

Unidad 1: Introducción al Machine Learning y la Ciberseguridad

Adarsh KUMAR

Universitat Politècnica de Catalunya
Department of Computer Science

Coordinador del proyecto:
Prof. Ilker Demirkol

Proyecto MERiT
3 de septiembre de 2025

Co-funded by the European Union

[Diapositiva 2: Índice]

Fundamentos | Conceptos básicos de Machine Learning | ML y Ciberseguridad | Desafíos de la Ciberseguridad | Pipeline de ML

Índice

1. Fundamentos

2. Conceptos básicos de Machine Learning

3. ML y Ciberseguridad

4. Desafíos de la Ciberseguridad

5. Pipeline de ML

[Diapositiva 3: Fundamentos de Ciberseguridad]

Fundamentos de Ciberseguridad

Definición

La ciberseguridad protege los sistemas, las redes y los datos frente a ataques digitales.

Conceptos clave

- **Amenazas:** intencionadas (hackers), no intencionadas (errores humanos) o naturales (desastres).
- **Ataques:** acciones que explotan vulnerabilidades (p. ej., MITM, phishing, ransomware).
- **Vulnerabilidades:** debilidades en el software, hardware o los procesos.

Fórmula del riesgo

Riesgo = Amenaza × Vulnerabilidad × Impacto

(Mitigación: reducir las vulnerabilidades, monitorizar las amenazas y minimizar el impacto.)

[Diapositiva 4: Ejemplo numérico]

Ejemplo numérico: Amenaza vs. Vulnerabilidad vs. Ataque

Escenario

Una empresa tiene 100 cuentas de correo electrónico de empleados.

Vulnerabilidad

25 de estas cuentas utilizan contraseñas débiles como 123456 o password.

Co-funded by the European Union

→ **25 cuentas vulnerables.**

Amenaza

Un hacker crea un script para adivinar contraseñas utilizando una lista de credenciales filtradas.
→ **El hacker es la amenaza.**

Ataque

El hacker consigue comprometer 5 de las 25 cuentas vulnerables.
→ **Esto constituye el ataque real.**

[Diapositiva 5: Amenazas, ataques y vulnerabilidades]

Amenazas, ataques y vulnerabilidades

Amenaza → Vulnerabilidad → Ataque → Activo objetivo

- **Amenaza:** motiva
- **Vulnerabilidad:** permite
- **Ataque:** afecta al activo objetivo

Definiciones

Amenaza
Una causa potencial de un incidente no deseado.

Vulnerabilidad
Una debilidad que puede ser explotada por las amenazas.

Ataque
Una acción realizada para explotar una vulnerabilidad.

[Diapositiva 6: Ejemplos]

Ejemplos de amenazas, ataques y vulnerabilidades

Amenazas	Ataques	Vulnerabilidades
• Malware	• Ingeniería social	• Sistema operativo sin actualizar
• Phishing	• Inyección SQL	• Credenciales predeterminadas
• Ataques DoS	• APT	• APIs inseguras

Comprender estos componentes es fundamental para la gestión del riesgo de ciberseguridad.

Co-funded by
the European Union

[Diapositiva 7: Aprendizaje supervisado]

Aprendizaje supervisado en ciberseguridad: un ejemplo numérico

Definición

El **aprendizaje supervisado** es un tipo de Machine Learning en el que un modelo se entrena con un conjunto de datos etiquetado. Cada ejemplo de entrenamiento incluye características de entrada y una etiqueta de salida asociada. El modelo aprende a predecir la etiqueta para datos nuevos que no ha visto previamente.

Escenario de ciberseguridad

Objetivo: detectar si un intento de inicio de sesión es legítimo o malicioso a partir de datos históricos de inicio de sesión etiquetados.

[Diapositiva 8: Ejemplo de aprendizaje supervisado]

Aprendizaje supervisado en ciberseguridad: un ejemplo numérico

Datos de entrenamiento de ejemplo

Hora de inicio de sesión (24 h)	Puntuación de reputación IP	Éxito (1/0)	Etiqueta (0: legítimo, 1: malicioso)
09	0.9	1	0
23	0.2	0	1
17	0.7	1	0
03	0.1	0	1

Proceso de aprendizaje supervisado

- **Características:** hora de inicio de sesión, puntuación de reputación IP, éxito/fallo.
- **Etiqueta:** legítimo (0) o malicioso (1).
- **Modelo:** clasificador entrenado (p. ej., árbol de decisión).
- **Ejemplo de predicción:** nuevo inicio de sesión a las 03 h, puntuación IP = 0.1, inicio de sesión fallido ⟶ **Predicción: malicioso (1).**

Co-funded by the European Union

[Diapositiva 9: Aprendizaje no supervisado]

Aprendizaje no supervisado en ciberseguridad: un ejemplo numérico

Definición

El **aprendizaje no supervisado** es un tipo de Machine Learning en el que el modelo se entrena con datos sin etiquetas de salida. El objetivo es identificar patrones o estructuras ocultas (p. ej., agrupaciones o anomalías) dentro de los datos.

Escenario de ciberseguridad

Objetivo: agrupar los intentos de inicio de sesión en grupos normales y sospechosos utilizando datos de inicio de sesión sin etiquetar.

[Diapositiva 10: Ejemplo de aprendizaje no supervisado]

Aprendizaje no supervisado en ciberseguridad: un ejemplo numérico

Datos de entrada sin etiquetar

Hora de inicio de sesión (24 h)	Puntuación de reputación IP	Éxito (1/0)
09	0.9	1
23	0.2	0
17	0.7	1
03	0.1	0
02	0.1	0

Proceso de aprendizaje no supervisado

- No se proporcionan etiquetas al modelo.
- Se aplica un algoritmo de clustering (p. ej., **K-Means**).
- El modelo agrupa los intentos de inicio de sesión en 2 grupos:
    - **Grupo A:** probablemente normal (p. ej., inicios de sesión durante el día, puntuación IP alta).
    - **Grupo B:** probablemente sospechoso (p. ej., inicios de sesión nocturnos, puntuación IP baja, intentos fallidos).
- Los analistas de seguridad investigan posteriormente el **Grupo B**.

Co-funded by the European Union

[Diapositiva 11: Aprendizaje por refuerzo]

Aprendizaje por refuerzo en ciberseguridad: un ejemplo numérico

Definición

El **aprendizaje por refuerzo** es un tipo de Machine Learning en el que un agente aprende a realizar acciones en un entorno para maximizar una recompensa acumulada.

Aprende a partir de retroalimentación en forma de recompensas (positivas o negativas), en lugar de utilizar ejemplos etiquetados.

Escenario de ciberseguridad

Objetivo: entrenar un sistema autónomo para detectar y responder ante intrusiones en un entorno de red dinámico.

[Diapositiva 12: Ejemplo de aprendizaje por refuerzo]

Aprendizaje por refuerzo en ciberseguridad: un ejemplo numérico

Ejemplo: agente de detección de intrusiones

Estado	**Acción realizada**	**Recompensa**
Tráfico alto + IP desconocida	Bloquear IP	+10
Tráfico normal	Permitir	+5
Tráfico alto + IP conocida	Bloquear IP	-5
Escaneo de puertos sospechoso	Aislar host	+8

Proceso de aprendizaje por refuerzo

- El agente observa el estado de la red.
- Elige una acción (p. ej., bloquear, permitir, aislar).
- Recibe una recompensa en función de la eficacia de la acción.
- Aprende una política para maximizar la recompensa a largo plazo (p. ej., **Q-learning, Deep Q-Network**).

Co-funded by the European Union

[Diapositiva 13: Fundamentos de Machine Learning]

Fundamentos de Machine Learning

▸ **Concepto fundamental:** sistemas que aprenden patrones a partir de datos sin necesidad de programación explícita.

▸ **Aprendizaje supervisado**

- **Clasificación:** detección de spam, categorización de malware.
- **Regresión:** predicción de probabilidades de ataque.

▸ **Aprendizaje no supervisado**

- **Clustering:** detección de anomalías en el tráfico de red.
- **Reducción de dimensionalidad:** selección de características para la detección de amenazas.

▸ **Aprendizaje por refuerzo**

- Pentesting adaptativo.
- Optimización dinámica de estrategias de defensa.

Cada paradigma tiene aplicaciones específicas en ciberseguridad.

[Diapositiva 14: ML para Ciberseguridad]

ML para Ciberseguridad

Aprendizaje supervisado

- Clasificación de malware
- Sistemas de detección de intrusiones (IDS)
- Detección de spam/phishing

Aprendizaje no supervisado

- Detección de anomalías
- Clustering de flujos de red
- Detección de amenazas internas

Aprendizaje por refuerzo

- Firewalls adaptativos
- Búsqueda automatizada de amenazas

Fuentes de datos

- Logs del sistema
- Tráfico de red
- Telemetría de endpoints

Desafíos técnicos

- Ataques adversarios
- Concept Drift
- Calidad de los datos

Consideraciones éticas

- Problemas de privacidad
- Sesgo en los modelos
- Explicabilidad

Ciberseguridad y Machine Learning

[Diapositiva 15: Tipos de ML y Ciberseguridad]

Tipos de Machine Learning y Ciberseguridad

Aprendizaje supervisado	**Aprendizaje no supervisado**	**Aprendizaje por refuerzo**
• Detección de spam (análisis del contenido del correo electrónico)	• Detección de anomalías (patrones del tráfico de red)	• Defensa adaptativa (ajuste dinámico de reglas)
• Clasificación de malware (análisis estático/dinámico)	• Clustering de amenazas (agrupación de IoC)	• Optimización de honeypots (interacción con atacantes)
• Detección de fraude (monitorización de transacciones)	• Modelado de líneas base (perfiles de comportamiento normal)	• Automatización de pentesting (descubrimiento de rutas de ataque)

Cada enfoque de ML aborda diferentes desafíos de seguridad.

[Diapositiva 16: Casos de uso]

Machine Learning en Ciberseguridad: principales casos de uso

Sistemas de detección

- **Detección de anomalías:** identificar patrones inusuales en el tráfico de red.
- **Clasificación de malware:** categorizar malware mediante análisis estático/dinámico.
- **Detección de phishing (NLP):** analizar el contenido de correos electrónicos y las URL.

Mejora de la seguridad

- **Mejora de IDS:** reducir los falsos positivos en la detección de intrusiones.
- **Correlación de inteligencia sobre amenazas:** conectar indicadores de compromiso.
- **Análisis del comportamiento de los usuarios (UEBA):** detectar amenazas internas.

ML permite una seguridad proactiva mediante el reconocimiento de patrones y la automatización.

[Diapositiva 17: Limitaciones técnicas]

Machine Learning en Ciberseguridad: principales limitaciones

Desafíos técnicos

- **Problemas de calidad de los datos:** *Garbage in, garbage out*.
- **Ataques adversarios:** envenenamiento/evasión del modelo.
- **Falsas alarmas:** tanto falsos positivos como falsos negativos.

Desafíos técnicos (con ejemplos numéricos)

- **Problemas de calidad de los datos (Garbage in, garbage out):**
  Un conjunto de datos contiene 10.000 intentos de inicio de sesión, pero 3.000 registros tienen direcciones IP ausentes o resultados etiquetados incorrectamente. Esto induce al modelo a errores y reduce la precisión de detección.
- **Ataques adversarios (envenenamiento/evasión del modelo):**
  Un atacante inyecta 500 muestras especialmente diseñadas y aparentemente benignas en el conjunto de entrenamiento. Como resultado, el modelo entrenado clasifica incorrectamente el 70%.
- **Falsas alarmas (falsos positivos/negativos):**
  En una prueba con 1.000 eventos, el modelo genera 150 alertas:
  - 40 son verdaderos positivos (amenazas reales detectadas).
  - 30 son falsos negativos (amenazas no detectadas).
  - 80 son falsos positivos (acciones benignas marcadas como amenazas).

Co-funded by the European Union

Esto da como resultado un **53,3%**.

[Diapositiva 18: Limitaciones técnicas]

Machine Learning en Ciberseguridad: principales limitaciones

Desafíos técnicos

- **Problemas de calidad de los datos:** *Garbage in, garbage out*.
- **Ataques adversarios:** envenenamiento/evasión del modelo.
- **Falsas alarmas:** tanto falsos positivos como falsos negativos.

Desafíos técnicos (con ejemplos numéricos)

- **Problemas de calidad de los datos (Garbage in, garbage out):**
  Un conjunto de datos contiene 10.000 intentos de inicio de sesión, pero 3.000 registros tienen direcciones IP ausentes o resultados etiquetados incorrectamente. Esto induce al modelo a errores y reduce la precisión de detección.
- **Ataques adversarios (envenenamiento/evasión del modelo):**
  Un atacante inyecta 500 muestras especialmente diseñadas y aparentemente benignas en el conjunto de entrenamiento. Como resultado, el modelo entrenado clasifica incorrectamente el 70%.
- **Falsas alarmas (falsos positivos/negativos):**
  En una prueba con 1.000 eventos, el modelo genera 150 alertas:
  - 40 son verdaderos positivos (amenazas reales detectadas).
  - 30 son falsos negativos (amenazas no detectadas).
  - 80 son falsos positivos (acciones benignas marcadas como amenazas).

Esto da como resultado un **53,3%**.

[Diapositiva 19: Factores operativos]

Machine Learning en Ciberseguridad: principales limitaciones

Factores operativos

- **Concept Drift:** evolución del panorama de amenazas.
- **Explicabilidad:** toma de decisiones de tipo caja negra.

Factores operativos (con ejemplos numéricos)

- **Concept Drift (evolución del panorama de amenazas):**
  Un modelo de detección de malware fue entrenado con patrones de ransomware de 2023. En 2025, los atacantes comienzan a utilizar malware polimórfico. Como resultado, la precisión de detección cae del 95% al 63% en nuevos incidentes sobre un conjunto de prueba de 1.000 muestras. Las actualizaciones periódicas del modelo se vuelven esenciales.
- **Explicabilidad (toma de decisiones de caja negra):**
  Una red neuronal marca como malicioso el inicio de sesión de un usuario. Cuando el analista de seguridad pregunta por qué, el sistema no puede explicar su razonamiento. El mes pasado se generaron más de 200 alertas de este tipo y el 60% fueron falsos positivos, lo que reduce la confianza en el sistema y retrasa los tiempos de respuesta.

[Diapositiva 20: Factores operativos]

Machine Learning en Ciberseguridad: principales limitaciones

Factores operativos

- **Concept Drift:** evolución del panorama de amenazas.
- **Explicabilidad:** toma de decisiones de tipo caja negra.

Factores operativos (con ejemplos numéricos)

- **Concept Drift (evolución del panorama de amenazas):**
  Un modelo de detección de malware fue entrenado con patrones de ransomware de 2023. En 2025, los atacantes comienzan a utilizar malware polimórfico. Como resultado, la precisión de detección cae del 95% al 63% en nuevos incidentes sobre un conjunto de prueba de 1.000 muestras. Las actualizaciones periódicas del modelo se vuelven esenciales.
- **Explicabilidad (toma de decisiones de caja negra):**
  Una red neuronal marca como malicioso el inicio de sesión de un usuario. Cuando el analista de seguridad pregunta por qué, el sistema no puede explicar su razonamiento. El mes pasado se generaron más de 200 alertas de este tipo y el 60% fueron falsos positivos, lo que reduce la confianza en el sistema y retrasa los tiempos de respuesta.

[Diapositiva 21: Consideraciones éticas]

Machine Learning en Ciberseguridad: principales limitaciones

Consideraciones éticas

- **Riesgos de privacidad:** problemas relacionados con la recopilación de datos.
- **Sesgo:** limitaciones de los datos de entrenamiento.
- **Responsabilidad:** responsabilidad sobre las decisiones.

Co-funded by
the European Union

Consideraciones éticas (con ejemplos numéricos)

- **Riesgos de privacidad (problemas relacionados con la recopilación de datos):**
  Un sistema de detección de amenazas recopila 5 millones de inicios de sesión de usuarios al mes, incluidas IP, ubicaciones e identificadores de dispositivos. Una regla de registro mal configurada expone 100.000 de estos registros a desarrolladores internos, infringiendo el GDPR y dando lugar a una multa de cumplimiento de 250.000 $.
- **Sesgo (limitaciones de los datos de entrenamiento):**
  Un modelo de detección de phishing se entrena principalmente con correos electrónicos procedentes de regiones occidentales. Al evaluarlo con 10.000 muestras de organizaciones de Asia, alcanza solo un 55% de precisión frente al 92% obtenido con muestras occidentales, debido al sesgo relacionado con el idioma y el formato regional.
- **Responsabilidad (responsabilidad sobre las decisiones):**
  Un sistema automatizado bloquea a 200 usuarios basándose en comportamientos de inicio de sesión sospechosos. Posteriormente se descubre que 15 eran empleados legítimos que accedían mediante VPN. Sin un mecanismo de revisión o anulación humana, la empresa se enfrenta a interrupciones operativas y reclamaciones internas.

[Diapositiva 22: Consideraciones éticas]

Machine Learning en Ciberseguridad: principales limitaciones

Consideraciones éticas

- **Riesgos de privacidad:** problemas relacionados con la recopilación de datos.
- **Sesgo:** limitaciones de los datos de entrenamiento.
- **Responsabilidad:** responsabilidad sobre las decisiones.

Consideraciones éticas (con ejemplos numéricos)

- **Riesgos de privacidad (problemas relacionados con la recopilación de datos):**
  Un sistema de detección de amenazas recopila 5 millones de inicios de sesión de usuarios al mes, incluidas IP, ubicaciones e identificadores de dispositivos. Una regla de registro mal configurada expone 100.000 de estos registros a desarrolladores internos, infringiendo el GDPR y dando lugar a una multa de cumplimiento de 250.000 $.
- **Sesgo (limitaciones de los datos de entrenamiento):**
  Un modelo de detección de phishing se entrena principalmente con correos electrónicos procedentes de regiones occidentales. Al evaluarlo con 10.000 muestras de organizaciones de Asia, alcanza solo un 55% de precisión frente al 92% obtenido con muestras occidentales, debido al sesgo relacionado con el idioma y el formato regional.
- **Responsabilidad (responsabilidad sobre las decisiones):**
  Un sistema automatizado bloquea a 200 usuarios basándose en comportamientos de inicio de

sesión sospechosos. Posteriormente se descubre que 15 eran empleados legítimos que accedían mediante VPN. Sin un mecanismo de revisión o anulación humana, la empresa se enfrenta a interrupciones operativas y reclamaciones internas.

[Diapositiva 23: Tipos de datos de seguridad]

Tipos de datos de seguridad: la base de la ciberseguridad

Datos operativos

- **Logs del sistema:** autenticación, aplicaciones y eventos de AV.
- **Datos de red:**
    - Flujos (NetFlow/IPFIX)
    - Paquetes (análisis mediante Wireshark/Zeek)
- **Registros de endpoints:** telemetría EDR, monitorización de archivos/procesos.

Inteligencia de seguridad

- **Feeds de amenazas:** IOCs (IP, dominios, hashes).
- **Datos de alertas:** firmas IDS/IPS, correlaciones SIEM.
- **Escaneos de vulnerabilidades:** evaluación de activos y configuraciones.

Características clave

- **Volumen:** TB/día en empresas.
- **Velocidad:** necesidades de procesamiento en tiempo real.
- **Variedad:** estructurados/no estructurados.
- **Veracidad:** ruido y falsos positivos.

Un análisis de seguridad eficaz requiere correlacionar estos diferentes tipos de datos.

[Diapositiva 24: Tipos de datos de seguridad]

Tipos de datos de seguridad: la base de la ciberseguridad

Datos operativos

- **Logs del sistema:** autenticación, aplicaciones y eventos de AV.
- **Datos de red:**
    - Flujos (NetFlow/IPFIX)

    - Paquetes (análisis mediante Wireshark/Zeek)
- **Registros de endpoints:** telemetría EDR, monitorización de archivos/procesos.

Inteligencia de seguridad

- **Feeds de amenazas:** IOCs (IP, dominios, hashes).
- **Datos de alertas:** firmas IDS/IPS, correlaciones SIEM.
- **Escaneos de vulnerabilidades:** evaluación de activos y configuraciones.

Características clave

- **Volumen:** TB/día en empresas.
- **Velocidad:** necesidades de procesamiento en tiempo real.
- **Variedad:** estructurados/no estructurados.
- **Veracidad:** ruido y falsos positivos.

Un análisis de seguridad eficaz requiere correlacionar estos diferentes tipos de datos.

[Diapositiva 25: Tipos de datos de seguridad]

Tipos de datos de seguridad: la base de la ciberseguridad

Datos operativos

- **Logs del sistema:** autenticación, aplicaciones y eventos de AV.
- **Datos de red:**
    - Flujos (NetFlow/IPFIX)
    - Paquetes (análisis mediante Wireshark/Zeek)
- **Registros de endpoints:** telemetría EDR, monitorización de archivos/procesos.

Inteligencia de seguridad

- **Feeds de amenazas:** IOCs (IP, dominios, hashes).
- **Datos de alertas:** firmas IDS/IPS, correlaciones SIEM.
- **Escaneos de vulnerabilidades:** evaluación de activos y configuraciones.

Características clave

- **Volumen:** TB/día en empresas.
- **Velocidad:** necesidades de procesamiento en tiempo real.
- **Variedad:** estructurados/no estructurados.

- **Veracidad:** ruido y falsos positivos.

Un análisis de seguridad eficaz requiere correlacionar estos diferentes tipos de datos.

[Diapositiva 26: Datos operativos]

Datos operativos (con ejemplos numéricos)

Entradas de ejemplo de un log del sistema de autenticación

Marca temporal	Nombre de usuario	IP de origen	Resultado	Método
2025-06-17 08:13:45	jsmith	192.168.10.5	Éxito	Contraseña
2025-06-17 08:14:12	jsmith	192.168.10.5	Fallido	Contraseña
2025-06-17 08:15:02	asharma	172.16.4.22	Éxito	OTP
2025-06-17 08:17:55	tgupta	203.0.113.25	Fallido	Contraseña
2025-06-17 08:18:20	tgupta	203.0.113.25	Fallido	Contraseña

[Diapositiva 27: Datos operativos]

Datos operativos (con ejemplos numéricos)

Entradas de ejemplo de un log del sistema de aplicaciones

Marca temporal	Nombre de usuario	Aplicación	Tipo de evento	Estado
2025-06-17 09:03:21	jsmith	CRM Portal	Inicio de sesión	Éxito
2025-06-17 09:04:10	jsmith	CRM Portal	Descarga de archivo	Éxito
2025-06-17 09:05:45	asharma	ERP System	Exportación de informe	Éxito
2025-06-17 09:06:00	tgupta	Webmail	Inicio de sesión	Fallido
2025-06-17 09:07:18	tgupta	Webmail	Inicio de sesión	Fallido

[Diapositiva 28: Eventos de antivirus]

Datos operativos (con ejemplos numéricos)

Entradas de ejemplo de un log de eventos de antivirus (AV)

Marca temporal	Nombre del host	Nombre del archivo	Amenaza detectada	Acción realizada

2025-06-17 10:12:45	LAPTOP-22A1	invoice.exe	Trojan/Emotet	Cuarentena
2025-06-17 10:15:08	DESKTOP-R2X4	macro.docm	W97M.Downloader	Eliminado
2025-06-17 10:17:30	SERVER-FIN01	backup.zip	Heur.Generic	Bloqueado
2025-06-17 10:19:02	LAPTOP-22A1	game_crack.exe	Trojan/Generic	Cuarentena
2025-06-17 10:21:55	DESKTOP-R2X4	autorun.inf	Worm.E	Eliminado

---

[Diapositiva 29: Datos de red]

Datos de red (con ejemplo numérico)

Entradas de ejemplo de logs NetFlow/IPFIX

Hora de inicio	IP origen	IP destino	Protocolo	Bytes	Paquetes
10:01:12	192.168.1.10	10.0.0.5	TCP (443)	5.432	12
10:01:15	192.168.1.10	8.8.8.8	UDP (53)	123	1
10:01:18	10.0.0.5	192.168.1.10	TCP (443)	6.102	14
10:01:21	192.168.1.10	203.0.113.9	TCP (80)	2.000	5

Datos de ejemplo a nivel de paquete (análisis Wireshark/Zeek)

N.º	Tiempo	Origen	Destino	Protocolo	Información
1	0.000000	192.168.1.10	8.8.8.8	DNS	Consulta estándar A google.com
2	0.023145	8.8.8.8	192.168.1.10	DNS	Respuesta estándar a consulta A 142.250.64.78
3	0.512832	192.168.1.10	142.250.64.78	TCP	SYN, Seq=0, Win=64240
4	0.514002	142.250.64.78	192.168.1.10	TCP	SYN, ACK, Seq=0, Ack=1

[Diapositiva 30: Características de los datos de seguridad]

Características de los datos de seguridad

Tipo de datos	Granularidad	Uso principal	Desafíos clave
Logs	Media	• Registros de auditoría • Detección	• Falta de contexto • Variedad de formatos
Flujos de red	Baja	• Análisis de tráfico • Líneas base	• No contienen payloads • Problemas de muestreo

Co-funded by the European Union

Datos de paquetes	Alta	• Inspección profunda • Análisis forense	• Necesidades de almacenamiento • Limitaciones de privacidad
Datos de endpoints	Alta	• Visibilidad del host • Seguimiento de procesos	• Coste de recursos • Escala de despliegue
Inteligencia sobre amenazas	Externa	• Coincidencia de IOC • Enriquecimiento de alertas	• Trabajo de integración • Comprobaciones de relevancia
Alertas de seguridad	Alto nivel	• Triaje de incidentes • Priorización	• Falsas alarmas • Fatiga por alertas

La combinación estratégica de datos permite obtener una visibilidad integral de las amenazas.

[Diapositiva 31: Pipeline de Machine Learning]

Pipeline de Machine Learning para aplicaciones de seguridad

1. Definición y alcance del problema

Identificar un desafío de seguridad específico (p. ej., detección de malware) y las métricas de éxito.

Ejemplo: detección de malware → criterios de éxito.

2. Recopilación y etiquetado de datos

Recopilar datos de seguridad relevantes (logs, tráfico de red) con datos de *ground truth* adecuados.

Ejemplo: logs de seguridad, tráfico de red, datos de *ground truth*.

3. Ingeniería de características

Extraer patrones significativos (IOCs, firmas de comportamiento) a partir de datos de seguridad sin procesar.

Ejemplo: extraer indicadores (IOCs) y patrones de comportamiento.

4. Desarrollo del modelo

Seleccionar el algoritmo y aplicar restricciones específicas de seguridad.

Ejemplo: elegir el algoritmo y aplicar restricciones específicas de seguridad.

5. Evaluación y validación

Utilizar métricas centradas en la seguridad.

Ejemplo: tasa de detección, falsas alarmas y resistencia a la evasión.

6. Despliegue operativo

Integración con el SOC y monitorización en tiempo real.

Bucle continuo de retroalimentación

Reentrenar con nuevos patrones de amenazas y adaptarse a ataques en evolución.

[Diapositiva 32: Pipeline de ML — continuación]

Pipeline de Machine Learning para aplicaciones de seguridad — continuación

4. Desarrollo del modelo

- **Selección del algoritmo** (p. ej., Random Forest para clasificación).
- **Entrenamiento con restricciones específicas de seguridad** (límites FAR).

5. Evaluación y validación

Métricas centradas en la seguridad:

- Tasa de detección.
- Tasa de falsas alarmas.
- Resistencia a la evasión.

6. Despliegue operativo

Integración con los flujos de trabajo del SOC y monitorización del rendimiento en tiempo real.

Bucle continuo de retroalimentación

- Reentrenar con nuevos patrones de amenazas.
- Adaptarse a ataques en evolución.

El bucle continuo de retroalimentación mejora el modelo frente a las amenazas en evolución.

[Diapositiva 33: Ejemplo numérico del pipeline]

Ejemplo numérico: Pipeline de ML para la detección de intrusiones

Etapa	Explicación	Ejemplo numérico
1. Definición del problema	Detectar intrusiones de red a partir de datos de paquetes	**Objetivo:** clasificar las conexiones como normales (0) o ataques (1)
2. Recopilación de datos	Utilizar el conjunto de datos KDD Cup 99 o logs internos de red	Se recopilan 1.000 registros; 700 etiquetados como “0” y 300 como “1”
3. Ingeniería de	Extraer campos relevantes del flujo	P. ej., duración = 5 s, protocolo = TCP,

características	de paquetes	src_bytes = 300
4. Desarrollo del modelo	Entrenar un algoritmo de ML con restricciones	Random Forest entrenado con 800 registros y 10 características seleccionadas
5. Evaluación y validación	Evaluar mediante métricas de detección	Precisión = 93%, tasa de falsas alarmas (FAR) = 4% en 200 registros
6. Despliegue operativo	Integración en tiempo real con herramientas de monitorización	El modelo identifica 5 flujos sospechosos durante los primeros 60 segundos

Bucle continuo de retroalimentación: reentrenar el modelo mensualmente con datos etiquetados actualizados.

[Diapositiva 34: Cierre]

¡Gracias!

Presentado por **Adarsh Kumar**

Department of Computer Science, UPC, Dehradun

Co-funded by the European Union