Module: Intelligent Agents

September 2025

Development Team Project: Project Report

Digital Forensics

The role of autonomous verification agents in reducing malware risks and ensuring compliance

Group E

Thiago Contardi, Pëllumb Dalipi, Paul Dogar

Introduction

Cybersecurity incidents demonstrate that simple actions such as downloading a file or opening a document can lead to serious organisational consequences. Malicious content is not limited to suspicious websites; it often arrives through email attachments, shared platforms, or even trusted vendor portals (Verizon, 2025). Traditional antivirus tools miss a considerable share of new malware (Acar et al., 2019), while human error remains a major weakness, with phishing and unsafe downloads among the most common entry points (Verizon, 2025). As malware complexity increases, often aided by AI, the limits of user awareness and conventional defences are increasingly clear (Wolsey, 2022). Based on this context, the report therefore asks: *What organisational advantages can be achieved by deploying an autonomous agent that pre-screens file downloads to reduce malware risks, ensure compliance, and improve operational efficiency?*

Downloaded files continue to be a significant attack vector, as seen in past and recent incidents. The "NotPetya" campaign, spread via a compromised update, caused global disruption (Greenberg, 2018), while the "Cleaner" case saw over two million infected downloads (Newman, 2018). Drive-by downloads remain common (Ibrahim et al., 2020), with recent examples like a trojanized configuration tool from "Endgame Gear" (TechRadar, 2025). Malware also increasingly hides in routine formats such as PDFs and commonly used office work files rather than executables (Acar et al., 2019). Outdated systems amplify the risk: Windows 10 support ends in October 2025, yet this operating system still runs on over 40% of desktops (Microsoft, 2025a; Crider, 2025). These conditions make a strong case for the need for autonomous agents to manage downloads before they reach end-users. As the browser is the primary gateway for downloading files and interacting with online content, it is an obvious security vector to consider. With these factors in mind, we propose developing a Chrome browser extension that intercepts and streams downloads to a server-hosted AI agent, which flags high-risk content before a download is permitted by the end-user. The choice to focus on Chrome fell due to it holding over 65% of the global market share (StatCounter, 2025). Our proposal is built on the following workflow:

System and Library Requirements

System and Library Requirements: The agent will be built in Python (≥3.11) using scikit-learn for anomaly detection, optional lightweight TensorFlow/PyTorch models, yara-python for rule-based filtering, and PostgreSQL for structured storage. The Chrome Extension relies on Manifest V3. Supporting libraries include pynacl for Ed25519 signing and cryptography for encryption. Deployment will use Docker to ensure portability.

Architecture and Components

Management Layer: Coordinates scanning, filtering, information collection, and reporting.

Scan: Detects files by type, extension, signature, and metadata. Parallel processing to increase efficiency.

Relevance Filter: Applies YARA rules to compare with known clean datasets and detect suspicious patterns.

Information Collection: Extracts metadata and forwards it to the AI model for anomaly scoring.

Packaging and Storage: Bundles files into encrypted archives, signs them using Ed25519, and securely stores them with associated metadata in PostgreSQL.

Reporting: Generates a report summarising scans, detected anomalies, and actions taken.

As illustrated in Figure 1, the diagram outlines the whole sequence of interactions, beginning with the user initiating a download and ending with the system's decision to deliver or block the file.

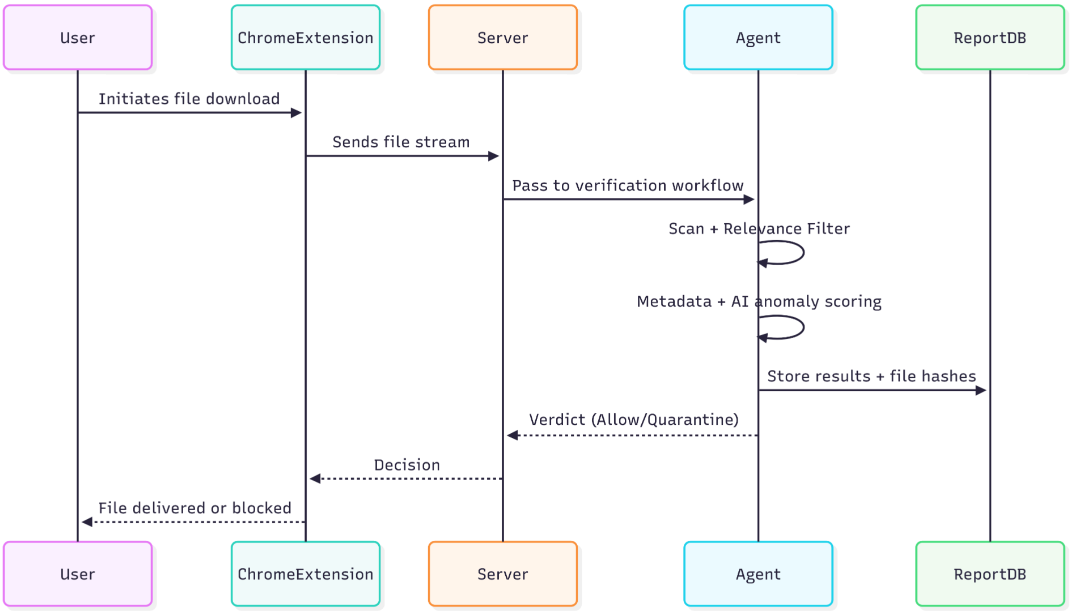

Figure 1: Sequence diagram of the download-verification workflow, illustrating interactions between the user, Chrome extension, server, agent, and reporting database, from file initiation to final verdict (allow or quarantine).

To develop the extension, we propose the following technology stack:

Python is selected as the primary language due to its mature ecosystem in digital forensics and machine learning. For classification and anomaly detection, scikit-learn is utilised, supplemented by lightweight TensorFlow or PyTorch models. A relational database is chosen for structured storage.

This stack is designed to remain lightweight and efficient, ensuring that the agent can be deployed without significant resource demands. The structural design of this stack is shown in Figure 2.

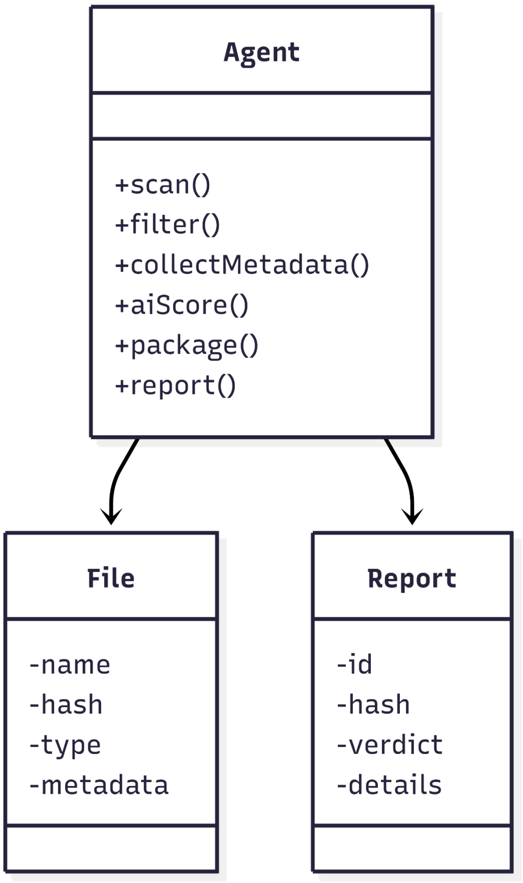

Figure 2: Class diagram of the agent, showing core methods (scan, filter, metadata collection, anomaly scoring, packaging, and reporting) and its relationships with file and report entities.

Design Decisions and Alternatives

Although the agent primarily relies on deterministic scanning, machine learning provides an additional layer of detection. One approach is PDF risk scoring, which examines embedded JavaScript, object counts, and other unusual features that may indicate malicious content (Damodaran et al., 2017). A second application is image forensics, where autoencoder-based anomaly detection can flag irregularities such as missing metadata that suggest tampering (Verdoliva, 2020). A third is document classification, where lightweight pre-trained language models such as DistilBERT can analyse extracted text to detect phishing or fraud indicators (Devlin et al., 2019). These methods were chosen for their strong balance of accuracy and ease of use. PDF risk scoring and autoencoder-based anomaly detection focus on specific types of malware, while lighter transformer models like DistilBERT (Devlin et al., 2019) allow for efficient document classification without the heavy computational demands of larger deep learning models.

Alternative approaches like secure web gateways or endpoint response systems, such as Microsoft IRE, work post-delivery, requiring remediation after exposure (Microsoft, 2025b). The proposed agent intervenes earlier, blocking threats before they reach endpoints. Python was chosen over Go or Node for its strong forensic and ML ecosystem, with PostgreSQL selected for auditability and compliance reporting.

Development Approach and Methodology

The overall decision-making process combines scanning, filtering, anomaly detection, and reporting into a final verdict (Figure 3).

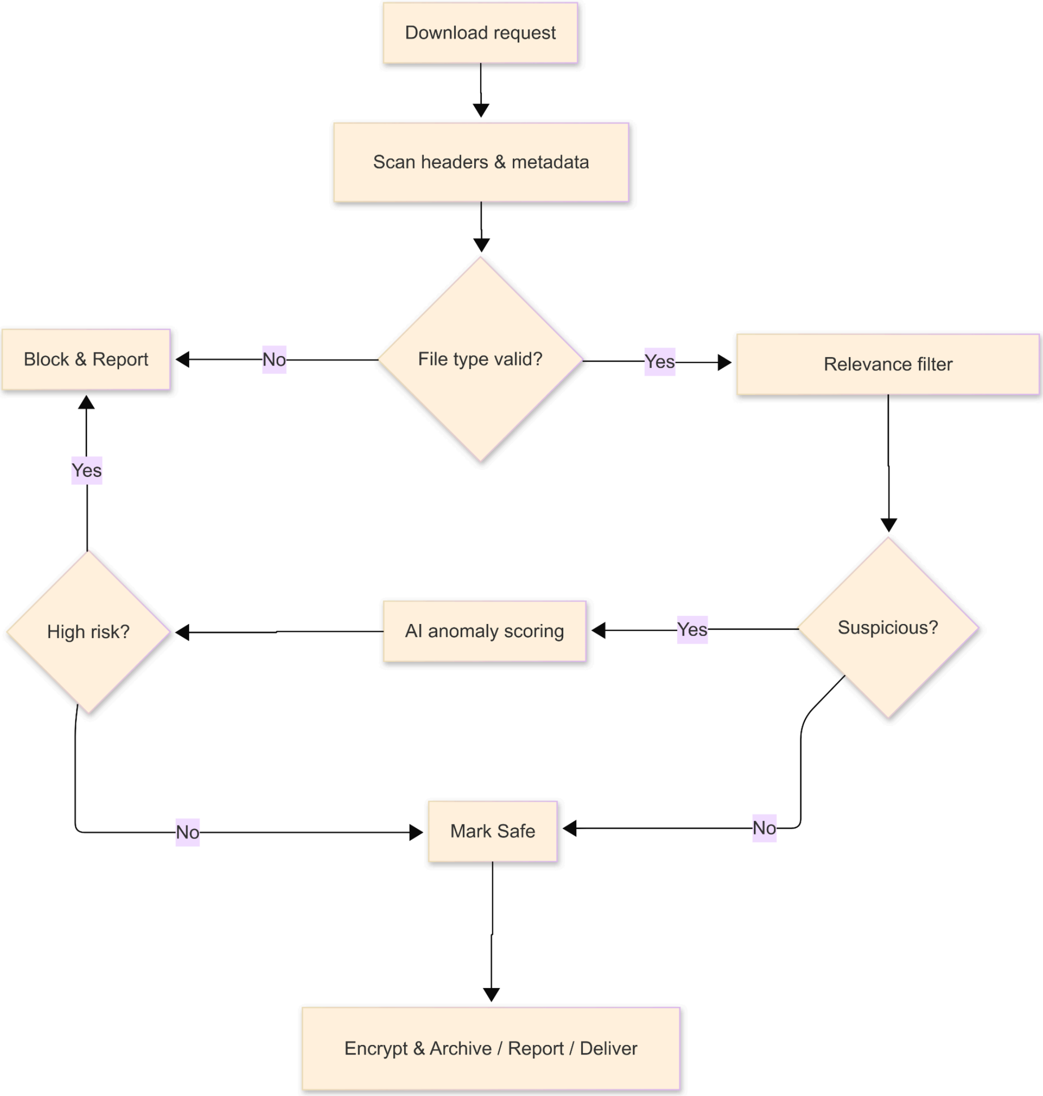

Figure 3: Activity diagram: Decision flow of the autonomous download-verification agent, from request to final disposition (block and report, or encrypt, archive, report and deliver), showing header scan, file-type validation, relevance filtering, suspicion check, AI anomaly scoring, and risk assessment.

Introducing a pre-screening agent offers clear organisational benefits. Security is strengthened by intercepting malicious files before they reach workstations, reducing the risk of network-wide compromise (Verizon, 2025). Compliance is supported through detailed logs and reports that demonstrate adherence to frameworks such as GDPR and ISO 27001 (Suorsa and Helo, 2024). Efficiency improves as analysts are freed from routine checks, and downtime during incidents is reduced; the IBM Security (2025) report estimates average breach costs at $4.45 million. The financial case is reinforced by high-profile losses such as NotPetya, which caused hundreds of millions in damage (Greenberg, 2018).

Challenges and Mitigation

Challenges remain: machine learning can produce false positives, performance must be balanced with speed, and staff may resist restrictive tools. These can be mitigated through conservative thresholds, efficient processing, transparent overrides, and clear reporting. Compared with Microsoft IRE and similar endpoint response systems, which focus on detection and response at the host level, a pre-screening agent shifts protection earlier in the chain, blocking threats before they reach the endpoint and reducing remediation overhead.

Conclusion

This proposal specifies an autonomous pre-screening agent that inspects downloads before they reach endpoints. By combining deterministic scanning with lightweight AI modules, it strengthens protection, supports audit and compliance, and reduces analyst workload. The agent complements rather than replaces endpoint solutions by moving control earlier in the chain. While no single control eliminates risk, the defined architecture, metrics, and mitigations provide a practical and measurable path to deployment.

References

Acar, A., Lu, L., Uluagac, A.S. & Kirda, E. (2019) ‘An Analysis of Malware Trends in Enterprise Networks’, *arXiv preprint* arXiv:1910.00508. doi:10.48550/arXiv.1910.00508. Available at: https://arxiv.org/abs/1910.00508 (Accessed: 31 August 2025).

Crider, M. (2025) *Windows 11 overtakes Windows 10 in users, just in time*. PCWorld, 7 July 2025. Available at: https://www.pcworld.com/article/2839068/windows-11-overtakes-windows-10-in-users-just-in-time.html (Accessed: 4 September 2025).

Damodaran, A., Di Troia, F., Visaggio, C.A., Austin, T.H. & Stamp, M. (2017) ‘A comparison of static, dynamic, and hybrid analysis for malware detection’, *Journal of Computer Virology and Hacking Techniques*. Available at: https://arxiv.org/abs/2203.09938 (Accessed: 3 September 2025).

Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. (2019) ‘BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding’, in *Proceedings of NAACL-HLT 2019*. Minneapolis, MN: ACL. Available at: https://aclanthology.org/N19-1423/ (Accessed: 3 September 2025).

Greenberg, A. (2018) ‘The Untold Story of NotPetya, the Most Devastating Cyberattack in History’, *WIRED*, 22 August. Available at: https://www.wired.com/story/notpetya-cyberattack-ukraine-russia-code-crashed-the-world/ (Accessed: 3 September 2025).

IBM Security (2025) *Cost of a Data Breach Report 2025*. IBM Security. Available at: https://www.ibm.com/reports/data-breach (Accessed: 3 September 2025).

Ibrahim, S., Al Herami, N., Al Naqbi, E. & Aldwairi, M. (2020) ‘Detection and Analysis of Drive-by Downloads and Malicious Websites’, *arXiv preprint* arXiv:2002.08007.

doi:10.48550/arXiv.2002.08007. Available at: https://arxiv.org/abs/2002.08007 (Accessed: 3 September 2025).

Microsoft (2025a) *End of support for Windows 10, Windows 8.1, and Windows 7*. Available at: https://www.microsoft.com/en-us/windows/end-of-support (Accessed: 3 September 2025).

Microsoft (2025b) *Project IRe autonomously identifies malware at scale*. Microsoft Research Blog. Available at: https://www.microsoft.com/en-us/research/blog/project-ire-autonomously-identifies-malware-at-scale/ (Accessed: 6 September 2025).

Newman, L.H. (2018) ‘Inside the Unnerving Supply Chain Attack That Corrupted CCleaner’, *WIRED*, 17 April. Available at: https://www.wired.com/story/inside-the-unnerving-supply-chain-attack-that-corrupted-ccleaner/ (Accessed: 3 September 2025).

StatCounter. (2025). *Browser market share worldwide*. StatCounter Global Stats. Available at: https://gs.statcounter.com. (Accessed: 07 September 2025).

Suorsa, M. and Helo, P. (2024) ‘Information security failures identified and measured – ISO/IEC 27001:2013 controls ranked based on GDPR penalty case analysis’, *Information Security Journal: A Global Perspective*. doi:10.1080/19393555.2023.2270984.

TechRadar (2025) ‘Endgame Gear warns mouse config tool has been infected with malware’, *TechRadar Pro*, 29 July. Available at: https://www.techradar.com/pro/security/endgame-gear-warns-mouse-config-tool-has-been-infected-with-malware (Accessed: 3 September 2025).

Verdoliva, L. (2020) ‘Media Forensics and DeepFakes: An Overview’, *IEEE Journal of Selected Topics in Signal Processing*.

doi:10.1109/JSTSP.2020.3002101. Available at https://ieeexplore.ieee.org/document/9115874 (Accessed: 3 September 2025).

Verizon (2025) *2025 Data Breach Investigations Report*. Verizon Enterprise Solutions. Available at: https://www.verizon.com/business/resources/Te7d/reports/2025-dbir-data-breach-investigations-report.pdf (Accessed: 31 August 2025).

Wolsey A. (2022). *The State-of-the-Art in AI-Based Malware Detection Techniques: A Review*. https://arxiv.org/abs/2210.11239. (Accessed 06 September 2025).