

Arquitectura Híbrida para la Detección de Anomalías en Logs de Aplicaciones Web

- Introducción
- Arquitectura Propuesta: Justificación y Diseño
 - Visión General de la Arquitectura
 - Justificación del Enfoque Híbrido
- Desarrollo del Modelo de ML
 - Composición del Dataset
 - Ingeniería de Características
 - Diseño Experimental
 - Modelos Implementados
 - Resultados Experimentales y Análisis
 - Optimización de Hiperparámetros
 - Selección del Modelo
- Limitaciones y Consideraciones Futuras
 - Limitaciones Identificadas
 - Direcciones de Investigación Futura

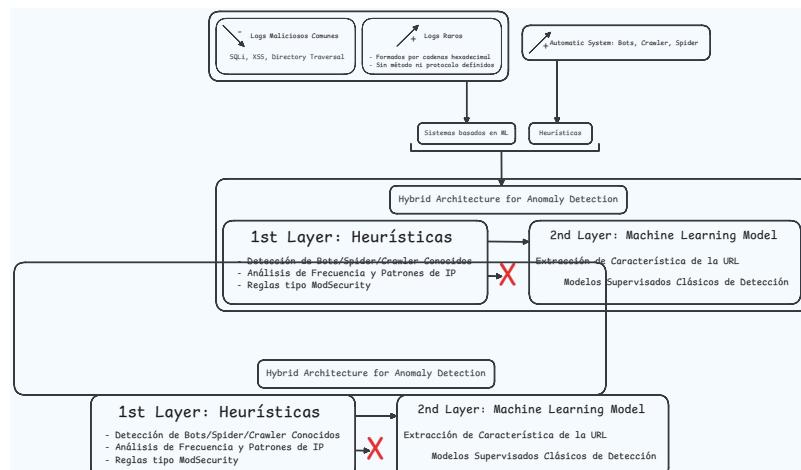
Introducción

La creciente complejidad y volumen de tráfico web ha hecho que la detección de amenazas basada únicamente en firmas o heurísticas tradicionales sea insuficiente. Los ataques modernos emplean técnicas de evasión, bots avanzados y comportamientos aparentemente legítimos que desafían los sistemas de seguridad convencionales. Este trabajo propone una **Arquitectura Híbrida de Dos Capas** que combina la eficiencia de las reglas heurísticas con la capacidad de aprendizaje de modelos de machine learning, optimizando la detección de anomalías en logs de acceso web.

Arquitectura Propuesta: Justificación y Diseño

Visión General de la Arquitectura

La arquitectura implementa un sistema en cascada donde cada capa filtra progresivamente el tráfico:



Justificación del Enfoque Híbrido

Limitación Fuerte de los Sistemas basados en Modelos de ML: Ataques como DoS o Fuerza Bruta no pueden ser detectados, ya que imitan el comportamiento normal.

Ventajas de la Capa Heurística:

- *Bajo Costo Computacional:* 1. Las reglas basadas en listas negras y umbrales simples se ejecutan en tiempo constante $O(1)$
- *Efectividad contra Amenazas Conocidas:* Bots de motores de búsqueda, scrapers automatizados y ataques de fuerza bruta son fácilmente identificables
- *Reducción de Falsos Positivos:* Comportamientos claramente maliciosos son eliminados antes de llegar a modelos complejos
- *Interpretabilidad:* Cada decisión es explicable mediante reglas concretas

Ventajas de la Capa de Machine Learning:

- *Detección de Patrones Complejos:* Identifica correlaciones no evidentes para reglas heurísticas
- *Adaptabilidad:* Aprende de nuevos patrones de ataque sin reescribir reglas manualmente
- *Reducción de Falsos Negativos:* Detecta ataques sofisticados que evaden reglas simples

Justificación del Flujo en Cascada:

- *Eficiencia:* El 30 – 40% del tráfico web actual proviene de bots automáticos. Filtrarlos en la primera capa reduce la carga computacional para el modelo
- *Precisión:* Los modelos ML pueden enfocarse en casos más sutiles y complejos, en vez de crear un modelo para detectar ataques como DoS o Fuerza Bruta, que requerirían considerar el tiempo y la IP del usuario en el entrenamiento.

Desarrollo del Modelo de ML

Composición del Dataset

El conjunto de datos utilizado para el entrenamiento de los modelos de aprendizaje automático se compone de registros de acceso web estructurados, meticulosamente etiquetados mediante un proceso de validación manual. La distribución inicial presenta un escenario de desbalanceo característico de los sistemas de detección de intrusiones, donde las instancias anómalas representan una minoría significativa:

- *Instancias Normales:* 85%
- *Instancias Anómalas:* 15%

Esta distribución refleja un escenario realista donde las actividades maliciosas constituyen una fracción minoritaria pero crítica del tráfico total, planteando desafíos significativos para los algoritmos de clasificación tradicionales.

Ingeniería de Características

Se desarrolló un conjunto de 23 características, organizadas en tres categorías principales que capturan diferentes dimensiones del comportamiento HTTP:

- *Características Estructurales de URL:* Medidas de complejidad sintáctica: longitud total, distribución de caracteres (dígitos, letras, símbolos especiales); indicadores de codificación y ofuscación: presencia de codificación URL, proporción de caracteres no convencionales; y patrones de construcción: frecuencia de símbolos específicos (interrogación, porcentaje, igual, guion)
- *Indicadores de Contenido Sospechoso:* Marcadores léxicos de ataques conocidos: términos asociados con inyección SQL, cross-site scripting (XSS), ejecución remota de comandos; patrones de autenticación y autorización: vocabulario típico de intentos de escalamiento de privilegios; e indicadores de exploración maliciosa: terminología relacionada con directorios, archivos de configuración y errores del sistema
- *Metadatos de Petición:* Métodos HTTP normalizados mediante codificación categórica, y volumen de respuesta medido en bytes transmitidos

Diseño Experimental

Pipeline de Evaluación:

1. *Validación Cruzada Estratificada* con 5 folds
2. *Balanceo con SMOTE* para manejar el desbalance de clases

Oversampling Sintético (SMOTE): Generación de instancias sintéticas de la clase minoritaria mediante interpolación en el espacio de características, aplicada exclusivamente durante la fase de entrenamiento de cada fold para evitar contaminación de datos

Métricas de Evaluación: Según el Estado del Arte de Sistemas de Detección de Anomalías en Access Log, se deben tener en cuenta las siguientes métricas:

- *Accuracy*: Proporción de predicciones correctas
- *Precision*: Capacidad de no etiquetar normales como anómalos
- *Recall*: Capacidad de encontrar todas las anomalías
- *F1-Score*: Media armónica de precision y recall
- *AUC-ROC*: Capacidad de distinguir entre clases

Modelos Implementados

A partir del Estado del Arte, se seleccionaron los siguientes modelos para analizar sus resultados frente al dataset: Regresión Logística y Random Forest. Se seleccionó un modelo adicional en el que no se ha encontrado estudios que lo analicen en este dominio: XGBoost (Gradient Boosting con ajuste de peso para clases desbalanceadas). La inclusión de este modelo responde a la hipótesis de que las estructuras de boosting gradient, con su enfoque iterativo de corrección de errores residuales, pueden identificar patrones de anomalía sutiles que los métodos basados en bagging o modelos lineales pasan por alto

Se implementaron otros modelos como: SVM con una configuración con kernel tanto lineal como radial (RBF) y Redes Neuronales. Para la exploración de enfoques complementarios también se incluyeron algoritmos de detección de anomalías no-supervisada: Isolation Forest (basada en el principio de aislamiento mediante particiones aleatorias del espacio de características), One-Class SVM (modelización de la distribución normal como un hiper-esfera en espacio transformado, con detección de outliers como desviaciones significativas), Autoencoder (red neuronal reconstructiva que aprende una representación comprimida de los patrones normales, utilizando el error de reconstrucción como indicador de anomalía)

Resultados Experimentales y Análisis

Modelo	Accuracy	Precision	Recall	F1-Score	AUC
Random Forest	0.9884 ± 0.0002	0.9308 ± 0.0012	0.9950 ± 0.0004	0.9619 ± 0.0006	0.9998
XGBoost	0.9962 ± 0.0002	0.9837 ± 0.0010	0.9907 ± 0.0008	0.9872 ± 0.0006	0.9998
Regresión Logística	0.8958 ± 0.0032	0.5942 ± 0.0085	0.9128 ± 0.0019	0.7198 ± 0.0062	0.9734

Análisis de Capacidades Discriminativas: La evaluación del área bajo la curva ROC demostró excelente capacidad discriminativa en todos los modelos evaluados, con valores superiores a 0.97. Esta métrica, independiente del umbral de clasificación, confirma la robustez de las representaciones aprendidas para separar comportamientos normales y anómalos

Optimización de Hiperparámetros

Mediante la búsqueda aleatoria sistemática, se identificaron configuraciones óptimas que mejoraron sustancialmente el rendimiento:

- *Random Forest*: Incremento del F1-Score a 0.9874 mediante ajuste del número de estimadores (413), profundidad no definida, y criterios de división más estrictos

- *Regresión Logística*: Mejora a F1-Score de 0.7251 con factor de regularización $C = 14.65$ y penalización L2

Selección del Modelo

Considerando el equilibrio entre rendimiento, interpretabilidad y eficiencia, se seleccionó Random Forest como modelo principal para despliegue en producción.

Limitaciones y Consideraciones Futuras

Limitaciones Identificadas

La principal limitación es:

- Los access log no poseen el contenido del POST por lo que no es posible con garantía segura si los POST son maliciosos o no, pero se pueden reconocer algunos como anomalías, lo que no es posible identificar si es un ataque malicioso o no

Observaciones:

- Dependencia de Calidad de Etiquetado

Direcciones de Investigación Futura

- *Aprendizaje Semi-supervisado*: Aprovechamiento de instancias no etiquetadas para mejorar generalización
- *Modelos de Secuencias Temporales*: Análisis de patrones de comportamiento a nivel de sesión mediante arquitecturas recurrentes
- *Explicabilidad Avanzada*: Desarrollo de justificaciones intuitivas para decisiones de clasificación
- *Detección de Ataques Zero-Day*: Incorporación de técnicas de detección de novedades (novelty detection) para identificar amenazas previamente no observadas