Centro de Ciencias Básicas

Sistemas Expertos Probabilísticos
Profesor: Eunice Esther Ponce de León Senti

Tarea #2 – Sistemas Expertos, factibilidad y pasos para su construcción

Universidad Autónoma de Aguascalientes
Ingeniería en Computación Inteligente
Semestre 8° A

Integrantes:
Dante Alejandro Alegría Romero – 265853
Diego Alberto Aranda Gonzalez – 262021
Andrea Margarita Balandrán Félix – 331696
Diego Emilio Moreno Sánchez – 264776

Desarrollo de un Sistema Experto para la Detección de Intrusos en Redes: Ciberseguridad en Sistemas Informáticos

1.- Planteamiento del Problema

Problema a Resolver:

La detección de intrusos en redes informáticas es un problema crítico en el ámbito de la ciberseguridad. Los sistemas informáticos están expuestos a diversas amenazas, como ataques de denegación de servicio (DDoS), intrusiones no autorizadas, malware y robo de información. Los métodos tradicionales de detección, como listas negras y firewalls, no son suficientes para combatir ataques sofisticados. Por lo tanto, es necesario desarrollar un sistema experto capaz de identificar patrones de comportamiento anómalos y prevenir posibles ataques en tiempo real.

2.- Encontrar Expertos Humanos que Puedan Resolver el Problema

Expertos en el Campo:

Para diseñar un sistema experto eficaz, es necesario recopilar conocimientos de especialistas en:

- **Ingeniería en Ciberseguridad**: Profesionales que diseñan estrategias de protección en redes informáticas.
- **Administradores de Redes**: Expertos que gestionan infraestructuras de red y pueden identificar patrones de tráfico sospechosos.
- **Analistas de Seguridad Informática**: Especialistas que estudian vulnerabilidades y crean modelos de detección.
- **Científicos de Datos**: Profesionales que pueden diseñar algoritmos de aprendizaje automático para la detección de anomalías.
- **Hackers Éticos y Penetration Testers**: Profesionales que realizan pruebas de penetración para identificar fallas en la seguridad.
- **Especialistas en Inteligencia Artificial**: Encargados de desarrollar modelos avanzados de detección de intrusiones mediante IA.

Estos expertos pueden proporcionar conocimiento para construir una base de reglas efectiva, definir los indicadores de compromiso (IoC) y entrenar modelos de detección basados en datos históricos y patrones de ataque.

3.- Diseño de un Sistema Experto

El diseño del sistema experto incluirá los siguientes componentes:

1. **Estructuras para Almacenar el Conocimiento**:
    - Base de reglas definidas por expertos en ciberseguridad.

    - Base de datos de incidentes pasados y patrones de ataques.
    - Modelos de aprendizaje automático para la detección de anomalías.
2. **Motor de Inferencia**:
    - Uso de reglas lógicas y técnicas de aprendizaje automático para clasificar eventos sospechosos.
    - Aplicación de algoritmos de detección de anomalías y correlación de eventos.
3. **Subsistema de Explicación**:
    - Justificación de las decisiones del sistema en base a patrones detectados.
    - Registro de auditoría con detalles de ataques identificados.
4. **Interfaz de Usuario**:
    - Panel de monitoreo con alertas en tiempo real.
    - Visualización de patrones de tráfico y reportes de seguridad.

4.- Elección de la Herramienta de Desarrollo, Shell o Lenguaje de Programación

Opciones Consideradas:

Para desarrollar el sistema experto, es fundamental elegir herramientas que permitan una implementación eficiente y escalable. Se consideran las siguientes opciones:

Lenguajes de Programación:

- **Python**: Amplia disponibilidad de librerías para procesamiento de datos, inteligencia artificial y detección de anomalías (Scikit-learn, TensorFlow, Keras, Pandas, NumPy, etc.).
- **Prolog**: Útil para sistemas expertos basados en reglas y lógica de inferencia.
- **Java**: Adecuado para sistemas de seguridad de alto rendimiento y compatibilidad con infraestructuras empresariales.
- **C++**: Optimo para soluciones que requieren alto desempeño y baja latencia.

Frameworks y Herramientas:

- **ELK Stack (Elasticsearch, Logstash, Kibana)**: Para el análisis de registros de eventos y visualización de datos en tiempo real.
- **Splunk**: Herramienta de monitoreo y análisis de seguridad con capacidades de correlación de eventos.

- **Snort y Suricata**: Sistemas de detección de intrusos (IDS) basados en firmas y análisis de tráfico.
- **TensorFlow y PyTorch**: Para la implementación de modelos de inteligencia artificial y detección de patrones en tráfico de red.
- **Metasploit**: Para realizar pruebas de penetración y evaluar la efectividad del sistema experto.

La elección de la herramienta dependerá de los requisitos específicos del sistema, la escalabilidad, facilidad de integración con otras plataformas y el nivel de automatización requerido para la detección de intrusiones.

5.- Desarrollo y Prueba de Prototipo

1. Implementación de una versión inicial del sistema con reglas básicas de detección.
2. Integración de algoritmos de detección de anomalías.
3. Pruebas con datos históricos y en entornos controlados.
4. Evaluación de la precisión y tasas de falsos positivos/negativos.
5. Ajuste del modelo y refinamiento de reglas.

6.- Refinamiento y Generalización

1. Incorporación de nuevos casos de intrusión detectados en tiempo real.
2. Ajuste de umbrales de detección para reducir falsos positivos.
3. Optimización del motor de inferencia para mejorar la velocidad de respuesta.
4. Implementación de aprendizaje continuo para adaptarse a nuevas amenazas.

7.- Mantenimiento y Puesta al Día

1. Monitoreo y corrección de errores detectados por los usuarios.
2. Actualización de la base de reglas con amenazas emergentes.
3. Implementación de mejoras en la interfaz y el motor de inferencia.
4. Integración con nuevas tecnologías y plataformas de seguridad.