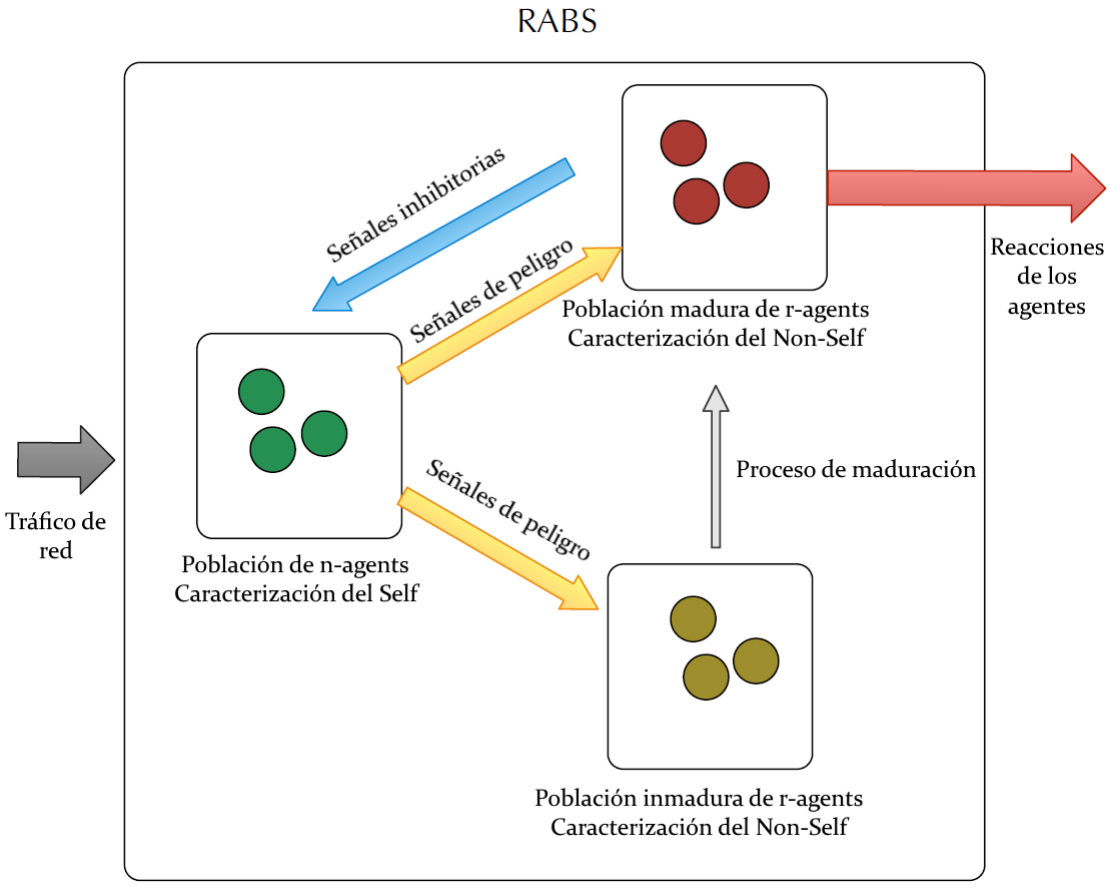

Figura 3.1: Modelo RABS

comunican mediante señales, una perturbación de los *n-agents* causa que se emita una señal que origina la activación de una población de *r-agents*, de una forma muy similar al funcionamiento de la inmunidad natural de un organismo.

Cuando los *r-agents* ya han madurado y caracterizado la amenaza, pueden producir señales que indican que está ocurriendo el mismo evento. Estas señales inhiben la evolución de los *n-agents*, evitando la contaminación de la caracterización de la normalidad. Las relaciones entre ambos modelos pueden ser descritas con las siguientes reglas básicas:

1. ▼(n-agents)∧▲(r-agents)$\overset{produce}{\rightarrow}$ {Reacción completa}
2. ▼(n-agents)∧▼(r-agents)$\overset{produce}{\rightarrow}$ {Nueva población, Media reacción}
3. ▲(n-agents)∧▲(r-agents)$\overset{produce}{\rightarrow}$ {Reacción completa, i-NSA}
4. ▲(n-agents)∧▼(r-agents)$\overset{produce}{\rightarrow}$ $\emptyset$

Este conjunto de reglas demuestran el funcionamiento del RABS. Las reglas no son lanzadas por un observador externo, si no que por la acción directa de los agentes.

En la regla (1), hay una situación con el RABS completamente operacional y bajo un ataque. En esta situación, los *n-agents* pierden afinidad con el tráfico entrante, reflejado en señales de peligro del modelo. Además ocurre que una de las poblaciones de *r-agents* presenta una alta afinidad con el tráfico actual, produciendo la reacción de los *r-agents* por una anomalía conocida y aprendida.

La regla (2) representa una situación con una nueva amenaza. En este caso, la afinidad de los *n-agents* frente al tráfico entrante disminuye pero ninguna de las poblaciones de *r-agents* genera una respuesta inmediata de alta afinidad, por lo tanto, el sistema lanza una nueva población de *r-agents* para que caractericen la nueva amenaza. Eso produce una media reacción que es general en vez de una reacción completa que sería específica.

La regla (3) muestra una posible inconsistencia entre las poblaciones. Un aumento en la afinidad de los *r-agents* indicaría una situación de ataque conocido, pero la población de *n-agents* no se ve perturbada por esta situación y no

presenta cambios significativos en su afinidad. En esta situación, RABS aplica una variación de un algoritmo de selección negativa (i-NSA) para resolver esta inconsistencia y poder lograr una reacción completa y correcta del modelo.

Finalmente, la regla (4) muestra el estado de funcionamiento de RABS sin presencia de ataque, en este estado, la afinidad de los *n-agents* es alta y ninguna población de *r-agents* está reaccionando con alta afinidad al tráfico entrante.

3.1. Algoritmo de selección negativa inversa (I-NSA)

El principal objetivo de un algoritmo de selección negativa es permitir la creación segura de nuevos agentes detectores, destruyendo detectores que clasifican lo incorrecto antes que maduren. El algoritmo se basa en la rigidez o inmutabilidad del Self. Para aplicar esto en RABS, se considera una premisa un poco distinta.

El comportamiento normal de los sistemas es continuamente cambiante, mientras que las amenazas son estáticas. De esta forma, la exposición a un nuevo ataque genera la creación de una nueva población de *r-agents* para caracterizar esta situación particular. En otras palabras, se está caracterizando lo que es inmutable. Considerando esto, RABS utiliza los *r-agents* creados para eliminar coincidencias que puedan existir en los *n-agents*.

Dicho de otra forma, utilizando a los *r-agents* como base, se eliminan todos los *n-agents* que tengan un parecido a ellos, así se evita que los *n-agents* caractericen lo que no les corresponde.

Por esto se habla de una selección negativa inversa (i-NSA), ya que se utiliza el non-Self para limpiar al Self, lo contrario de la idea original del algoritmo.

3.2. RABS como un NIPS

Técnicamente, RABS es un *Network-based Intrusion Prevention System* (NIPS), o un sistema diseñado para detectar y actuar contra amenazas en la red. En este sentido, utiliza, como entrada, datos del sistema y genera, como salida, una serie de medidas de mitigación de la amenaza. En un inicio,

RABS fue concebido como trabajando en conjunto con su sistema anfitrión, manteniendo un cierto nivel de "*salud*" en la red para sobrevivir y preservar la integridad de su anfitrión.

La entrada que utiliza RABS es tráfico de la red que está monitoreando. Se considera un conjunto de características binarias extraídas de las cabeceras de los paquetes de red de la capa de transporte. También son considerados los puertos más comúnmente atacados o que tienen servicios importantes, los detalles de las características observadas son como sigue:

- **Bits reservados de la capa de red:** RB, MF, DF, F1, F2.
- **Flags TCP de la capa de transporte:** URG, ACK, PSH, RST SYN, FIN.
- **Puertos más comúnmente atacados:** Telnet, SSH, FTP, Netbios, Rlogin, RPC, NFS, NNTP, Lockd, Netbios, Xwin, DNS, LDAP, SMTP, POP, NTP, IMAP, HTTP, SSL, px, Serv, Time, TFTP, Finger, lpd, Syslog, SNMP, bgp, Socks.

Finalmente, se crea un vector binario V de largo l_v para cada paquete observado, donde l_v es el número de características que se quieren observar en el tráfico, y cada posición del vector se asocia con una característica particular del tráfico. Este vector es el que determina las partículas que se crean a partir del paquete analizado.

La salida del RABS son una serie de acciones disparadas por las poblaciones (*n-agents* y *r-agents*). Estas acciones pueden incluir el cierre de sesiones de red, baneo temporal de direcciones IP, notificación a un administrador de la situación u otras que puede ser definidas según las necesidades específicas.

3.3. Simplificación del modelo

Se propone una simplificación del modelo ABS, con una topología unidimensional, un número fijo de agentes, lo que trae consigo una menor cantidad de parámetros y reglas de interacción más simples.

Se realizó una significativa simplificación del modelo ABS original, de una topología bidimensional a una unidimensional. La principal razón para este cambio es que no había una justificación para que el modelo ABS usara una topología bidimensional, no había uso de ninguna de las características propias de un mundo bidimensional, no se extraía información ni de la posición de los agentes ni sus distancias o movimientos.

El tener un número fijo de agentes permite controlar más fácilmente el tamaño de la población. Buscamos reducir la población a un tamaño tal que no impida su funcionamiento como especie indicadora pero que signifique la menor cantidad de poder de cómputo que se pueda necesitar.

Una menor cantidad de parámetros y reglas más simples ayudan a que el modelo sea más fácilmente manejable, tener una gran cantidad de parámetros hace innecesariamente complejo al modelo, muchas veces los parámetros son interdependientes por lo que modificar uno de ellos causa cambios en otras partes del sistema, cambios que deben ser controlados modificando otros parámetros.

Por lo tanto, se busca reducir la cantidad de parámetros al mínimo para simplificar cualquier proceso de parametrización o tuning que se desee hacer. De la misma forma, reglas de interacción más simples hacen que el modelo completo sea más simple y que requiera menor poder de cómputo.

Específicamente, el modelo ABS contenía un total de 11 parámetros interdependientes, mientras que RABS solo tiene 6 parámetros que podrían requerir ajustes. Esto se traduce en una enorme reducción del espacio de búsqueda del conjunto de parámetros óptimo.

3.4. Los agentes del modelo RABS

El RABS está basado en una población de agentes que evolucionan, estos agentes se adaptan a las condiciones de la red, permitiendo de esta forma, la caracterización y detección de distintos escenarios. Para esto, se definieron los dos tipos de agentes principales: *n-agents* y *r-agents*. Ambos tienen los mismos parámetros y configuración genética y un proceso de maduración similar que

se describe a continuación:

3.5. La evolución de la población

El RABS mantiene intacta la idea de usar a los agentes como bioindicadores. Por lo tanto, la presión de selección la provee el ambiente y la población de agentes se adapta a estas condiciones cambiantes. En este contexto, los individuos mejor adaptados son aquellos con los niveles de energía más altos. Como todos estos agentes tiene capacidades evolutivas, ellos tienen un vector genético. Este vector G permite establecer el nivel de afinidad del agente con el ambiente. La representación genética de G es la misma que usa el vector binario V para el tráfico entrante.

Los mecanismos evolutivos de RABS están explicados en los algoritmos 1 y 2. El primer pseudocódigo muestra el comportamiento de la población de *n-agents*, el segundo pseudocódigo muestra a los *r-agents*. Ambos procedimientos son muy similares entre ellos y además son interdependientes. La población de *n-agents* activa la evolución de los *r-agents* y la población de *r-agents* puede inhibir la adaptación de los *n-agents*. Ambos algoritmos se ejecutan por cada paquete que entra al sistema.

Entrando más en detalle en el funcionamiento de estos, el algoritmo 1 muestra que la población de *n-agents* A es expuesta a cada uno de los paquetes de red p_{cap}, luego en la línea 4, se calcula la cantidad de agentes con un nivel de energía inferior a $energy_{thres}$ y se almacena en DS (*Danger Signals*). Esta variable es continuamente monitoreada por los *r-agents*, ya que es esta la que dispara su ejecución inicial.

En la línea 5, el algoritmo revisa si las señales de alerta por ataque AS están bajo un cierto nivel AS_{thres}. Cuando esta condición se cumple, el modelo funciona con normalidad; en el caso contrario, el algoritmo inhibe el proceso de evolución de los *n-agents* para preservar la normalidad lo más intacta posible.

Cuando el modelo satisface su condición de funcionamiento normal, el algoritmo selecciona un grupo de agentes padres A' con un nivel de energía superior a $energy_{thres}$, estos agentes se consideran adaptados al tráfico debido

a su alto nivel energético.

De la misma forma, el modelo selecciona un segundo grupo de agentes $A*$ de A usando un nivel Mem_{thres} que siempre cumple la condición de que $Mem_{thres} >> energy_{thres}$. Los agentes en este conjunto $A*$ son considerados como maduros por su alta afinidad con el tráfico entrante. El algoritmo define r como la cantidad de agentes que van a ser creados en esa iteración y almacenados en A_n; el valor de r es determinado usando una proporción R de la población total de *n-agents* y considerando que además hay que reemplazar los agentes muertos.

En las líneas 11 y 12, dos padres p_1 y p_2 son seleccionados de A' usando un FPS (*Fitness Proportionate Selection*) y se combinan en o_1 y o_2 con un cruzamiento uniforme CX_{uni}, luego, los nuevos hijos son agregados a A_n. Este proceso se repite hasta obtener una cantidad de agentes igual o superior a r.

En la línea 15 se presenta el mecanismo de reemplazo para los *n-agents*, este consiste simplemente en ir reemplazando los agentes con menor energía con los nuevos agentes de A_n.

El comportamiento del sistema puede variar en el tiempo, por lo tanto RABS cuenta con un mecanismo para actualizar la representación del Self. Esto se logra mediante la función $update$ en la línea 16 que reemplaza a los *n-agents* maduros con un nuevo conjunto de agentes de alta energía.

El mecanismo evolutivo de los *r-agents* demostrado en el algoritmo 2 es muy similar al ya explicado. La diferencia esencial es que la evolución y evaluación de los *r-agents* solo pueden ocurrir en la presencia de señales de peligro DS. En este sentido, el algoritmo toma como entrada el valor de DS determinado por la población de *n-agents*. Cuando este valor alcanza un cierto nivel DS_{thres}, los *r-agents* son expuestos al tráfico de la red p_{cap} y evolucionan con los mismos mecanismos que los *n-agents*. Otra diferencia significativa es que los *r-agents* no tiene un mecanismo de actualización, por lo que luego de caracterizar un escenario en particular, dejan de evolucionar y solo se utilizan para clasificar. Los *r-agents* son responsables de calcular y proveer las señales de alerta AS usadas por los *n-agents* para inhibir su proceso evolutivo.

Algorithm 1 Mecanismo evolutivo de los *n-agents* en RABS

input: $A, energy_{thres}, R, p_{cap}, AS$
input: AS_{thres}, Mem_{thres}
$expose(A, p_{cap})$
$DS = |\{a \in A^{**} | a_{energy} < energy_{thres}\}|$
if $AS < AS_{thres}$ **then**
  $A' := \{a \in A | a_{energy} \geq energy_{thres}\}$
  $A^* := \{a \in A | a_{energy} \geq Mem_{thres}\}$
  $A_n := \emptyset$
  $r := argMax(deadAgents, R \cdot |A|)$
  while $|A_n| < r$ **do**
    $p_1, p_2 := FPS(A')$
    $o_1, o_2 := CX_{uni}(p_1, p_2)$
    $A_n := A_n \cup \{o_1, o_2\}$
  end while
  $replace(A, An, policy(lower\_energy))$
  $update(A^{**}, A^*, policy(lower\_energy))$
end if

Algorithm 2 Mecanismo evolutivo de los *r-agents* en RABS

input: $A, energy_{thres}, R, p_{cap}, DS$
input: $DS_{thres}, Mem_{thres}, AS_{thres}$
if $DS > DS_{thres}$ **then**
  $expose(A, p_{cap})$
  $AS = |\{a \in A^* | a_{energy} \geq AS_{thres}\}|$
  $A' := \{a \in A | a_{energy} \geq energy_{thres}\}$
  $A^* := A^* \cup \{a \in A | a_{energy} \geq Mem_{thres}\}$
  $A_n := \emptyset$
  $r := argMax(deadAgents, R \cdot |A|)$
  while $|A_n| < r$ **do**
    $p_1, p_2 := FPS(A')$
    $o_1, o_2 := CX_{uni}(p_1, p_2)$
    $A_n := A_n \cup \{o_1, o_2\}$
  end while
  $replace(A, An, policy(lower\_energy))$
end if

Capítulo 4

Evaluación Experimental

El principal objetivo de la evaluación experimental es comparar el desempeño del modelo RABS con la base de ABS, en particular, se desea probar las ventajas que otorgan las nuevas características de RABS por sobre ABS. La configuración de los experimentos fue diseñada para destacar las capacidades que tiene RABS para aprender de ataques y, de esta forma, mejorar el tiempo de respuesta en ataques conocidos. Para cumplir con estos requerimientos, los experimentos tienen la siguiente estructura:

- Fase 1: El modelo es expuesto a tráfico de red normal. El objetivo es lograr una adaptación inicial a las condiciones normales de funcionamiento de la red.
- Fase 2: El modelo es expuesto a un ataque para evaluar su respuesta.
- Fase 3: El tráfico de red cambia de nuevo a una condición de normalidad.
- Fase 4: El modelo es expuesto de nuevo al mismo ataque presente en la Fase 2 para probar la respuesta secundaria del modelo.

El hardware y software utilizados para los experimentos son los siguientes:

- Sistema Operativo: Windows 10 Home 64-bit
- CPU: Intel Core i7 6600U @ 2.50GHz
- RAM: 16.0GB Dual-Channel DDR3 @ 790MHz
- Experimentación: NetLogo 6.1.0

Para lidiar, en la medida de lo posible, con la naturaleza no-determinista del comportamiento de los algoritmos evaluados, todos los experimentos fueron ejecutados diez veces y posteriormente promediados. El dataset utilizado se compone de elementos de DARPA'98 con tráfico de normalidad y de ataque complementario capturado para esta investigación. La razón principal para usar DARPA'98, pese a que tiene problemas conocidos es que sigue siendo ampliamente utilizado hoy en día y tener un dataset en común permite una fácil comparación con modelos existentes. Los detalles del dataset se encuentran en la siguiente tabla:

Tipo de ataque	**Nombre**	**Paquetes**
Denial of Service (DoS)	land	1100
	syslog	1002
	udp-storm	1667
	custom*	3000
Total DoS		6769
Remote to Local (R2L)	http-tunnel	1196
	sendmail	1548
Total R2L		2744
User to Root (U2R)	sql	1026
	ffbconfig	511
	ffbconfig	1022
	ps	703
Total U2R		3262
Tráfico normal	normal	8618
Total		21393

La mayor parte de las capturas fueron tomadas de DARPA porque es un dataset público, bien conocido y bien rotulado, utilizado comúnmente para medir el desempeño de variados modelos IPS/IDP. El dataset ha sido aumentado

con un ataque generado por la herramienta *GoldenEye* [1]. Las pruebas no pretenden validar el modelo RABS como una propuesta de IPS competitiva. Los experimentos están diseñados para mostrar y enfatizar las características del RABS, en particular, su capacidad para caracterizar ataques desconocidos y posteriormente responder rápidamente a ellos.

Los paquetes de cada captura fueron segmentados para simplificar la evaluación, el tráfico fue dividido en secciones de largo 100 paquetes, al final de cada segmento se reportan las métricas y se evalúa el desempeño. Si no han habido detecciones en ese segmento, entonces se considera como un negativo, si hubo al menos una detección, entonces es considerado como un positivo.

La detección puede ser disparada por uno de tres mecanismos:

1. Señales de peligro de los *n-agents* sin respuesta de los *r-agents*.
2. Señales de peligro de los *n-agents* junto con una alerta de los *r-agents*.
3. Una alerta de los *r-agents* sin señales de peligro de los *n-agents*.

Los experimentos en el modelo RABS fueron ejecutados con la ya mencionada estructura de 4 fases.

Los experimentos para el modelo ABS fueron ejecutados en una estructura más simple de solo 3 fases. En estos experimentos se eliminó la segunda fase de ataque (anteriormente la fase 4) ya que el modelo bidimensional no tiene capacidad de respuesta secundaria, el desempeño esperado en esta segunda fase de ataque es igual o peor al mostrado en la primera, pruebas preliminares del modelo ABS mostraron esta tendencia.

Adicionalmente, se ejecutó una prueba de un escenario ideal para el modelo ABS, este escenario consiste de una fase normal seguida por una fase de ataque, pero en este caso la evaluación del modelo se detiene tan pronto como el modelo comienza a adaptarse al ataque, esto es consistente con los experimentos hechos en la publicación original del modelo [6]. Esto fue hecho para evitar que genere falsos negativos al terminar este proceso de adaptación indeseada al ataque, esta prueba representa el mejor caso posible para el ABS.

[1] https://sourceforge.net/projects/goldeneye/

El punto importante que se quiere remarcar con esta prueba es que el modelo ABS funciona bien como un detector de anomalías cuando el ataque es rápido y corto, pero su desempeño cae cuando es expuesto a un escenario más realista en el cual un ataque puede durar más tiempo.

La prueba del caso ideal para el ABS, por su carácter de prueba demostrativa, solo se ejecutó con una de las capturas de ataque, se eligió la captura de ataque de denegación de servicio de tipo *udp-storm*.

Resumiendo, la evaluación experimental se divide básicamente en dos tipos de experimentos.

El primero corresponde al escenario ideal para el ABS, ideado para destacar las capacidades que tiene como detector de anomalías.

El segundo, una prueba comparativa entre los modelos ABS y RABS, en un escenario realista que muestre la diferencia de desempeño de ambos modelos.

Capítulo 5

Resultados

Como fue mencionado anteriormente, los experimentos fueron realizados diez veces por captura, por lo tanto, los resultados representan un promedio de esas diez pruebas, esto para minimizar las variaciones inherentes a un modelo no-determinista.

5.1. Prueba de ABS en caso ideal

Los resultados para el modelo ABS para el experimento del caso ideal previamente detallado son como siguen:

		Valor Predicho		
		Ataque	**Normal**	**Total**
Valor Real	**Ataque**	4.2	0.9	5.1
	Normal	5	71	76
	Total	9.2	71.9	

La tabla anterior tiene una desviación estándar de 0.23. Con estos resultados, podemos calcular que, en un caso ideal, el modelo ABS tiene una sensitividad de 0.82 y una especificidad de 0.93.

5.2. Pruebas comparativas en escenario realista

Esta primera tabla muestra los resultados del modelo ABS en el escenario realista:

		Valor Predicho		
		Ataque	**Normal**	**Total**
Valor Real	**Ataque**	42.2	83.8	126
	Normal	111.2	1509.8	1621
	Total	153.4	1593.6	

Con una desviación estándar de 0.24, cuyos resultados indican que para un escenario de prueba más real, la sensitividad del modelo ABS es de 0.33 y su especificidad es de 0.93.

Finalmente, tenemos los resultados para el experimento del escenario realista del modelo RABS:

		Valor Predicho		
		Ataque	**Normal**	**Total**
Valor Real	**Ataque**	224.2	45.8	270
	Normal	201	1479	1680
	Total	425.2	1524.8	

Con una desviación estándar de 4.24, cuyos resultados indican que, para este escenario de pruebas, el modelo RABS tiene una sensitividad de 0.83 y una especificidad de 0.88.

5.3. Comparación de tiempos de respuesta

Adicionalmente a las pruebas anteriores, también se midió el tiempo de respuesta (medido en paquetes) para los modelos, en la siguiente tabla se puede observar como se desempeñaron ambos:

Nombre del ataque	ABS	RABS
land	80.5	2
syslog	85.1	1.5
udp-storm	86.4	2.5
custom*	112.7	2.75
http-tunnel	109	1.2
sendmail	117.3	2
sql	111	2
ffbconfig	112	1
ffbconfig	112	1.4
ps	110	2
Promedio	103.6	1.8

Esto muestra que, para una respuesta secundaria, en estos escenarios, el modelo RABS solo requiere del 1.73 % del tiempo que requiere el modelo ABS para detectar e identificar la anomalía.

Capítulo 6

Discusión

El primer análisis que se puede extraer de los resultados proviene del experimento de caso ideal realizado sobre el modelo ABS. Este experimento nos entrega una sensitividad de 0.82 y una especificidad de 0.93, con estos datos podemos decir que el modelo ABS tiene un desempeño bastante bueno en condiciones ideales de funcionamiento (ataque corto e intenso). El objetivo que se buscaba con este experimento es demostrar el buen funcionamiento del modelo ABS como detector de anomalías.

Habiendo cumplido el objetivo mencionado, podemos pasar a las pruebas comparativas entre los modelos.

Las pruebas comparativas entregaron resultados muy distintos del caso ideal para el modelo ABS, esta prueba dio una sensitividad de 0.33 y una especificidad de 0.93. La principal razón de esto es que el modelo ABS se adapta al ataque, cuando el ataque no es corto, los agentes tienen tiempo de adaptarse a las nuevas condiciones y eventualmente terminan considerando al ataque como una situación normal, esto es porque el modelo ABS carece de cualquier característica que le permita preservar la normalidad y separarla de manera más clara de un ataque.

Por otro lado, y en el mismo experimento, el modelo RABS obtuvo una sensitividad de 0.83 y una especificidad de 0.88, una mejora sustancial de la sensitividad sin mucho cambio en la especificidad, respecto del modelo ABS. Esto se debe a la capacidad que tiene RABS para preservar la pureza de la normalidad, el modelo tiene mecanismos que inhiben la adaptación de los agentes de normalidad a la situación de ataque, que eventualmente ocasionaría falsos negativos.

Sin embargo, la principal característica diferenciadora de RABS respecto al

modelo ABS es la capacidad de caracterizar amenazas para acelerar detecciones futuras, lo que llamamos la respuesta secundaria del modelo. La efectividad de esto se puede observar al comparar los tiempos de detección de los modelos. El modelo ABS tarda, en promedio, 103.6 paquetes para declarar una anomalía.

Por otro lado, el modelo RABS solo toma 1.8 paquetes en promedio en detectar un ataque ya caracterizado en respuesta secundaria. Esto significa que, en promedio, a RABS solo le toma el 1.73 % del tiempo que le toma al modelo ABS detectar una anomalía, como estamos hablando de un caso de respuesta secundaria, esta detección va acompañada, en el caso de RABS, de una clasificación inmediata del ataque según los ataques que se hayan observado con anterioridad.

Capítulo 7

Conclusiones

En este trabajo se presenta una propuesta de IDS/IPS en base a especies indicadoras artificiales con elementos y conceptos del sistema inmune. Propuesta basada en investigaciones anteriores que apuntaban en una dirección similar.

Se identificaron problemas en anteriores implementaciones de esta idea y se trabajó en solucionarlos, el resultado de esto es el modelo RABS. Este modelo posee la capacidad de detectar anomalías, al igual que versiones anteriores, pero además puede caracterizar amenazas. La caracterización de amenazas permite una respuesta secundaria del modelo que nos otorga varias ventajas, las principales de ellas son la capacidad de identificar de manera más precisa el ataque que está ocurriendo observando la población que responda a él y acelerar el tiempo de detección de amenazas conocidas.

Una característica poco mencionada de RABS, pero no menos importante, es la capacidad de guardar y migrar poblaciones de agentes que hayan caracterizado amenazas, una vez tenemos una población que se adaptó y caracterizó un ataque, entonces podemos tomar esos agentes y copiarlos a otro sistema, de esta forma le conferimos una inmunidad a ese nuevo sistema a un ataque que nunca ha visto, de la misma forma que funciona una vacuna.

Los resultados obtenidos en las pruebas muestran que RABS es superior al modelo ABS, tiene una sensitividad mucho mejor y una especificidad comparable y tiene una enorme ventaja en cuanto a tiempo requerido para la detección de un ataque, en respuesta secundaria.

Podemos decir que se logró tomar los conceptos básicos del modelo original, mejorarlos y trasladarlos a esta nueva implementación que obtuvo resultados que avalan las mejoras logradas.

Finalmente, queda para un trabajo futuro el realizar una prueba más exhaustiva del modelo en un dataset más amplio para validar de manera más completa al modelo, además de llevar el modelo a un lenguaje con mejores capacidades de integración que NETLOGO, algunas opciones para esto pueden ser python o C++, por su simplicidad y utilidad en el caso de python y por su eficiencia computacional en el caso de C++.

Merece atención una mirada más profunda a las medidas de afinidad de los agentes con las partículas. La utilizada en este trabajo es Distancia Hamming pero existen muchas otras medidas de afinidad que se pueden usar y habría que probar diferentes opciones.

Por último, algo que no se analiza en el modelo RABS es la secuencia de paquetes, el modelo solo sabe del paquete actual pero no importa el historial de paquetes, sería interesante observar si se puede obtener información relevante de la secuencia de paquetes que pasan por la red, aquí se entraría a realizar un análisis en dos niveles, un nivel en que se observa el contenido de cada paquete y otro en que se observa la secuencia de paquetes que pasan por la red.

Bibliografía

[1] P. Aggarwal and S.K. Sharma. Analysis of kdd dataset attributes-class wise for intrusion detection. *Procedia Computer Science*, 57:842–851, 2015.

[2] Uwe Aickelin, Peter Bentley, Steve Cayzer, Jungwon Kim, and Julie McLeod. Danger theory: The link between ais and ids? In *International Conference on Artificial Immune Systems*, pages 147–155. Springer, 2003.

[3] Uwe Aickelin and Steve Cayzer. The danger theory and its application to artificial immune systems. *arXiv preprint arXiv:0801.3549*, 2008.

[4] Amira Sayed A Aziz, EL Sanaa, and Aboul Ella Hassanien. Comparison of classification techniques applied for network intrusion detection and classification. *Journal of Applied Logic*, 24:109–118, 2017.

[5] J. Balthrop, F. Esponda, S. Forrest, and M. Glickman. Coverage and generalization in an artificial immune system. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '02)*, pages 3–10, 2002.

[6] Christian Blum, Jose A. Lozano, and Pedro Pinacho. An artificial bioindicator system for network intrusion detection. *Artificial Life*, 21-(2):93–118, 2015.

[7] Burnet FM. The clonal selection theory of acquired immunity. *Cambridge University Press*, 1959.

[8] CISCO. The zettabyte era: Trends and analysis. `https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/vni-hyperconnectivity-wp.html`, 2017.

[9] Leandro N. de Castro and Jonathan Timmis. *Artificial Immune Systems: A New Computational Intelligence Approach.* Springer, 2002.

[10] L.N. de Castro and F.J. Von Zuben. Artificial immune systems: part i-basic theory and applications. *Tech. Rep., Universidade Estadual de Campinas*, 1999.

[11] L.N. de Castro and F.J. Von Zuben. Learning and optimization using the clonal selection principle. *IEEE Trans. Evol. Comput.*, 6(3):239–251, 2002.

[12] L. Dhanabal and S.P. Shantharajah. A study on nsl-kdd dataset for intrusion detection system based on classification algorithms. *International Journal of Advanced Research in Computer and Communication Engineering*, 4(6):446–452, 2015.

[13] P. D'haeseleer, S. Forrest, and P. Helman. Immunological approach to change detection: algorithms, analysis and implications. *Proceedings of the 17th IEEE Symposium on Security and Privacy*, page 110–119, 1996.

[14] Diogo A.B. Fernandes, Mário M. Freire, Paulo A.P. Fazendeiro, and Pedro Inácio. Applications of artificial immune systems to computer security: A survey. *Journal of Information Security and Applications*, 35:138–159, 2017.

[15] S. Forrest, A.S. Perelson, L. Allen, and R. Cherukuri. Self-nonself discrimination in a computer. *Proceedings of the IEEE symposium on security and privacy (SP)*, page 202, 1994.

[16] F. González, D. Dasgupta, and J. Gómez. The effect of binary matching rules in negative selection. *Genetic and Evolutionary Computation-GECCO*, 2723:195–206, 2003.

[17] P.K. Harmer, P.D. Williams, G.H. Gunsch, and G.B. Lamont. An artificial immune system architecture for computer security applications. *IEEE Transactions on Evolutionary Computation*, 6:252–280, 2002.

[18] R. Heady, G. Luger, A. Maccabe, and M. Servilla. The architecture of a network level intrusion detection system. *Los Alamos National Lab*, 1990.

[19] R. Hightower, S. Forrest, and A.S. Perelson. The evolution of secondary organization in immune system gene libraries. *Proceedings of the 2nd European Conference on Artificial Life*, page 458–470, 1994.

[20] S. A. Hofmeyr and S. Forrest. Immunity by design: An artificial immune system. *Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation*, 2:1289–1296, 1999.

[21] O. Igbe, I. Darwish, and T. Saadawi. Distributed network intrusion detection systems: An artificial immune system approach. *CHASE 2019, IEEE Press*, pages 27–29, 2016.

[22] B. Ingre and A. Yadav. Performance analysis of nsl-kdd dataset using ann. *International Conference on Signal Processing and Communication Engineering Systems*, pages 92–96, 2015.

[23] J. Jabez and B. Muthukumar. Intrusion detection system (ids): anomaly detection using outlier detection approach. *Procedia Computer Science*, 48:338–346, 2015.

[24] Z. Ji and D. Dasgupta. Revisiting negative selection algorithms. *Evol Comput*, 15(2):223–251, 2007.

[25] K. Jungwon and P. Bentley. The human immune system and network intrusion detection. *EUFIT*, 99:1244–1252, 1999.

[26] J. Kim and P.J. Bentley. Towards an artificial immune system for network intrusion detection: an investigation of clonal selection. *Proceedings of the Congress on Evolutionary Computation (CEC '02)*, 2:1015–1020, 2002.

[27] J. Kimand and P.J. Bentley. Immunememory in the dynamic clonal selection algorithm. *Proceedings of the 1st International Conference on Artificial Immune Systems (ICARIS '02)*, pages 59–67, 2002.

[28] M. O. Lermanda Sandoval. Especies indicadoras artificiales un clasificador adaptativo basado en vida artificial. *Doctoral dissertation, Universidad de Concepción. Facultad de Ingeniería. Departamento de Ingeniería Informática y Ciencias de la Computación*, 2018.

[29] J. Maestre, A. Sandoval, and L. Garca. Adaptive artificial immune networks for mitigating dos flooding attacks. *Swarm and Evolutionary Computation 38*, pages 94–108, 2018.

[30] P. Matzinger. Essay 1: the danger model in its historical context. *Scandinavian Journal of Immunology*, 54:4–9, 2001.

[31] Polly Matzinger. Tolerance, danger, and the extended family. *Annual review of immunology*, 12(1):991–1045, 1994.

[32] M. Oprea and S. Forrest. How the immune system generates diversity: Pathogen space coverage with random and evolved antibody libraries. *Tech*, 1999.

[33] M.R. Parsaei, S.M. Rostami, and R. Javidan. A hybrid data mining approach for intrusion detection on imbalanced nsl-kdd dataset. *International Journal of Advanced Computer Science and Applications*, 7(6):20–25, 2016.

[34] A.S. Perelson, R. Hightower, and S. Forrest. Evolution and somatic learning in v-region genes. *Research in Immunology*, 147:202–208, 1996.

[35] Pedro Pinacho, Iván Pau, Max Chacón, and Sergio Sánchez. An ecological approach to anomaly detection: the eia model. In *International Conference on Artificial Immune Systems*, pages 232–245. Springer, 2012.

[36] I. Sharafaldin, A. Gharib, A.H. Lashkari, and A.A. Ghorbani. Towards a reliable intrusion detection benchmark dataset. *Software Networking*, 2018(1):177–200, 2018.

[37] I. Sharafaldin, A.H. Lashkari, and A.A. Ghorbani. Toward generating a new intrusion detection dataset and intrusion traffic characterization. *ICISSP*, pages 108–116, 2018.

[38] Steven R Snapp, James Brentano, Gihan Dias, Terrance L Goan, L Todd Heberlein, Che-Lin Ho, and Karl N Levitt. Dids (distributed intrusion detection system)-motivation, architecture, and an early prototype. 1991.

[39] M. Sokolova, N. Japkowicz, and S. Szpakowicz. Beyond accuracy, f-score and roc: a family of discriminant measures for performance evaluation. *Australasian joint conference on artificial intelligence*, pages 1015–1021, 2006.

[40] T. Stibor, P. Mohr, and J. Timmis. Is negative selection appropriate for anomaly detection? *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '05)*, 99:321–328, 2005.

[41] A. Syed and I. Biju. Performance comparison of intrusion detection systems and application of machine learning to snort system. *Future Generation Computer Systems*, 80:157–170, 2018.

[42] M. Tavallaee, E. Bagheri, W. Lu, and A. Ghorbani. A detailed analysis of the kdd cup 99 data set. *Second IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA)*, 2009.

[43] F.G. Varela, H.R. Maturana, and R. Uribe. Autopoiesis: The organization of living systems, its characterization and a model. *Biosystems*, 5(4):187–196, 1974.

[44] H. Yang, T. Li, X. Hu, F. Wang, and Y. Zou. A survey of artificial immune system based intrusion detection. *The Scientific World Journal*, 2014.

[45] J. Yang, X. Liu, T. Li, G. Liang, and S. Liu. Distributed agents model for intrusion detection based on ais. *Knowledge-Based Systems*, 22:115–119, 2009.

[46] A. Özgür and H. Erdem. A review of kdd99 dataset usage in intrusion detection and machine learning between 2010 and 2015. *PeerJ Preprints*, 2016.