

Detectores Especializados

Triple Capa de Protección

1. ML Detector

- Análisis de patrones de tráfico de red
- Modelo entrenado con datos reales

2. Behavior Detector

- Monitoreo de comportamientos sospechosos
- Análisis de actividad de procesos
- Detección de patrones anómalos

3. Network Detector



Monitores del Sistema

Vigilancia Continua 24/7



File Monitor

- Archivos creados/modificados
- Actividad de escritura sospechosa



Network Monitor

- Conexiones entrantes/salientes
- Tráfico de datos en tiempo real
- Análisis de patrones de comunicación




Process Monitor

- Procesos en ejecución

Resultados y Métricas

Rendimiento del Sistema

Sistema Operativo:

- CPU Usage: 14.0%
- RAM Usage: 58.3%
- Conexiones activas: 108 monitoreadas
- Estado:  FUNCIONANDO AL 100%

Capacidades de Detección:

- Tiempo real - Detección instantánea
- Bajo consumo - Recursos optimizados
- Alta precisión - 73 78% de aciertos

Versión de Producción

Sistema Listo para Despliegue

Executables Disponibles:

```
# Instalación simple
cd ANTIVIRUS_PRODUCTION
pip install -r requirements.txt

# Ejecución
python simple_launcher.py
```

Estructura Optimizada:

- 350MB total - Solo archivos esenciales
- Modelos ML incluidos y optimizados
- Configuración lista para usar

Tecnologías Utilizadas

Stack Tecnológico Robusto

Backend:

- Python 3.11+ - Lenguaje principal
- scikit-learn - Machine Learning
- ONNX Runtime - Optimización de modelos
- psutil - Monitoreo del sistema

Data Science:

- pandas - Manipulación de datos
- numpy - Cálculos numéricos
- joblib - Serialización de modelos

Web Dashboard (Bonus)

Interfaz de Monitoreo

Características del Dashboard:

- Visualización de logs en tiempo real
- Métricas del sistema y detecciones
- Historial de amenazas detectadas
- Configuración remota del antivirus

Despliegue:

```
# API REST disponible
cd web_api
python main.py
# Dashboard en http://localhost:8000
```

Demo en Vivo

¡Veamos el Sistema en Acción!

Proceso de Demostración:

1. Arranque del sistema anti-keylogger
2. Monitoreo en tiempo real
3. Simulación de actividad sospechosa
4. Detección y alertas
5. Logs y reportes generados

Nota: Demo realizada en entorno controlado para fines educativos



Casos de Uso Reales

Aplicaciones Prácticas



Empresas:

- Protección de datos corporativos
- Monitoreo de estaciones de trabajo
- Compliance y auditoría



Usuarios Domésticos:

- Protección de información personal
- Seguridad en banca online
- Privacidad familiar

✓ Logros Alcanzados

¿Qué hemos conseguido?

✓ Tecnológicos:

- Sistema ML funcional al 73.78% de precisión
- Arquitectura modular y escalable
- Optimización ONNX para producción
- Monitoreo multicapa integrado

✓ Prácticos:

- Executable listo para despliegue
- Configuración plug-and-play
- Dashboard web operativo

Limitaciones Actuales

Áreas de Mejora Identificadas

Técnicas:

- Precisión - Objetivo 85%+ para producción
- Falsos positivos - Refinamiento necesario
- Datasets - Más datos de entrenamiento
- Features - Análisis de comportamiento expandido

Operacionales:

- Instalación - Simplificar proceso
- Interfaz - GUI más intuitiva
- Documentación - Manual de usuario



Próximos Pasos

Roadmap de Desarrollo



Corto Plazo (1-3 meses):

- Mejorar precisión del modelo ML
- Interfaz gráfica para usuarios finales
- Sistema de actualizaciones automáticas
- Testing automatizado completo



Largo Plazo (6+ meses):

- Deep Learning con redes neuronales
- Detección de malware general
- Integración cloud y sincronización

Valor y Aplicabilidad

¿Por qué es importante este proyecto?

Académico:

- Aplicación práctica de Machine Learning
- Integración de múltiples tecnologías
- Experiencia en ciberseguridad
- Metodología de desarrollo completa

Profesional:

- Portfolio de proyecto completo
- Experiencia en ML aplicado
- Conocimiento en seguridad informática

Aspectos Técnicos Avanzados

Detalles de Implementación

Pipeline de ML:

```
Raw Network Data → Feature Extraction →  
Model Prediction → Risk Assessment →  
Action Trigger → Logging & Alerts
```

Arquitectura de Software:

- Patrón Facade - Interfaz simplificada
- Observer Pattern - Monitores en tiempo real
- Strategy Pattern - Detectores intercambiables
- Factory Pattern - Creación de componentes



Reconocimientos

Agradecimientos y Créditos



Fuentes de Inspiración:

- Papers académicos sobre detección de malware
- Datasets públicos de ciberseguridad
- Comunidad open source de Python
- Documentación técnica de ONNX y scikit-learn



Herramientas Utilizadas:

- VS Code - Desarrollo
- Git - Control de versiones
- Railway - Despliegue cloud

? Preguntas y Respuestas

¡Momento de Interacción!

Temas de Discusión:

- Aspectos técnicos del machine learning
- Implementación y arquitectura
- Aplicaciones y casos de uso
- Mejoras y evolución futura
- Experiencias durante el desarrollo

Contacto:

- GitHub: [proyecto-Anti-keylogger](#)
- Documentación: Ver carpeta `MD's Explicativos/`

 ¡Gracias!

Sistema Anti-Keylogger con ML

 Lo que hemos visto:

- Problema real de ciberseguridad
- Solución innovadora con ML
- Implementación completa y funcional
- Resultados medibles y prometedores
- Futuro prometedor del proyecto

 Mensaje Final:

"La ciberseguridad no es solo sobre proteger datos, es sobre proteger vidas digitales y preservar la confianza en la tecnología"

