**Project Sentinel:**

**Confidence-Based Malware Detection for Common File Formats**

East Carolina University

Department of Technology Systems

ICTN 6862

Andrew Bentkowski

Ashley Dickens

Submitted 7/20/2025

**Abstract**

Malicious files are a persistent threat across all digital landscapes, regardless of size and complexity. Should this malware be executed by a user with administrative permissions, untold damage can be done in moments. Project Sentinel was constructed to assist users by offering a second opinion on a file’s security through AI-based analysis. Sentinel accepts common extensions such as PDF, DOCX, XLSX, EXE, as well as MD5 hashes, with the user having to simply drag/drop a file or paste a hash value. Should there be a match via Open Source Intelligence (OSINT), AI will communicate this to the user. If not, it will conduct additional AI-driven analysis, and assign a percentage between 0 and 100% confidence the file is malicious, along with rationale for this decision. Sentinel is not a replacement for a security team, but may help users without access to one if resources are limited.

**Introduction**

In today's interconnected digital landscape, malicious files represent a significant threat to organizational security. Malicious software was present in 40% of data breaches in 2023, and remains a vital pipeline for cybercrime. Computer literacy and awareness of modern cyber threats can vary greatly throughout organizations, but it only takes one user with the right privilege set to compromise an entire network. While some organizations may have a security team that can investigate potential incidents in a timely fashion, many rely on their user base to keep their enterprise out of harm’s way. 47% of businesses with fewer than 50 employees have no cybersecurity budget (Rahmonbek 2025), meaning no full-time security hires, no advanced SaaS tools, and limited education in the workplace surrounding security threats. Project Sentinel was created for instances such as these, where access to knowledgeable security personnel is out of budget, or logistically impossible.

Project Sentinel is a comprehensive AI-driven malware detection system. It takes in files of common formats (DOCX, XLSX, PDF, EXE) and MD5 hashes for analysis. These formats were chosen as they are among the most common file types for malware payloads (Ang & Castillo, 2018). The system parses OSINT for known matches, before also conducting AI-driven pattern analysis for further analysis. The output provides a quick, educated, and readable assessment of the given file’s legitimacy and safety. Sentinel is not meant to replace a security analyst’s work, nor should it be taken as a definitive assessment, as it has not been trained on a large enough sample size at this time, and AI is still prone to error. The project’s intent is to provide actionable insight into a potential threat from a user who cannot measure the validity of a file on their own.

**Methodology**

Project Sentinel consists of 3 connected entities: A web-based frontend user interface, a flask-based API server, and an analysis engine. After completing a request, the user is provided output based on the analysis done.

The frontend is a concise webpage that allows users to drag/drop/input the file in question. The system was designed to be as readable and user-friendly as possible, stating the various files supported, how to upload them, size restrictions, and what the platform does. The results are made to be readable by technical and non-technical users alike, using color schemes to indicate risk levels alongside the numerical results (Red = Bad, Green = Good).

The backend is the orchestration layer, managing uploads, coordinating the flow of analyses, and interconnecting the frontend to the results of the analysis. It is built using the flask web framework and utilizes a REST API to handle the processing of files, the analysis queues, and the return of structured data to the frontend. It will also check for misinput and ensure secure file handling upon evaluation.

The analysis engine has a two-phased approach. It will first query for known malware via rapid hash-based lookups using the API’s connection to OSINT tools such as VirusTotal, and others. Should no results be available or it is not conclusively malicious, the next step will be to conduct AI-driven file analysis. The file will be examined for suspicious patterns, tailored to each file extension, to uncover behavior that may signal malicious intent. PDFs will look for embedded JavaScript, execution triggers, embedded files, and more. Office documents will be searched for macros, API calls, metadata, and more. Executables will check for packing, obfuscation, structural anomalies, and more.

Upon the completion of an investigation, the results will be displayed to the user in a readable, and transparent fashion thanks to its LLM capabilities. Information on what contributed to the score given will be provided in an easily accessible format, with numeric and text-based context available.

\*\*\*Insert Dataset info/Model info/Data Preprocessing Here\*\*\*

**Evaluation and Results**

\*\*\*Results of Demo Here\*\*\*

**Conclusions**

**Appendix**

Ang, M. C., & Castillo, D. (2018, October 29). *New File Types Emerge in Malware Spam Attachments*. Trend Micro.<https://www.trendmicro.com/en_us/research/18/j/same-old-yet-brand-new-new-file-types-emerge-in-malware-spam-attachments.html>

Rahmonbek, K. (2025, January 2). *35 Alarming Small Business Cybersecurity Statistics for 2025*. StrongDM.<https://www.strongdm.com/blog/small-business-cyber-security-statistics>