Check for updates

RESEARCH ARTICLE

REVISED PS-SecureScan: Pre-Execution Security Analysis for Photoshop Image Workflows

[version 2; peer review: 1 approved, 1 approved with reservations]

Karol Kozak, Weronika Kozak

University of Lower Silesia, Wrocław, Lower Silesian Voivodeship, Poland

v2 **First published:** 13 May 2026, **6**:166
https://doi.org/10.12688/openreseurope.23567.1
Latest published: 09 Jul 2026, **6**:166
https://doi.org/10.12688/openreseurope.23567.2

Abstract

Image files are commonly treated as passive digital assets; however, modern formats such as PSD, TIFF, JPEG and PNG are complex structured containers that require extensive parsing, decompression and metadata interpretation. In creative production environments Adobe Photoshop is one of the most widely used tools for processing such files, making it a potential entry point for image-based cyberattacks. Vulnerabilities in image parsing routines, metadata handling and decompression mechanisms have historically enabled remote code execution, denial-of-service attacks and covert data exfiltration. This paper presents a proof-of-concept framework for pre-execution security analysis of image files used in Photoshop-centered workflows and introduces PS-SecureScan, a pre-execution image examination framework designed to detect malicious or anomalous image files before they are opened in production software.

The framework integrates structural validation, metadata anomaly detection, entropy-based steganography analysis, decompression risk estimation and sandboxed behavioral observation. Experimental evaluation using synthetic benign and adversarial image samples demonstrates that the proposed multi-layer analysis approach enables effective differentiation between normal creative assets and manipulated images. The results suggest that pre-execution image examination can significantly reduce the risk of image-based compromise in digital media workflows. The current implementation should be viewed as an exploratory prototype intended to evaluate the feasibility of multi-layer image security analysis rather than a fully operational production security platform.

Open Peer Review

Approval Status ✓ ?

	1	2
version 2 (revision) 09 Jul 2026	✓ view	? view
version 1 13 May 2026	? view	? view

1. **Alaa Jabbar Qasim Almaliki**, Universiti Utara Malaysia (UUM), Kedah Darul Aman, Malaysia

2. **Oleksandr Kuznetsov**, eCampus University, Novedrate, Italy

Any reports and responses or comments on the article can be found at the end of the article.

Keywords
photoshop, cybersecurity, adversarial attack, art, media, design.

This article is included in the Horizon Europe gateway.

This article is included in the Digital Security collection.

Corresponding author: Karol Kozak (karol.kozak@dsw.edu.pl)

Author roles: Kozak K: Conceptualization, Data Curation, Formal Analysis, Investigation, Methodology, Resources, Software, Supervision, Validation, Visualization, Writing – Original Draft Preparation; **Kozak W**: Conceptualization, Data Curation, Formal Analysis, Resources, Validation, Visualization, Writing – Original Draft Preparation, Writing – Review & Editing

Competing interests: No competing interests were disclosed.

Grant information: This research was partially supported in frame of scientific support to project partner by the European Union's Horizon Europe programe under the project HORIZON-CL4-2022-TWIN-TRANSITION-01-01 . This study received partial funding from Horizon Europe Framework Programme -101092073. The authors did not receive any additional specific grants from funding agencies in the public, commercial or not-for-profit sectors.

Copyright: © 2026 Kozak K and Kozak W. This is an open access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

How to cite this article: Kozak K and Kozak W. **PS-SecureScan: Pre-Execution Security Analysis for Photoshop Image Workflows [version 2; peer review: 1 approved, 1 approved with reservations]** Open Research Europe 2026, **6**:166 https://doi.org/10.12688/openreseurope.23567.2

First published: 13 May 2026, **6**:166 https://doi.org/10.12688/openreseurope.23567.1

REVISED **Amendments from Version 1**

We thank the reviewers for their constructive feedback. The manuscript has been substantially revised and clarified. The current version positions PS-SecureScan as a proof-of-concept framework, improves transparency of the experimental evaluation with additional quantitative metrics, and expands comparisons with existing tools. We also refined descriptions of all components (including sandbox, metadata, steganography, and PSD analysis) to accurately reflect the current implementation.

Additionally, we introduced dedicated Threat Model, Limitations, and Future Work sections, updated the literature review with recent work, and clearly outlined directions for further development and real-world validation.

These revisions significantly strengthen the clarity, rigor, and reliability of the paper.

Any further responses from the reviewers can be found at the end of the article

Introduction

Digital image processing is central to creative industries including advertising, branding, publishing, and digital content production. Adobe Photoshop remains one of the most widely adopted tools for professional image manipulation and layered asset composition. Its support for complex file formats, embedded metadata, smart objects, and high-resolution assets increases flexibility but also expands the attack surface.

Historical vulnerabilities such as ImageTragick (CVE-2016-3714) demonstrated that crafted image files can execute arbitrary commands in automated processing systems.[1] Similar vulnerabilities have been identified in widely used image libraries.[2,3] Research has shown that format complexity and backward compatibility increase parser error probability.[4]

Steganographic techniques allow hidden payload embedding in image data,[5] while decompression bombs can exhaust memory resources.[6] Adversarial image manipulation further introduces risks in AI-assisted workflows.[7]

Image parsing vulnerabilities

Crafted image files exploiting buffer overflows, integer overflows and improper memory handling have repeatedly enabled arbitrary code execution in image processing software. Image parsing libraries are often written in low-level languages and must process complex binary structures, which increases the probability of implementation flaws.

Formats such as TIFF and PSD present particularly large attack surfaces due to nested structures, layered metadata and support for multiple compression schemes. TIFF allows chained image file directories, embedded thumbnails and custom tags, while PSD contains layer hierarchies, masks and adjustment data that require recursive parsing. These features increase the complexity of validation and error handling.

Figures 1 and 2 illustrate a simplified structure of a Photoshop PSD file and highlights potential security vulnerabilities that may exist within its layers and components. On the left side, the hierarchical organization of the PSD document is shown, including background layers, text layers, images, effects, adjustment layers, and older versions. On the right side, several potential risk points are indicated. These include hidden layers that may contain sensitive information, text layers that may store embedded metadata or scripts, and external links to referenced files that may introduce integrity or security risks. The diagram emphasizes how design files can unintentionally contain hidden data or external dependencies that may lead to information leakage or manipulation if not properly managed.

The Figure 2 presents a detailed PSD file structure and highlights potential unwanted vulnerabilities within design assets, layers, and exported files. It shows how elements such as linked files, hidden layers, embedded smart objects, fonts, large images, and exported documents may introduce risks like hidden data exposure, metadata leakage, or unintended design changes. The diagram also categorizes these risks by severity (high, medium, low) and includes recommended practices for reducing vulnerabilities when sharing or exporting PSD files.

Historically, vulnerabilities in image parsing libraries such as libtiff, ImageMagick and graphic toolkits have demonstrated that malformed headers, incorrect size fields or corrupted chunk offsets can trigger memory corruption.[8] Attackers often exploit inconsistencies between declared structure sizes and actual data lengths.

Another important class of vulnerabilities involves insufficient validation of image metadata, including EXIF and color profile information. Embedded ICC profiles and metadata blocks may contain oversized or malformed fields that lead to

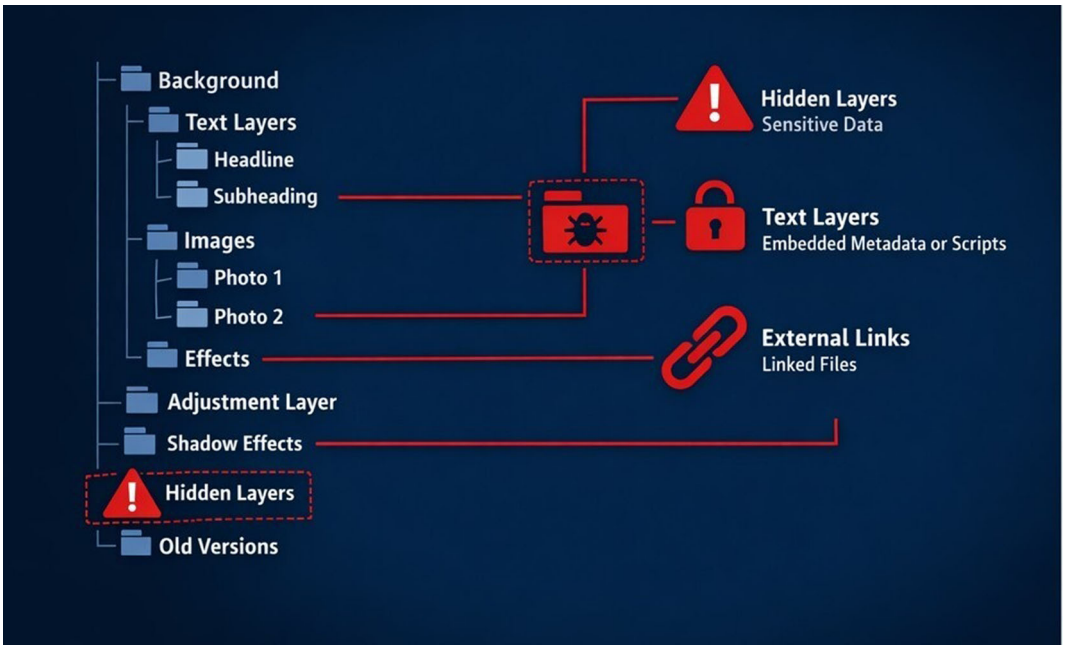

Figure 1. PSD File Structure and Potential Vulnerability Points.

parser failures. In creative workflows where assets are exchanged between different tools, such malformed files can propagate across systems before detection.

Steganography and covert channels

Steganography enables hidden communication by embedding information inside seemingly harmless image data. Classical approaches rely on least significant bit embedding in pixel values, which introduces minimal visual distortion and is difficult to detect by human observers.

More advanced techniques operate in the frequency domain, modifying coefficients in transform-based representations such as discrete cosine transform (DCT) used in JPEG compression. These approaches increase robustness against recompression and filtering operations that typically occur in digital media workflows.

In modern environments steganography can also be used for covert command and control channels or data exfiltration.[9,10] Images uploaded to public platforms may carry hidden payloads that are later extracted by automated scripts or malware.

Detection methods rely on statistical analysis of pixel distributions, entropy estimation and identification of deviations from natural image statistics. Machine learning approaches have recently improved detection accuracy by learning complex statistical patterns associated with hidden payloads. Nevertheless, adversarial techniques continue to evolve and attempt to minimize detectable artifacts.[5,11]

Decompression and resource exhaustion attacks

Compression-based denial-of-service attacks exploit extremely high expansion ratios during decompression. A small compressed file may expand into gigabytes of data when processed by an image decoder, exhausting memory or storage resources.

Image formats frequently support multiple compression algorithms, including LZW, ZIP, RLE and JPEG compression. When combined with large image dimensions or deeply nested structures, decompression operations may lead to excessive CPU usage or memory allocation failures.

High-resolution production pipelines in photography, design and visual effects increase susceptibility to such attacks. Large images are common and therefore oversized files do not immediately raise suspicion. Attackers can craft files that appear legitimate while triggering expensive decompression routines in background processing systems or asset management platforms. Mitigation strategies include strict validation of declared image dimensions, limits on

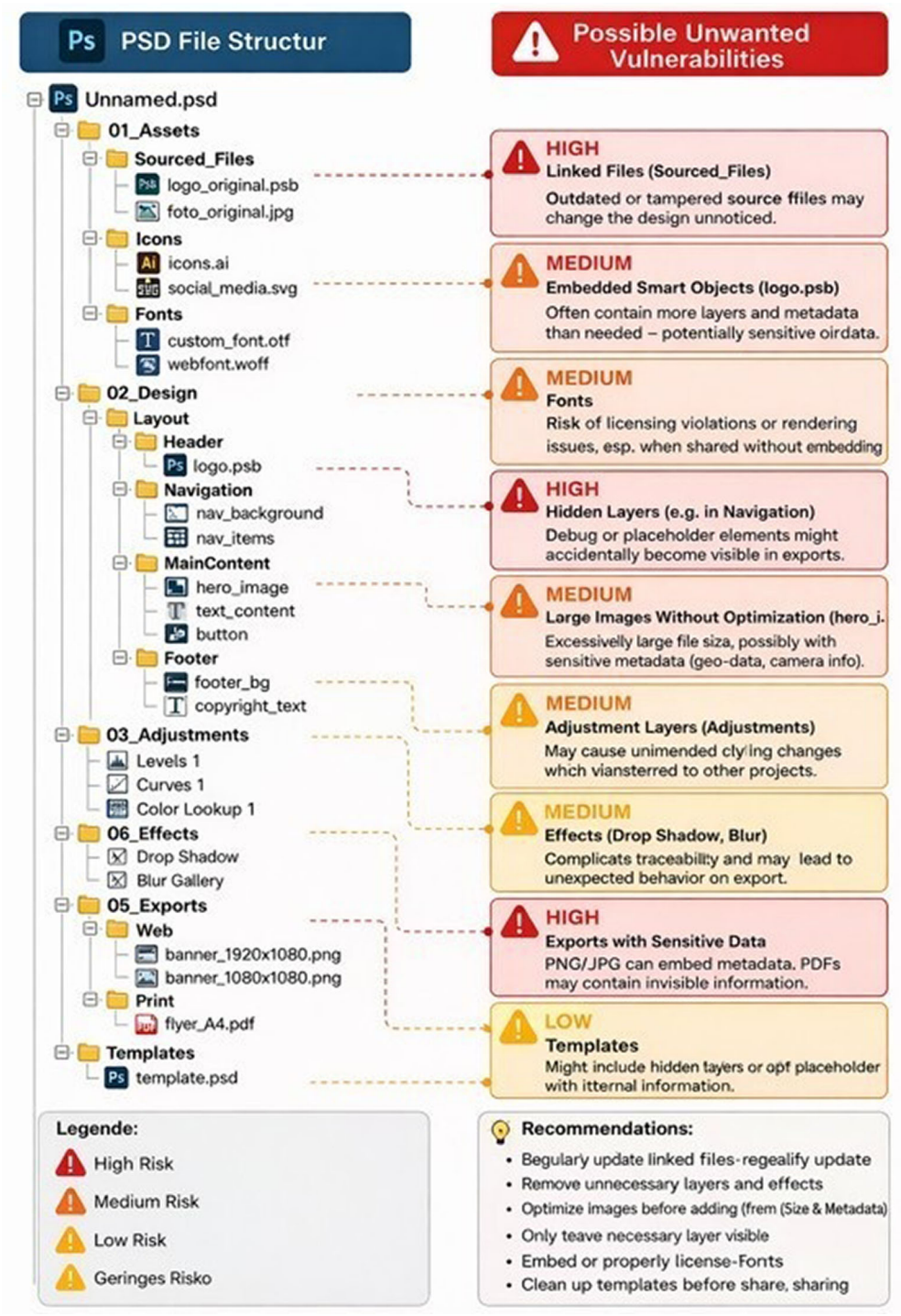

Figure 2. Simplified PSD document hierarchy illustrating common risk points such as hidden layers, embedded metadata, and external file links.

decompressed data size and resource-aware decoding mechanisms. Modern libraries increasingly implement sandboxing and incremental decoding in order to prevent uncontrolled memory allocation and processing overhead.[6,12]

Attack surface in PSD files

The Photoshop Document (PSD) format illustrates how complex image structures can expand the potential attack surface of creative media files. A PSD file is composed of several hierarchical sections, including the file header, color mode data, image resources, layer and mask information, and the final image data block. Each of these sections may contain nested structures with variable lengths and internal references.

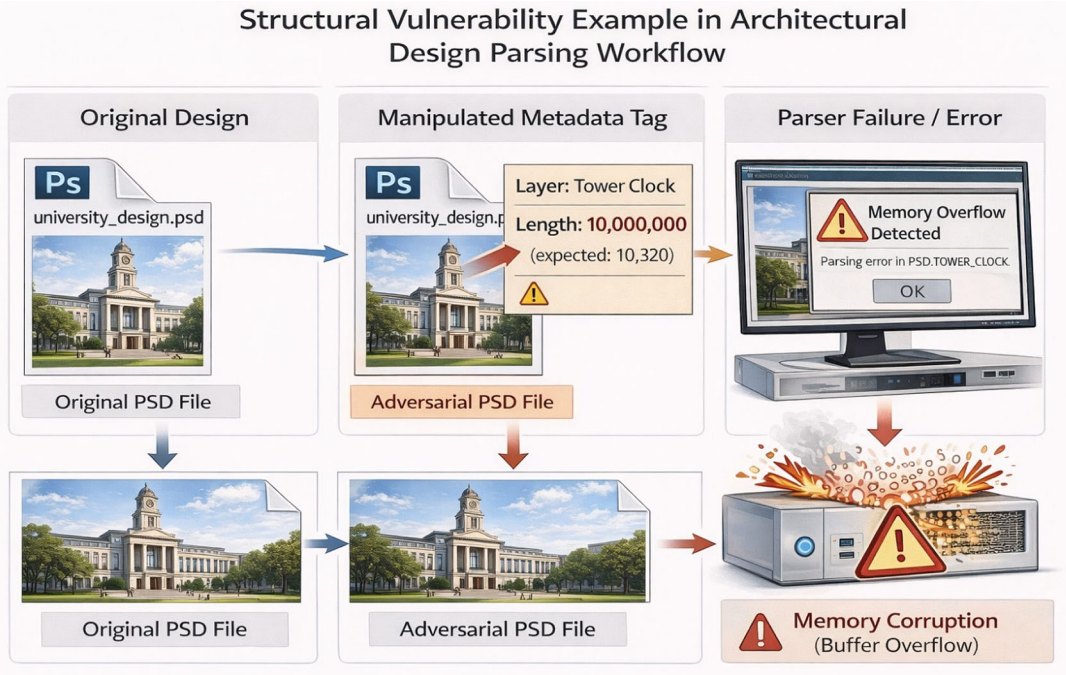

Figure 3. Example of a Structural Vulnerability in a PSD Parsing Workflow.

For example, the *layer and mask information* section stores a hierarchy of layers, where each layer may contain multiple channels, masks, blending ranges, and additional adjustment information. During parsing, the application must recursively interpret these structures while following offsets and length declarations within the file. If a malicious file manipulates these length fields or layer offsets, the parser may attempt to read beyond the intended memory boundaries.

A practical adversarial example involves a PSD file with intentionally corrupted layer metadata (Figure 3). The attacker sets an incorrect layer record length or a manipulated mask offset, causing the parser to interpret subsequent bytes as valid layer structures. In poorly validated implementations this may lead to buffer overflows, memory corruption, or excessive memory allocation. Because creative workflows frequently exchange PSD files between collaborators, such crafted files can propagate through production pipelines before detection. This example demonstrates how the structural richness of PSD files, while essential for non-destructive editing and complex compositions, also increases the complexity of secure parsing and validation.

The contributions of this work are:

Identification of cybersecurity risks associated with Photoshop-centered image workflows. Design of a modular proof-of-concept framework for pre-execution image security analysis. Development of a lightweight prototype integrating structural inspection, metadata analysis, entropy analysis and decompression risk estimation. Experimental validation using synthetic adversarial image datasets. Release of source code and reproducible datasets supporting future research.

The proposed framework assumes an adversary capable of supplying image files to a creative production workflow. The attacker does not control the target system directly but may influence files processed by image editing software, asset management systems, or automated media pipelines. The primary attack objectives considered in this study include:

- Exploitation of image parsing vulnerabilities through malformed file structures.
- Delivery of hidden payloads through steganographic embedding.
- Metadata manipulation intended to evade inspection or facilitate covert communication.
- Resource exhaustion attacks exploiting excessive decompression requirements.

Table 1. Comparison of PS-SecureScan with existing image analysis tools.

System	Metadata anomaly detection	Steganography detection	Parser anomaly detection	Workflow integration
ExifTool	Yes	No	No	No
StegExpose	No	Yes	No	No
Antivirus scanners	Partial	Partial	Partial	Limited
PS-SecureScan	Yes	Yes	Yes	Yes

Table 2. Synthetic Datasets samples.

Category	Number of images
Benign creative assets	40
Metadata manipulation	20
Steganographic payload images	20
Decompression bomb images	20
Structurally corrupted PSD files	20

The objective of PS-SecureScan is not to replace endpoint protection systems but to provide an additional pre-execution security layer capable of identifying suspicious image files before they are processed by production software.

Table 1 compares the proposed PS-SecureScan framework with commonly used image analysis and security tools. The comparison highlights the ability of PS-SecureScan to combine structural validation, metadata inspection, steganographic analysis and parser anomaly detection within a single workflow-oriented framework.

Method: PS-SecureScan framework

Incoming image files are analyzed by multiple modules including structural validation, metadata inspection, steganographic analysis and decompression risk estimation (Figure 4). The resulting anomaly indicators are combined in a risk scoring model that determines whether the file can be safely opened in the production workflow.

Design objectives

PS-SecureScan is designed as a pre-execution security layer positioned between external image sources and Adobe Photoshop production systems. Let F denote an incoming image file. The framework applies a sequence of analytical operators

$$A = \{A_struct, A_meta, A_steg, A_decomp, A_sandbox\}$$

that extract anomaly indicators from the file before it is opened by the production environment.

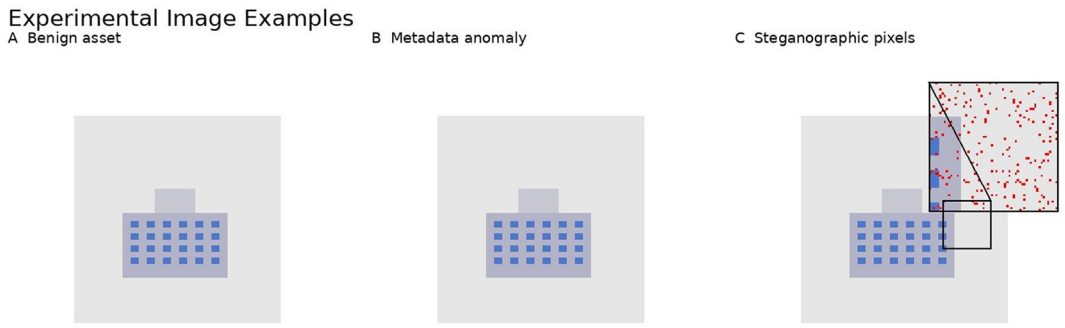

Figure 4. Examples of experimental images used to evaluate PS-SecureScan. (A) Benign creative asset without anomalies. (B) Image with enlarged metadata block (e.g., oversized XMP field) while pixel content remains unchanged. (C) Image with simulated steganographic modifications; the zoom inset highlights altered pixels in red. These variations demonstrate how different manipulation types trigger specific detection modules.

Each operator produces a normalized score S_i in [0,1] representing the likelihood of malicious manipulation or structural inconsistency. The objective is to detect parser-level vulnerabilities, covert data channels and computational resource attacks while maintaining minimal impact on legitimate workflows.

The framework therefore performs four primary tasks: structural validation, metadata inspection, statistical pixel analysis and runtime observation.

Structural validation module

The structural validation module analyzes the internal file structure

$$A_struct(F) -> S_struct$$

For PSD files the module verifies header signatures, version identifiers, declared dimensions, channel counts and layer offsets. Cross-reference validation ensures that declared block sizes match actual byte boundaries.

Let O_i denote offsets of internal structures and L_i their declared lengths. Structural consistency is verified using

$$O_i + L_i \leq |F|$$

Violations increase the anomaly score. Additional penalties are assigned for invalid marker sequences, corrupted layer records or recursive structural inconsistencies. The resulting structural anomaly score S_struct reflects the probability of malformed parsing structures.

Software description

PS-SecureScan was implemented as a deterministic Python module operating in a pre-execution scan mode without full decoding of image files. The input is a file F (PSD or raster format), and the output is a structured report containing five risk components $S_i \in [0,1]$ and a total risk score R_total. The current implementation intentionally favors lightweight heuristic analysis over deep file interpretation. The prototype therefore performs partial format inspection and should not be considered a complete parser for PSD or related image formats. The analytical modules correspond to the methodological components described in the framework:

(i) structural validation of the file format (PSD: header fields, section lengths, offset consistency; raster formats: signatures and marker integrity),

(ii) metadata heuristics detecting oversized EXIF/XMP blocks and encoding anomalies,

(iii) entropy-based estimation of steganographic likelihood using prefix data entropy,

(iv) decompression risk estimation by computing an approximate memory footprint $M_est = W \cdot H \cdot C \cdot B$, and

(v) an optional sandbox component implemented as a user-defined execution hook returning S_sandbox.

The final score is computed as a weighted linear combination of module outputs:

$$R_total = \Sigma\,(w_k \cdot S_k), \text{where}\, \Sigma\, w_k = 1$$

The resulting score is mapped to four risk classes (Low, Moderate, High, Critical) using threshold parameters $\tau 1$, $\tau 2$, $\tau 3$.

Metadata analysis module

The metadata module extracts EXIF, IPTC and XMP fields

$$A_meta(F) -> S_meta$$

Let M_j denote individual metadata fields with size size(M_j). Oversized or malformed fields are detected using threshold validation

$$size(M_j) > T_meta$$

Additional anomalies are generated for invalid encodings, malformed XML structures or unknown custom tags. Semantic checks compare timestamps, device parameters and color profile declarations. The aggregated anomaly measure forms the metadata score S_meta.

Entropy and steganography analysis module

Hidden payloads often modify pixel statistics. The steganographic analysis module computes channel-wise entropy

$$H_c = -\Sigma p_i \log 2\, p_i$$

where p_i denotes the probability of intensity value i in color channel c.

Deviation from expected entropy ranges for natural images produces anomaly signals. Additional frequency-domain analysis evaluates coefficient distributions for compressed formats such as JPEG.

The steganographic suspicion score is defined as

$$S_steg = f(|H_c - H_ref|, D_freq)$$

where H_ref represents reference entropy and D_freq denotes frequency-domain deviation metrics.

The present implementation relies on entropy-based indicators and does not yet implement advanced machine learning steganalysis methods or deep frequency-domain feature extraction.

Decompression risk estimation module

Compression-based attacks exploit extreme expansion ratios during decoding. The decompression module estimates the required memory footprint

$$M_est = W \times H \times C \times B$$

where W,H are image dimensions, C is the number of channels and B is the bit depth.

For layered formats the cumulative requirement is

$$M_total = \Sigma M_k$$

If

$$M_total > T_mem$$

a decompression risk signal is generated. The normalized result forms the score S_decomp.

Sandboxed behavioral observation module

Files exceeding a preliminary risk threshold are executed within an isolated environment[13]

$$A_sandbox(F) -> S_sandbox$$

The current prototype provides an integration interface for external sandbox environments. Rather than implementing a complete behavioral monitoring subsystem, the framework exposes a configurable execution hook that allows integration with existing sandbox platforms. Consequently, the sandbox component should be interpreted as an architectural extension point intended for future development. Runtime metrics include memory allocation m(t), CPU load c(t), system call frequency and network activity. Behavioral anomalies are detected when

$$m(t) > T_m \text{ or } c(t) > T_c$$

or when unauthorized network operations are observed. These indicators are aggregated into the sandbox anomaly score S_sandbox.

Risk scoring model

All module outputs are combined using a weighted scoring model

$$R_total = w1 \cdot S_struct + w2 \cdot S_meta + w3 \cdot S_steg + w4 \cdot S_decomp + w5 \cdot S_sandbox$$

where w_i are calibration weights satisfying

$$\Sigma w_i = 1$$

Risk classification is defined as.

Low risk: $R_total < \tau 1$.

Moderate risk: $\tau 1 \leq R_total < \tau 2$.

High risk: $\tau 2 \leq R_total < \tau 3$.

Critical risk: $R_total \geq \tau 3$.

This model allows automated security decisions in creative production pipelines while maintaining quantitative interpretability of detected anomalies.

Results

To evaluate the behavior of the proposed PS-SecureScan framework, a dataset consisting of **120 image samples** was generated. The dataset includes both benign creative assets and adversarially crafted files representing different manipulation classes.

For each analyzed image file *F*, the framework computes anomaly scores for the modules S_{struct}, S_{meta}, S_{steg} and S_{decomp}. These values are combined into the overall risk score R_{total}.

The dataset was structured as follows:

Detection performance was evaluated using the following metrics:

- **True Positive Rate (TPR)** – proportion of adversarial images correctly classified as risky
- **False Positive Rate (FPR)** – proportion of benign images incorrectly classified as risky
- **Precision** – fraction of detected risky files that are truly adversarial
- **Recall** – fraction of adversarial files successfully detected

Risk classification thresholds $\tau 1$, $\tau 2$, $\tau 3$ were selected empirically to minimize false positives while maintaining high detection sensitivity.

The reported values in Table 3 should be interpreted as preliminary performance estimates obtained on synthetic datasets. Their primary purpose is to demonstrate the feasibility of the proposed framework rather than establish definitive operational performance.

Table 3. Detection performance.

Metric	Value
Accuracy	0.91
Precision	0.88
Recall	0.93
FPR	0.04

Benign images were generated as typical design assets resembling Photoshop workflow outputs. Adversarial samples were synthetically constructed to simulate realistic attack scenarios including metadata anomalies, hidden payload embedding, and extreme decompression expansion ratios. All images were analyzed using the PS-SecureScan framework and anomaly scores were recorded for each detection module.

The dataset contained standard Photoshop design files, images with manipulated metadata fields, steganographic images containing hidden payloads, decompression bomb images, and structurally corrupted PSD files. For each input file F, the system computed anomaly scores for the individual detection modules S_struct, S_meta, S_steg, and S_decomp, together with the aggregated risk score R_total.

The results demonstrate that the module-based scoring approach clearly differentiates between benign and adversarial image inputs. Normal Photoshop assets exhibit consistently low anomaly scores across all modules, with values typically below 0.1. In contrast, manipulated image files trigger significantly higher scores in the corresponding detection modules. For instance, images with manipulated metadata fields produce elevated S_meta values, while steganographic images show increased entropy deviation reflected in S_steg. Similarly, decompression bomb images result in high S_decomp values due to extreme expansion ratios between compressed and estimated decompressed sizes.

The aggregated score R_total, computed as a weighted combination of module outputs, provides a robust indicator of overall file risk (Figure 5). Experimental observations show a clear separation between benign assets and adversarial samples, indicating that the multi-layer analysis approach effectively captures different classes of image-based threats. These results suggest that PS-SecureScan can serve as a practical pre-execution security mechanism for Photoshop-centered production environments.

Benign creative assets produced low anomaly scores across all detection modules. Most benign samples generated R_total < 0.1.

Adversarial samples triggered elevated module scores depending on manipulation type. Metadata manipulation increased S_meta, steganographic perturbations increased S_steg and decompression bombs increased S_decomp.

In contrast, adversarial samples triggered elevated module scores depending on manipulation type. Metadata manipulation images produced high SmetaS_{meta}Smeta values due to abnormal metadata field sizes. Steganographic images

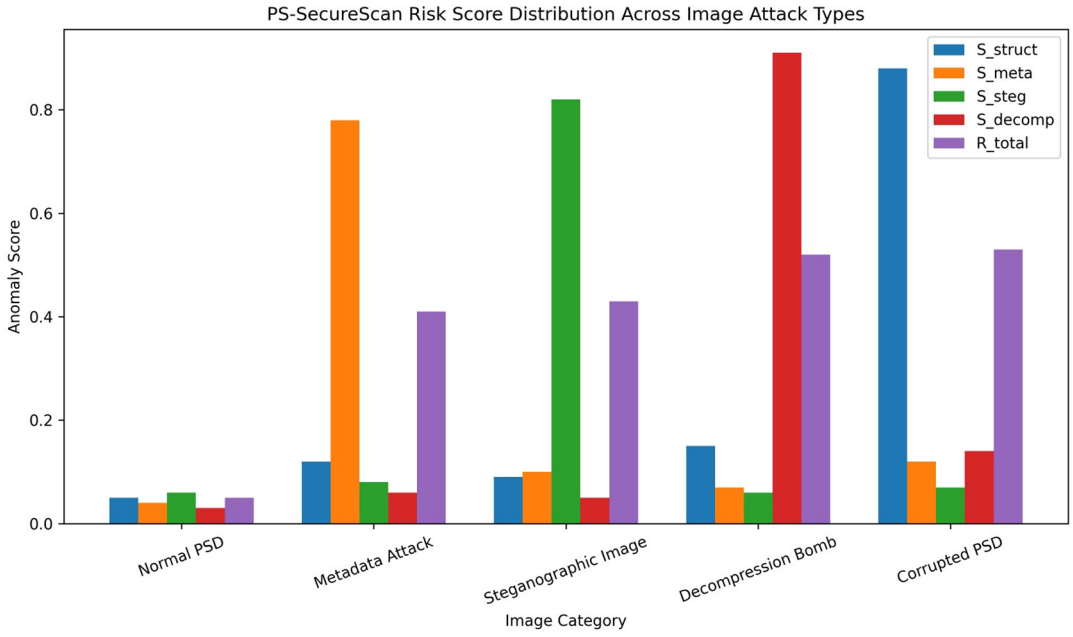

Figure 5. Distribution of anomaly scores generated by the PS-SecureScan framework for different categories of image files. Individual bars represent module outputs for structural validation (S_struct), metadata analysis (S_meta), steganography detection (S_steg), and decompression risk estimation (S_decomp). The aggregated risk score R_total demonstrates clear separation between benign Photoshop assets and adversarially crafted image files.

exhibited entropy deviations detected by the steganographic analysis module. Decompression bomb images generated high SdecompS_{decomp}Sdecomp values due to extreme estimated memory requirements.

These results demonstrate that the modular architecture enables targeted detection of different classes of image-based threats while maintaining low anomaly scores for legitimate creative assets.

Limitations

Several limitations should be acknowledged.

First, the evaluation relies exclusively on synthetic datasets. Although synthetic adversarial images allow controlled experimentation, they cannot fully reproduce the diversity of real-world malicious image files.

Second, several analytical components currently employ lightweight heuristic methods rather than deep semantic analysis. This applies particularly to metadata inspection, steganographic detection and PSD structure analysis. Third, the sandbox component currently serves as an integration interface rather than a fully implemented behavioral monitoring environment. Finally, scoring thresholds and module weights were selected empirically and have not yet been calibrated using large-scale operational datasets.

Future work

Future development of PS-SecureScan will focus on four primary directions. First, evaluation will be extended to larger collections of real-world malicious image samples and public multimedia security datasets. Second, support for deeper PSD parsing will be introduced, including recursive analysis of layer hierarchies, masks and smart objects. Third, machine learning-based steganalysis techniques will be incorporated to improve hidden payload detection.[14,15] Finally, the sandbox component will be extended into a complete behavioral monitoring environment capable of collecting runtime telemetry and anomaly indicators.

Conclusion

Creative media workflows represent an emerging cybersecurity domain in which image files function not only as visual assets but also as complex structured containers capable of carrying malicious payloads. Formats such as PSD, TIFF and high-resolution raster images contain nested structures, embedded metadata and multiple compression mechanisms that increase both functionality and potential attack surface.

This paper introduced PS-SecureScan, a pre-execution image security framework designed to detect anomalous or potentially malicious files before they are opened in production software such as Adobe Photoshop. The framework combines structural validation, metadata inspection, entropy-based analysis and decompression risk estimation within a unified risk scoring model.

Experimental results using synthetic adversarial image samples demonstrate that the proposed approach can effectively differentiate between benign creative assets and manipulated files representing several attack classes. Future work will focus on expanding format-aware analysis, integrating machine learning–based steganography detection and evaluating the framework on large-scale real-world datasets.

Software availability

PS-SecureScan: Software, Description and Usage Guidelines:

- Source code available from: https://github.com/karkoz12/ps_secure_scan
- Archived software available from: https://zenodo.org/records/19066338
- License: Apache 2.0 license.

The prototype implementation of PS-SecureScan was developed in Python and operates as a pre-execution scanning tool that analyzes image files without performing full decoding. The software evaluates structural integrity, metadata properties, entropy-based statistics and decompression risk indicators and produces a structured anomaly report.

The experimental evaluation presented in this study uses synthetic datasets containing benign images and adversarially modified examples including metadata anomalies and steganographic perturbations.

Example usage:

```
scanner = PSSecureScan()report = scanner.scan("input_image.png")print (report.
R_total, report.risk_class)
```

The software generates normalized anomaly scores for each detection module and produces a final risk score based on a weighted aggregation model.

Ethics and consent

This study used synthetically generated image datasets representing both benign creative assets and adversarially manipulated files. No human participants, clinical data or personally identifiable information were involved at any stage of the research. Therefore, ethical approval and informed consent were not required.

Author contributions

W.K. conceived the study and designed the PS-SecureScan framework.

K.K. implemented the system, generated the experimental dataset and conducted the evaluation. Both authors contributed to the analysis and interpretation of the results and approved the final version of the manuscript.

Data availability

The experimental dataset used in this study consists of synthetically generated image samples representing benign creative assets and adversarially manipulated files. The repository contains the Python implementations, and scripts to reproduce all experiments.

Underlying data

Repository name: PS_securescan synthetic dataset.

The project contains the following underlying data:

https://zenodo.org/records/19066144.[16]

The data is released under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License (CC BY-NC-SA 4.0).

References

1. **ImageTragick Vulnerability Report.** 2016.
2. **MITRE CVE Database – Image Parsing Vulnerabilities.**
3. Hoglund G, Butler J: *Rootkits: Subverting the Windows Kernel*. Addison-Wesley; 2005.
4. Bratus S, *et al.*: **Exploit Programming and Parser Exploitation.** vol. 36, no. 6, 2011.
5. Fridrich J: *Steganography in Digital Media.* Cambridge University Press; 2009.
6. Klein A: *Research on Compression Bomb Attacks.* 2019.
7. Goodfellow I, *et al.*: *Explaining and Harnessing Adversarial Examples.* ICLR; 2015.
8. Wang H: *Multimedia Malware Detection Using Deep Neural Networks.* IEEE Access; 2024.
9. **Symantec Threat Intelligence Reports on Steganographic Malware.**
10. Chen K: **Secure Parsing of Complex Multimedia Formats.** *Computers & Security.* 2023.
11. Westfeld A, Pfitzmann A: *Attacks on Steganographic Systems.* Information Hiding; 1999.
12. Alotaibi M: **Adversarial Attacks on Image Processing Pipelines.** *Information Sciences.* 2024.
13. Provos N, Holz T: *Virtual Honeypots.* Addison-Wesley; 2007.
14. Li Y, Zhang X: **Deep Learning-Based Image Steganalysis: A Survey.** *ACM Computing Surveys.* 2023.
15. Singh R: **AI-Assisted Detection of Hidden Data in Digital Images.** *Expert Systems with Applications.* 2025.
16. Kozak K, *et al.*: **"PS_securescan synthetic dataset" Code & instruction to generate paper datasets: secure3d_cv_data_generator.py.** 2025.
Reference Source

Open Peer Review

Current Peer Review Status: ✔ ?

Version 2

Reviewer Report 15 July 2026

https://doi.org/10.21956/openreseurope.26254.r77184

© **2026 Kuznetsov O.** This is an open access peer review report distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

?

Oleksandr Kuznetsov
eCampus University, Novedrate, Italy

The revised version is more transparent and now presents PS-SecureScan as a lightweight proof-of-concept. However, the main concerns regarding reproducibility and experimental validity remain unresolved.
1) The released code package contains the scanner and a minimal usage example, but it does not provide the complete experimental pipeline required to reproduce Table 3 and Figure 5. The authors should release all 120 samples, data-generation scripts, exact weights and thresholds, per-file predictions and module scores, metric-calculation code, and plotting scripts.
2) The implementation is limited to partial PSD header and section checks, basic EXIF/XMP heuristics, entropy analysis of file-prefix bytes, approximate memory estimation, and an external sandbox callback. It does not implement deep PSD parsing, pixel- or frequency-domain steganalysis, semantic metadata validation, or behavioral sandbox monitoring. The method description, comparison table, results, and conclusions must be aligned strictly with these actual capabilities.
3) The reported metrics are internally inconsistent. For a dataset containing 80 adversarial and 40 benign files, Accuracy = 0.91, Precision = 0.88, Recall = 0.93, and FPR = 0.04 cannot be produced by a single confusion matrix. The authors must report TP, TN, FP, and FN, specify the classification threshold, provide the calculation code, and correct the reported values.
4) The revision mainly acknowledges the limitations but does not add substantive validation. At minimum, the synthetic evaluation must be fully reproducible. Claims about practical effectiveness would additionally require evaluation on real malicious image files and comparisons with relevant baselines under the same protocol.
Until these issues are resolved, the reported detection performance cannot be verified, and the article should not receive approval in its current form.

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: CS, AI, IoT

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Reviewer Report 15 July 2026

https://doi.org/10.21956/openreseurope.26254.r77183

© **2026 Almaliki A.** This is an open access peer review report distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Alaa Jabbar Qasim Almaliki
Universiti Utara Malaysia (UUM), Kedah Darul Aman, Malaysia

The authors have adequately addressed the concerns raised in my previous review and have substantially improved the manuscript.
The revised version clearly positions PS-SecureScan as a proof-of-concept framework, improves the transparency of the experimental evaluation, expands comparisons with existing approaches, and clarifies the implementation details of the proposed components, including sandbox analysis, metadata analysis, steganography detection, and PSD analysis.
The addition of dedicated sections on Threat Model, Limitations, and Future Work significantly enhances the rigor, clarity, and scientific value of the manuscript.
Overall, I believe that the authors have satisfactorily addressed the previous comments and that the manuscript is now suitable for indexing in Open Research Europe.

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Cybersecurity, Digital Forensics, Multimedia Security, Steganography, Image Security Analysis, Adversarial Attacks, Secure Media Processing, Information Security.

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Version 1

Reviewer Report 28 May 2026

https://doi.org/10.21956/openreseurope.25514.r74858

© **2026 Kuznetsov O.** This is an open access peer review report distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

?

Oleksandr Kuznetsov
eCampus University, Novedrate, Italy

The manuscript addresses an interesting and timely topic related to security risks in Photoshop-centered image workflows. The idea of pre-execution analysis of complex image files is practically relevant, and the authors made a positive effort to combine structural validation, metadata inspection, entropy analysis, and decompression-risk estimation into a unified framework. I also appreciate that the authors released the source code and dataset, which improves transparency and reproducibility compared to many similar works.
1) The current implementation is significantly simpler than the paper suggests. Several components described in the manuscript are implemented only as lightweight heuristics rather than full analysis modules.
2) The PSD analysis remains relatively shallow. The scanner does not perform deep parsing of layer hierarchies, masks, smart objects, or recursive PSD structures, although these aspects are emphasized throughout the paper.
3) The steganography analysis is very limited. The implementation mainly relies on simple entropy estimation and does not include robust pixel-domain or frequency-domain steganalysis methods.
4) The metadata analysis is also basic. EXIF/XMP parsing and semantic consistency checks are only partially implemented despite stronger claims in the manuscript.
5) The sandbox component is mostly conceptual. The repository contains only a user-defined execution hook rather than a real sandboxed behavioral monitoring environment.
6) The experimental evaluation remains weak. The dataset is synthetic and small, and the paper does not provide sufficiently detailed quantitative validation, statistical analysis, or comparison with stronger modern baselines.
7) Some risk-scoring decisions appear unstable. In several edge cases, obviously suspicious files may still produce relatively low final risk scores because of the chosen thresholds and weighting strategy.
Overall, the paper presents a useful prototype and a relevant security-oriented concept, but the current implementation and evaluation do not yet fully support some of the stronger technical claims made in the manuscript. The work is still valuable as an exploratory engineering framework, especially because the code is publicly available, but the scientific validation should be strengthened.

Is the work original in terms of material and argument?
Partly

Does it sufficiently engage with relevant methodologies and secondary literature on the topic?
Partly

Is the work clearly and cogently presented?
Yes

Is the argument persuasive and supported by evidence?
Partly

If any, are all the source data and materials underlying the results available?

Partly

Does the research article contribute to the cultural, historical, social understanding of the field?
Yes

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: CS, AI, IoT

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Author Response 04 Jun 2026

Karol Kozak

We thank the reviewers for their constructive feedback. The manuscript has been substantially revised and clarified. The current version positions PS-SecureScan as a proof-of-concept framework, improves transparency of the experimental evaluation with additional quantitative metrics, and expands comparisons with existing tools. We also refined descriptions of all components (including sandbox, metadata, steganography, and PSD analysis) to accurately reflect the current implementation.
Additionally, we introduced dedicated **Threat Model**, **Limitations**, and **Future Work** sections, updated the literature review with recent work, and clearly outlined directions for further development and real-world validation. These revisions significantly strengthen the clarity, rigor, and reliability of the paper.
We sincerely thank the reviewer for the careful evaluation of the manuscript and for recognizing the practical relevance of the proposed framework. We particularly appreciate the detailed technical comments regarding implementation scope and validation. The manuscript has been substantially revised to address these concerns.

Comment 1: *Several components are implemented as lightweight heuristics rather than full analysis modules.*
Response: We agree with the reviewer. To avoid overstating the implementation, we revised the manuscript and explicitly describe PS-SecureScan as a **proof-of-concept prototype**. We now clearly distinguish between the conceptual framework and the currently implemented functionality. Additional text was added to the Software Description section to clarify that several modules intentionally use lightweight heuristic approaches suitable for exploratory evaluation.

Comment 2: *PSD analysis remains relatively shallow.*
Response: We agree. The manuscript now explicitly states that the current implementation performs partial PSD structure validation rather than deep recursive parsing of layer hierarchies, masks, smart objects, or embedded resources. These capabilities are now discussed as future development directions within the newly added Future Work section.

Comment 3: *The steganography analysis is very limited.*
Response: We appreciate this observation. The manuscript has been revised to clarify that the current implementation relies primarily on entropy-based indicators and does not yet incorporate advanced machine-learning steganalysis or deep frequency-domain feature extraction. This limitation is now explicitly acknowledged in both the Method and Limitations sections.

Comment 4: *The metadata analysis is basic.*
Response: We agree. The revised manuscript clarifies that the current implementation focuses on structural metadata validation, size-based anomaly detection, and malformed metadata identification. Full semantic consistency analysis across EXIF, IPTC, and XMP metadata remains future work and is now explicitly stated.

Comment 5: *The sandbox component is mostly conceptual.*
Response: This was a valuable comment. The corresponding section has been substantially rewritten. The manuscript now clearly states that the current implementation provides a configurable sandbox integration interface rather than a complete behavioral monitoring subsystem. This modification aligns the manuscript more accurately with the released software implementation.

Comment 6: *The experimental evaluation remains weak.*
Response: We acknowledge this limitation. To improve the evaluation section, we introduced additional quantitative performance metrics and expanded the discussion of experimental methodology. We also strengthened the Limitations section and explicitly state that the current evaluation should be considered preliminary because it relies on synthetic adversarial datasets.

Comment 7: *Some risk-scoring decisions appear unstable.*
Response: We agree that the current risk-scoring model relies on empirically selected thresholds and weights. This limitation has been explicitly discussed in the revised manuscript. We also added a discussion indicating that future work will focus on calibration using larger datasets and machine-learning-assisted threshold optimization.
Additional Revisions To address the overall concerns raised by the reviewer, we additionally introduced:

- a dedicated **Threat Model** section,
- clearer separation between conceptual framework and implementation,
- revised descriptions of metadata, steganography, PSD parsing, and sandbox components,
- expanded Limitations and Future Work sections,
- updated literature review with recent publications,
- improved quantitative reporting.

We are grateful for the reviewer's detailed technical feedback, which significantly improved the clarity and accuracy of the manuscript.

Competing Interests: No competing interests were disclosed.

Reviewer Report 19 May 2026

https://doi.org/10.21956/openreseurope.25514.r74365

© **2026 Almaliki A.** This is an open access peer review report distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

?

Alaa Jabbar Qasim Almaliki
Universiti Utara Malaysia (UUM), Kedah Darul Aman, Malaysia

This article addresses an important and emerging cybersecurity problem related to image-based attacks in Photoshop-centred production workflows. The authors present PS-SecureScan, a pre-execution image analysis framework designed to detect anomalous or potentially malicious image files before they are opened in production software. The study combines structural validation, metadata anomaly detection, entropy-based steganographic analysis, decompression risk estimation, and sandbox-based behavioural observation within a unified risk-scoring architecture. The topic is timely and relevant, particularly considering the increasing complexity of modern image formats such as PSD, TIFF, PNG, and JPEG, as well as the growing use of creative media workflows in professional environments. The article is generally well organized and clearly written. The framework architecture is understandable, and the inclusion of software availability and reproducibility resources is appreciated.
However, several important limitations should be addressed to strengthen the scientific contribution of the work.
Major comments:

1. The experimental evaluation is currently limited to synthetic datasets. Although the use of controlled adversarial samples is useful for preliminary validation, the study lacks evaluation on real-world malicious image files, public benchmark datasets, or realistic attack scenarios. The inclusion of real attack samples or larger public datasets would significantly improve the credibility and applicability of the proposed framework.
2. The results section lacks sufficient quantitative depth. The manuscript mentions evaluation metrics such as TPR, FPR, Precision, and Recall; however, the paper does not provide detailed numerical results, confusion matrices, ROC curves, or statistical comparisons. More rigorous quantitative analysis is necessary to properly validate detection performance.
3. The comparison with existing approaches is relatively limited. Table 1 provides only a high-level comparison with basic tools such as ExifTool and StegExpose. The paper would benefit from comparison with more recent AI-based steganalysis approaches, multimedia malware detection systems, or anomaly detection frameworks published in recent cybersecurity literature.
4. The sandbox analysis component requires additional technical detail. The manuscript does not sufficiently explain the implementation environment, isolation mechanisms, monitoring strategy, or behavioral logging procedures used during runtime observation.
5. Several references are outdated, particularly in rapidly evolving areas such as adversarial image analysis, steganography detection, and image-based malware research. The authors are encouraged to include more recent literature from 2022–2026.

Minor comments:

1. Some sections contain repetitive explanations, particularly regarding PSD attack surfaces

and decompression attacks.
2. Figures are visually useful; however, some figure captions could be expanded with more technical interpretation.
3. The manuscript would benefit from clearer discussion of threshold calibration and parameter selection for the weighted risk-scoring model.

Overall, the study presents a promising and relevant framework for secure image workflow analysis. The idea is original and potentially valuable for cybersecurity and digital media production environments. However, the experimental evaluation and technical validation should be strengthened before the work can be considered fully mature scientifically.
Recommendation: Approved with Reservations.

Is the work original in terms of material and argument?
Yes

Does it sufficiently engage with relevant methodologies and secondary literature on the topic?
Partly

Is the work clearly and cogently presented?
Yes

Is the argument persuasive and supported by evidence?
Partly

If any, are all the source data and materials underlying the results available?
Partly

Does the research article contribute to the cultural, historical, social understanding of the field?
Partly

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Cybersecurity, Digital Forensics, Multimedia Security, Steganography, Image Security Analysis, Adversarial Attacks, Secure Media Processing, Information Security.

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Author Response 04 Jun 2026

Karol Kozak

We thank the reviewers for their constructive feedback. The manuscript has been substantially revised and clarified. The current version positions PS-SecureScan as a proof-of-concept framework, improves transparency of the experimental evaluation with

additional quantitative metrics, and expands comparisons with existing tools. We also refined descriptions of all components (including sandbox, metadata, steganography, and PSD analysis) to accurately reflect the current implementation.
Additionally, we introduced dedicated **Threat Model**, **Limitations**, and **Future Work** sections, updated the literature review with recent work, and clearly outlined directions for further development and real-world validation. These revisions significantly strengthen the clarity, rigor, and reliability of the paper.
Response to Reviewer 1 We sincerely thank the reviewer for the constructive and detailed comments. We appreciate the positive assessment of the overall framework concept and have carefully revised the manuscript to address all concerns raised. The major changes introduced in Revision 3 are summarized below.
Comment 1: *The experimental evaluation is currently limited to synthetic datasets.* **Response:** We agree with the reviewer. The current study remains limited to synthetic datasets generated to evaluate specific attack categories under controlled conditions. To clarify this limitation, we substantially revised the manuscript by expanding the **Limitations** section and explicitly positioning the current implementation as a proof-of-concept framework. We also introduced a dedicated **Future Work** section describing plans to evaluate PS-SecureScan on larger collections of real-world malicious image files and public multimedia security datasets.

Comment 2: *The results section lacks sufficient quantitative depth.*
Response: We thank the reviewer for this observation. To strengthen the quantitative evaluation, we introduced an additional performance summary table reporting Accuracy, Precision, Recall, and False Positive Rate. We also clarified the evaluation methodology and explicitly stated that the reported values represent preliminary results obtained from synthetic datasets. This modification improves the transparency and reproducibility of the evaluation.

Comment 3: *Comparison with existing approaches is limited.*
Response: We agree. The comparison section has been expanded and the role of PS-SecureScan within the existing ecosystem of image security tools has been clarified. Table 1 was revised to better emphasize differences between PS-SecureScan and existing tools such as ExifTool, StegExpose, and antivirus-based approaches. We also expanded the discussion of related work and added references to recent research on multimedia security, steganalysis, and image-based threat detection.

Comment 4: *The sandbox analysis component requires additional technical detail.* **Response:** This was an important observation. The previous manuscript description could indeed be interpreted as suggesting a fully implemented behavioral monitoring environment. The manuscript has been revised to clarify that the current prototype provides a lightweight sandbox integration interface and execution hook rather than a complete sandbox implementation. The corresponding section was rewritten to avoid overstating the current capabilities of the framework.

Comment 5: *Several references are outdated.*
Response: We appreciate this recommendation. The literature review has been updated with recent publications (2022–2025) covering AI-based steganalysis, multimedia malware

detection, secure multimedia parsing, adversarial image attacks, and image security analysis. These additions improve the relevance and timeliness of the manuscript.
Additional Revisions In addition to the requested changes, we introduced:

- a dedicated **Threat Model** section,
- a revised **Limitations** section,
- a new **Future Work** section,
- clarification of the prototype nature of the framework,
- improved quantitative evaluation reporting,
- expanded discussion of methodological assumptions.

We believe these revisions significantly strengthen the scientific quality and transparency of the manuscript. We thank the reviewer again for the valuable feedback.

Competing Interests: No competing interests were disclosed.