AnonShield: Scalable On-Premise Pseudonymization for CSIRT Network Vulnerability Data

Cristhian Kapelinski[1], **Douglas Lautert**[1], **Beatriz Machado**[1]
Diego Kreutz[1], **Isadora Garcia Ferrão**[2]

[1]AI Labs, Federal University of Pampa (UNIPAMPA)
[2]Université de Bretagne Occidentale (UBO)
{cristhianavilla,douglaslautert,beatrizmachado}.aluno@unipampa.edu.br
diegokreutz@unipampa.edu.br, isadora/garciaferrao@univ-brest.fr

Abstract. *We present AnonShield, a high-throughput, on-premise pseudonymization system for network vulnerability scan reports that combines GPU-accelerated NER, streaming processing, caching, and schema-aware configuration. Evaluated on datasets up to 550 MB (70,951 records), AnonShield reduces processing time from over 92 hours to under 10 minutes (up to 738× speedup), reaching F1 = 94.2% and Recall = 96.4% on a specialist-annotated validation set. Our results show that scalable pseudonymization of network vulnerability data is feasible without sacrificing analytical utility, enabling compliant data sharing in operational CSIRT environments.*

1. Introduction

Computer Security Incident Response Teams (CSIRTs) routinely process vulnerability scan reports generated by network scanners such as OpenVAS and Tenable across large and heterogeneous infrastructures, including containerized services, legacy systems, campus and backbone networks. Unlike code-level vulnerability data, these reports embed network-specific identifiers such as IP addresses, hostnames, TLS certificates, and service fingerprints, which collectively map the organization's network topology and attack surface [Kapelinski et al. 2025]. Sharing such data across organizations or machine learning pipelines exposes network topology and attack surfaces to potential adversaries, conflicting with CSIRT data governance policies and, when identifiers can be linked to individuals, raising GDPR/LGPD compliance concerns. Indeed, network identifiers like IP addresses may qualify as personal data under GDPR when linked to individuals [Albakri et al. 2019, Nweke and Wolthusen 2020]. As vulnerability datasets grow in volume, addressing this exposure at operational scale is critical for compliant data sharing and for enabling downstream uses such as LLM-based analysis [Xu et al. 2025] and the generation of synthetic cyber threat data [Almorjan et al. 2025].

The scale of this problem is rapidly increasing. In 2025, the National Vulnerability Database recorded 48,448 CVEs, a 20% increase over 2024, and forecasts indicate approximately 59,400 CVEs in 2026 [CVE Details 2026, FIRST 2026]. VulnCheck reported 884 Known Exploited Vulnerabilities in 2025, with nearly 29% exploited at or before disclosure [VulnCheck 2026]. In Brazil, CAIS/RNP continuously scans its infrastructure using Tenable, generating datasets exceeding 70,000 vulnerability records per cycle (Dataset D2). Such volume makes manual analysis unfeasible and demands LLM-assisted processing. However, feeding raw operational data into external pipelines conflicts with data sovereignty and GDPR/LGPD policies. Regulatory guidance under LGPD

recognizes pseudonymization as a key mechanism for enabling lawful data processing, ensuring that downstream tools, such as LLM-based analyzers, receive context-rich but identifier-free records [Bandel et al. 2025]. While vulnerability scan data primarily raises security and sovereignty concerns, incident data also exposes PII of victims, adding a direct LGPD/GDPR compliance dimension. The tension between sharing vulnerability data and protecting the embedded network identifiers that reveal an organization's asset inventory has been documented in Cyber Threat Intelligence (CTI) literature [Albakri et al. 2019, Nweke and Wolthusen 2020, Wagner et al. 2016], yet no widely adopted solution exists for large-scale vulnerability scan reports.

Table 1 positions existing approaches across five dimensions relevant to CSIRT deployments. The solution space divides into three categories, each with structural limitations. Commercial cloud frameworks such as Google Cloud DLP [Google Cloud 2018] and Amazon Comprehend [Amazon Web Services 2017] provide scalable pipelines with extensive entity coverage but require data transmission to external endpoints, violating data sovereignty constraints. Microsoft Presidio [Microsoft 2018], an open-source on-premise SDK for PII detection and anonymization that orchestrates NLP and RegEx recognizers via a unified analyzer engine, lacks recognizers for cybersecurity-specific entities such as CVE identifiers, CPE strings, certificate serial numbers, and cryptographic artifacts, leaving parts of the vulnerability profile exposed. IRI DarkShield [IRI 2017] shares these limitations and its commercial licensing blocks adoption in many academic and public-sector CSIRTs. Open-source tools address narrower problems. Anonip [Digitale Gesellschaft 2014] supports only IP anonymization, ARX [Prasser et al. 2014] focuses on tabular data and may degrade analytical utility through k-anonymity [Slijepčević et al. 2021], and LogLicker [Ahl 2023] relies on static RegEx patterns that lead to higher false-negative rates in heterogeneous vulnerability data.

Prior AnonLFI generations were the first solutions tailored to CSIRT contexts. AnonLFI v1.0 [Bandel et al. 2025] introduced a hybrid NER and RegEx pipeline validated on 763 incidents, achieving Precision of 100% and Recall of 97.38%, but relies on saltless SHA-256 and lacks native support for XML and JSON. AnonLFI v2.0 [Kapelinski et al. 2025] extended support to network vulnerability data, adding HMAC-SHA256, native XML and JSON processing, and OCR capabilities, achieving F1-scores of 76.5% for OCR and 92.13% for OpenVAS XML. However, its estimated processing time exceeds 92 hours for operational datasets with 70,951 records, making it impractical at scale. Datasets anonymized within the AnonLFI line have supported downstream LLM- and SLM-based incident classification in CSIRTs, reaching up to 92% accuracy in NIST SP 800-61r3 categorization [Severo et al. 2025, Almeida et al. 2025]. Taken together, no existing solution simultaneously satisfies four key technical challenges: (i) cryptographic pseudonymization with referential integrity, (ii) native processing of hierarchical formats, (iii) domain-specific entity recognition, and (iv) scalable on-premise execution.

This paper presents AnonShield, a pseudonymization framework that addresses the limitations of prior frameworks and tools. AnonShield extends prior designs with GPU-accelerated NER, LRU caching, streaming I/O, and a schema-aware `anonymization_config` mechanism that enables per-field policies without NER overhead. The contributions are threefold: (1) a scalable pseudonymization architecture for CSIRT network vulnerability data, (2) a comparative evaluation of four strate-

Table 1. Anonymization and pseudonymization tools and their primary gap in the CSIRT vulnerability context. US = Unstructured; S = Structured; – = in-memory only (no native file I/O).

System	Main Technique	Target Domain	Type	File Formats	Gap in CSIRT context
Microsoft Presidio [Microsoft 2018]	NLP + RegEx + extensible recognizers	Generic text	US	–	No CVE/CPE/hash recognizers; domain-irrelevant patterns produce FPs in vulnerability data
Google Cloud DLP [Google Cloud 2018]	NLP + FPE + PII classification	General data	US+S	`.txt, .csv, .json, .pdf, .docx`	Cloud endpoint conflicts with CSIRT data sovereignty policies; no cybersecurity-specific entities
Amazon Comprehend [Amazon Web Services 2017]	Cloud NLP for PII detection and redaction	Generic text	US	`.txt`	Cloud endpoint conflicts with CSIRT data sovereignty policies; no vulnerability scanner format support
IRI DarkShield [IRI 2017]	Multi-source masking + REST API	Corporate data	US+S	`.txt, .pdf, .xml, .json, .sql, images`	Commercial licence limits adoption in academic and public-sector CSIRTs; no cybersecurity-specific entities
LogLicker [Ahl 2023]	RegEx + masking + remapping manifest	CloudTrail & Generic logs	US+S	`.json, .txt`	RegEx only; lacks contextual NER; high manual maintenance for custom logs
Anonip [Digitale Gesellschaft 2014]	Bit masking of trailing IP bits via pipe	Web server logs	US	`.log, .txt`	Covers IP addresses only
ARX [Prasser et al. 2014]	k-anonymity, l, t, δ privacy models	Biomedical data	S	`.csv, .xlsx, sql`	Tabular data only; inapplicable to vulnerability reports
AnonLFI v1.0 [Bandel et al. 2025]	Hybrid NER + RegEx + SHA-256 (no salt)	Incidents	US+S	`.txt, .docx, .csv, .xlsx`	Saltless SHA-256 vulnerable to rainbow-table attacks; no native XML/JSON support; impractical at scale
AnonLFI v2.0 [Kapelinski et al. 2025]	Hybrid NER + RegEx + OCR + HMAC-SHA256 + native XML/JSON processors	Incidents & Network Vulnerabilities	US+S	`.xml, .json, .pdf, .txt, .csv, .docx, .xlsx,` images	Fails at operational scale; superlinear latency from DOM parsing; no GPU or streaming; high variability (CV $>$ 0.96); $>$92 hours for 550 MB
AnonShield (this work)	**Hybrid NER + RegEx + OCR + HMAC-SHA256 + GPU + LRU cache + streaming**	**Incidents & Network Vulnerabilities**	**US+S**	`.xml, .json, .pdf, .txt, .csv, .docx, .xlsx, .jsonl, .log,` images	

gies (`standalone`, `hybrid`, `presidio`, `filtered`) on datasets of up to 550 MB (70,951 records), including comparisons with AnonLFI v1.0 and v2.0, and (3) an accuracy evaluation using a statistically representative sample of 67 vulnerability records annotated by three specialists. The evaluation spans both a controlled testbed with 130 network services and an operational dataset from RNP infrastructure, demonstrating the feasibility of large-scale, on-premise pseudonymization.

2. The AnonShield Framework

AnonShield[1] is a pseudonymization framework for CSIRTs processing vulnerability scan reports and incident data, leveraging Microsoft Presidio's NER orchestration in most strategies. It detects and replaces sensitive entities with deterministic pseudonyms, preserving referential integrity across correlated reports. All processing is performed locally, mitigating privacy risks associated with third-party data sharing and supporting GDPR requirements [Amoo et al. 2024] and LGPD requirements.

The framework is organized into five sequential stages, illustrated in Figure 1.

(1) Input. AnonShield supports XML, JSON, JSONL, CSV, TXT, LOG, PDF (text and image), DOCX, and XLSX formats. Large files are processed via incremental *streaming*, using `ijson` for JSON and SAX parsing for XML, avoiding full in-memory loading.

[1] `https://github.com/AnonShield/tool`

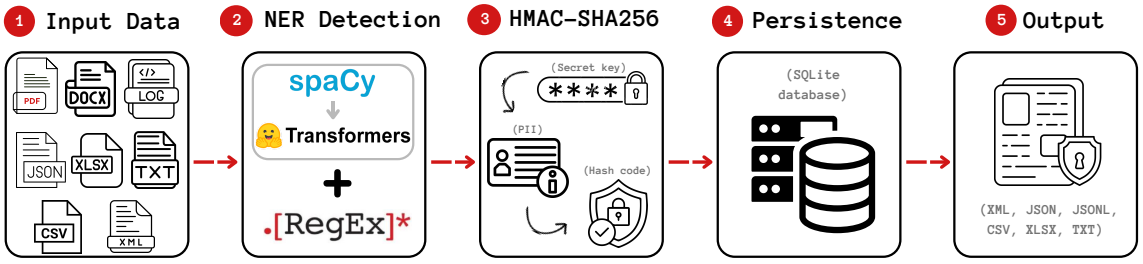

Figure 1. AnonShield pipeline.

(2) Entity Detection (NER). Before inference, extracted text is processed by a spaCy pipeline (`en_core_web_lg` / `pt_core_news_lg` for Presidio-based strategies, `spacy.blank()` for `standalone`) for tokenization and sentence segmentation, then submitted to the NER pipeline in batches of 8. For CSV, JSON, and XML, only field values traverse the NER pipeline. Detection combines a configurable transformer model (loaded via the Hugging Face `transformers` library) with 36 custom RegEx patterns covering 21 entity types (IPv4/IPv6, URLs, hostnames, hashes, certificates, CVE/CPE, and others). The default model is `Davlan/xlm-roberta-base-ner-hrl` (XLM-RoBERTa, 10 languages, PER/ORG/LOC). The alternative is the `attack-vector/SecureModernBERT-NER` (ModernBERT-large, ∼503K CTI spans, 22 cybersecurity entity types). Both are used off-the-shelf without fine-tuning. Inference runs on GPU (CUDA) when available with automatic CPU fallback, and uses an LRU cache to amortize repeated lookups across records.

(3) HMAC-SHA256 Pseudonymisation. Each entity is replaced by an HMAC-SHA256 slug computed with an operator-defined secret key. The slug length is configurable from 0 to 64 hexadecimal characters, and the entity type is preserved as a prefix, e.g., `100.111.20.23` → `[IP_ADDRESS_48624b5cdc]`.

(4) Persistence. The entity ↔ slug mapping is stored in a local SQLite database (`entity_type`, `original_name`, `slug_name`, `full_hash`, `first_seen`, `last_seen`), enabling controlled re-identification by an administrator holding both the database and the HMAC key. The database supports persistent and in-memory modes.

(5) Output. For structured formats (XML, JSON, JSONL, CSV, XLSX), the original schema and structure are preserved, with only sensitive values replaced. For PDF, DOCX, and image inputs, text is extracted via parsing or OCR, pseudonymised, and written to plain-text (`.txt`) output files.

Although publicly available, cybersecurity identifiers such as CVE IDs, CPE strings, UUIDs, and OIDs may expose an organization's vulnerability profile and enable targeted exploitation. Their sensitivity is therefore context-dependent. AnonShield addresses this by decoupling entity detection from replacement policy, enabling fine-grained control via `--entities-to-preserve` and `fields_to_exclude` for selective pseudonymization and field-level exclusion without altering the recognition pipeline.

AnonShield supports multiple anonymization strategies, selectable via `--anonymization-strategy`, to balance accuracy and performance. Presidio maximises recall but introduces false positives from domain-irrelevant recognizers.

Filtered mitigates this by restricting recognizers to cybersecurity-relevant entities, while Hybrid preserves detection behaviour and reduces overhead through a lightweight replacement mechanism, yielding identical accuracy. Standalone bypasses Presidio entirely for maximum performance. An experimental SLM-based strategy, exploring locally deployed small language models for entity detection, is currently under development and remains outside the scope of this paper's evaluation. Additionally, the schema-aware `anonymization_config` enables per-field control through `force_anonymize`, `fields_to_anonymize`, and `fields_to_exclude`, improving both accuracy and performance and supporting reusable templates for formats such as STIX 2.x. Full details, including entity coverage, strategies, and configuration options, are available in the tool's GitHub repository[1].

3. Experimental Methodology

The evaluation uses four datasets (Table 2), covering different scanning engines, operational scales, and processing challenges.

Table 2. Dataset summary.

ID	Source	Format	Size	Access
D1	OpenVAS (VulnLab)	CSV/TXT/PDF/XML	9.29/11.61/33.32/34.11 MB	Public
D1C	Converted from D1	XLSX/DOCX/JSON/PDF(img)	3.34/7.58/60.72/1,479.56 MB	Public
D2	Tenable (CAIS/RNP)	CSV/JSON	419.72/550.54 MB	Private
D3	Synthetic (Mock)	CSV/JSON	247.45/444.56 MB	Public

D1 – OpenVAS Heterogeneous Dataset. We scanned 130 targets from the VulnLab testbed across eight categories of network services and infrastructure (Table 3), yielding 520 reports in four formats. These reports, available in the AnonShield repository[1], include OpenVAS native `.anonymous` XML files, which demonstrate limitations in built-in redaction: while hostnames are masked, IP addresses, environment names, and internal UUIDs are retained in the output. Furthermore, 11 reports contained 71 unique cryptographic hashes and TLS fingerprints. Since these identifiers are often indexable by search engines, they may facilitate asset re-identification, indicating that native anonymization is insufficient for preventing information disclosure in these contexts.

Table 3. VulnLab service categories.

Category	Examples
Network & Infrastructure	DNS (Bind), FTP (ProFTPD), OpenLDAP
Web Applications & APIs	Juice Shop, DVWA, GraphQL
Web Servers & Platforms	Apache, Nginx, Tomcat, WordPress
Databases	MySQL, PostgreSQL, MongoDB, Redis
Monitoring & Logging	Elasticsearch, Prometheus, Grafana
DevOps & CI/CD	Jenkins, GitLab, Nexus
Messaging & Streaming	Kafka, RabbitMQ, Zookeeper
Operating Systems	Ubuntu (14/16), Debian (8/9), CentOS (6/7)

D1C – Converted Formats. Each D1 report was converted to XLSX, DOCX, JSON, and image-only PDF, yielding 520 additional files. Rasterized PDFs inflate mean size by $\approx 44\times$, forcing the pipeline to rely exclusively on its OCR subsystem.

D2 – CAIS/RNP Operational Dataset. Restricted Tenable scans comprising 70,951 vulnerability records (JSON 550 MB; CSV 420 MB), representative of large-scale enterprise environments and used as the primary operational benchmark. Not publicly released due to privacy and security constraints. In fact, publishing vulnerability scan data is infeasible even after partial PII removal: retaining a vulnerability name or identifier drastically reduces an attacker's search space. Removing all such identifiers to achieve safe publication would strip the dataset of its core content: a vulnerability dataset without vulnerability data has no analytical value. Consequently, for this data type, the most viable path to external sharing is synthetic data generation (an approach increasingly adopted for threat intelligence via LLMs [Almorjan et al. 2025]), while pseudonymization remains essential for enabling internal use of operational data in LLM pipelines without compromising organizational sovereignty or violating GDPR/LGPD. Pseudonymization and synthetic generation are thus complementary: anonymized data feeds LLMs that produce shareable synthetic datasets.

D3 – Synthetic Dataset. A public digital twin of D2, generated to replicate its structural and statistical characteristics. Unique identifiers from the original dataset were mapped to random entries from the official CVE list[2]. This process ensures that D3 maintains the same frequency distribution and entity density as the operational dataset while eliminating sensitive organizational information and specific vulnerability findings.

Hardware. All experiments ran on a single workstation: NVIDIA RTX 5060 Ti (16 GB VRAM), AMD Ryzen 5 8600G (6c/12t), 32 GB DDR5-6000 RAM.

Accuracy Evaluation. A statistically representative sample (n=67, population 6,472; 90% CL, E=10%) was annotated by three domain specialists across 13 entity types, yielding TP, FP, and FN counts for Precision, Recall, and F1-Score. AnonShield used `SecureModernBERT-NER`; a 9-entity preservation list (e.g., `TOOL`, `PLATFORM`, `MALWARE`) and `force_anonymize` policies for pattern-less fields (e.g., `Hostname`) were applied uniformly across all versions under comparison.

Statistical Analysis. A four-stage protocol was applied: Shapiro-Wilk normality test (α=0.05); Mann-Whitney U with Benjamini-Hochberg correction (p_{adj}<0.05); Cohen's d for effect size; and power-law regression (T=$a \cdot S^{\alpha}$) combined with polynomial fitting to separate fixed overhead from throughput scaling.

4. Results

We evaluate AnonShield on two dimensions: processing performance across heterogeneous (D1), converted (D1C), and large-scale operational (D2/D3) datasets; and pseudonymization accuracy on a 67-record specialist-annotated validation set.

4.1. Processing Performance

Heterogeneous formats (D1). Table 4 (Appendix A) reports latency across 520 files (N=260 runs/strategy). AnonShield_standalone consistently achieves the lowest latency and highest stability across all formats, with speedups of 16.51× (XML), 9.56× (CSV), 3.27× (PDF), and 3.04× (TXT) over AnonLFI v2.0, all statistically meaningful sizes (Cohen's $d \geq 0.42$). AnonLFI v2.0 exhibits pathological variability ($CV > 0.96$) caused by

[2] `https://cve.org/downloads`

memory-intensive DOM parsing; AnonShield reduces this to $CV < 0.8$ through iterative streaming and GPU inference. Notably, AnonLFI v2.0 regresses against AnonLFI v1.0 on CSV and TXT due to increased recognizer complexity without architectural compensation, a limitation addressed by AnonShield. Scalability analysis (Figure 2, Appendix A) confirms near-linear complexity ($\alpha \approx 1$) for AnonShield, with an amortization effect driving throughput above 60 KB/s for larger files, while AnonLFI v2.0 degrades superlinearly.

Converted formats (D1C). Results for XLSX, DOCX, JSON, and image-only PDF are detailed in Table 5 (Appendix B). AnonShield_standalone achieves a $23.24\times$ speedup on JSON (CV=0.60), $5.00\times$ on XLSX, and $2.08\times$ on DOCX. For image-only PDF, where the $\approx 44\times$ size inflation shifts the bottleneck to OCR, speedup narrows to $1.64\times$ while maintaining linear scaling ($R^2 \geq 0.989$). A single file (*openssh-server_images.pdf*) caused 2 AnonShield failures due to a malformed content stream triggering `PyMuPDF`'s strict parser, an edge case absent in AnonLFI v2.0 only because it lacks the corresponding memory-management step.

Operational scale (D2 and D3). Table 6 and Table 7 (Appendix C) present the most consequential results of this evaluation. On D2 (70,951 records, 550 MB JSON), AnonShield_standalone completes processing in 453 s (1,250 KB/s), whereas AnonLFI v2.0's estimated runtime exceeds ∼92 hours, a speedup of $\geq 738\times$. Activating the schema-aware `anonymization_config` reduces D2-CSV runtime further to 12.55 s (34,341 KB/s), a $46.9\times$ gain over full NER inference, by enabling deterministic field-level remapping that bypasses the inference pipeline entirely. On D3, GPU acceleration yields speedups of up to $3{,}532\times$ over AnonLFI v2.0; even on CPU-only hardware, AnonShield_standalone achieves $\geq 535\times$ speedup, confirming that performance gains are architectural rather than hardware-dependent. Throughput on D3 (3,473 KB/s, CSV) exceeds D2 (732 KB/s) due to higher entity redundancy enabling effective LRU cache reuse; D2's lower cache hit rate, driven by high-entropy operational data, reflects realistic worst-case deployment conditions. All strategy comparisons are statistically significant ($p_{\text{adj}} < 0.001$, using the Mann–Whitney U test with Benjamini–Hochberg correction).

4.2. Accuracy

Table 8 presents results on the 67-record specialist-annotated validation set (13 entity types). Recall is the operationally critical metric: an unredacted False Negative exposes sensitive infrastructure details to adversaries and, if identifiers qualify as personal data, may breach GDPR/LGPD, whereas a False Positive degrades analytical utility without compromising security or compliance. Accuracy improves substantially across generations: AnonLFI v1.0 achieves F1 = 23.8% (Recall = 18.8%); AnonLFI v2.0 reaches 54.2% (Recall = 40.7%); and AnonShield `filtered`/`hybrid` achieve F1 = 94.2% at Recall = 96.4%. `Standalone` reaches essentially the same accuracy (F1 = 93.8%, Recall = 96.1%) while delivering the highest throughput and lowest latency, making it the recommended default for operational workloads. The remaining 0.4 pp gap stems from `standalone`'s TLD-whitelist URL regex: it misses URLs with non-public TLDs (e.g., `.trabalho_vulnnet`) that Presidio's built-in recognizer catches as partials (losing TP credit), and truncates URLs with compound TLDs (e.g., `.co.uk`) leaving the suffix exposed (extra FN). The `presidio` strategy matches the same Recall but at lower Precision (71.9%), favoring high-sensitivity environments where over-anonymization is acceptable.

Full False Negative and False Positive breakdowns are provided in Appendix D.

5. Availability, Implementation and Demonstration Plan

AnonShield is publicly available on GitHub[1], including source code, documentation, datasets, and all evaluation artefacts. The repository provides step-by-step installation instructions, a minimal functional test executable in under 5 minutes, and scripts to reproduce the main experimental results. The tool is implemented in Python (3.12), uses Microsoft Presidio for NER orchestration, and is managed via `uv`, with no external service dependencies. It is also distributed as a Docker image in CPU and GPU (CUDA) variants on Docker Hub[3]. Full reproducibility artefacts are listed in the repository[4].

For the SBRC Tools Lounge, the demonstration will run on a standard laptop (4 vCPUs, 8 GB RAM) running Ubuntu, without requiring specialized hardware or network infrastructure. A GPU is optional, as the CPU version is fully functional. The demonstration includes live pseudonymization of OpenVAS reports, comparison of anonymization strategies, and illustration of throughput gains enabled by the schema-aware `anonymization_config` mechanism.

6. Conclusion and Future Work

This paper presented AnonShield, the third generation of the AnonLFI research line, achieving up to $738\times$ speedup (from $\sim$92.9 hours to under 10 minutes on 550 MB) and $\geq 535\times$ on CPU-only hardware. The `standalone` strategy combines the highest throughput with accuracy on par with `filtered`/`hybrid` (F1 = 93.8% vs. 94.2%, Recall = 96.1% vs. 96.4%), while `anonymization_config` provides additional gains of up to $47\times$. These results demonstrate that large-scale, on-premise pseudonymization is operationally practical for CSIRT workflows.

Two boundaries define the current scope. First, the framework is constrained by evaluation and processing limits: the accuracy baseline relies on a restricted sample size and dataset diversity, format extraction suffers from OCR/PyMuPDF breaks, and NER misses entities it fails to recognize. To structurally bypass format issues, we are extracting vulnerabilities directly as structured datasets from PDFs [Machado et al. 2025]. Second, formal privacy models (k-anonymity) primarily suit incident data with victim PII; for vulnerability contexts, publishing raw records remains infeasible regardless of the anonymization technique.

As future work, we plan to advance AnonShield in five directions: improving robustness for complex formats, especially image-only PDFs and malformed streams; extending entity recognition with cybersecurity-specific recognizers and locally deployed SLMs to reduce false negatives, including a more permissive URL recognizer in `standalone` to close the residual gap with Presidio-based strategies on non-public TLDs; expanding the evaluation to larger and more diverse CSIRT datasets; evaluating the analytical utility of pseudonymized datasets through downstream tasks such as vulnerability classification and trend analysis; and investigating formal privacy-utility trade-off models for secure data sharing and LLM-assisted cyber analytics.

[3] `https://hub.docker.com/r/anonshield/anon`

[4] `https://github.com/AnonShield/tool/blob/main/paper_data/AVAILABILITY.md`

Acknowledgments

This work was partially supported by RNP, CNPq (Grant 409743/2025-9), and FAPERGS (Grants 24/2551-0001368-7 and 24/2551-0000726-1).

References

Ahl, C. (2023). LogLicker: Anonymizing logs made easy. `https://github.com/Permiso-io-tools/LogLicker`. Permiso Security. Accessed: 2026.

Albakri, A., Boiten, E., and De Lemos, R. (2019). Sharing cyber threat intelligence under the General Data Protection Regulation. In *Privacy Technologies and Policy*, LNCS.

Almeida, G., Pohlmann, M., Severo, A., Kreutz, D., Heinrich, T., and Pereira, L. (2025). On-premise SLMs vs. commercial LLMs: Prompt engineering and incident classification in SOCs and CSIRTs. In *XXII ERRC*.

Almorjan, A., Basheri, M., and Almasre, M. (2025). Large language models for synthetic dataset generation of cybersecurity indicators of compromise. *Sensors*, 25(9):2825.

Amazon Web Services (2017). Amazon comprehend. `https://aws.amazon.com/comprehend/`. Accessed: 2026.

Amoo, O. O., Atadoga, A., Osasona, F., Abrahams, T. O., Ayinla, B. S., and Farayola, O. A. (2024). GDPR's impact on cybersecurity: A review focusing on USA and European practices. *International Journal of Science and Research Archive*, 11:1338–1347.

Bandel, C. T., Esteves, J. P. R., Guerra, K. P., Bertholdo, L. M., Kreutz, D., and Miani, R. S. (2025). Anonimização de incidentes de segurança com reidentificação controlada. In *Anais do SBSeg 2025*.

CVE Details (2026). Browse CVE vulnerabilities by date. Accessed: 2026-03-26. Reports 48,448 CVEs in 2025 and 40,308 in 2024.

Digitale Gesellschaft (2014). Anonip – IP address anonymisation tool. `https://github.com/DigitaleGesellschaft/anonip`. Accessed: 2026.

FIRST (2026). Vulnerability forecast. Median forecast: 59,427 CVEs in 2026.

Google Cloud (2018). Cloud data loss prevention (Cloud DLP). `https://cloud.google.com/sensitive-data-protection`. Accessed: 2026.

IRI (2017). IRI DarkShield – data discovery and masking. `https://www.iri.com/products/iri-darkshield/`. Accessed: 2026.

Kapelinski, C., Lautert, D., Machado, B., and Kreutz, D. (2025). AnonLFI 2.0: Extensible architecture for PII pseudonymization in CSIRTs with OCR and technical recognizers. In *ERRC 2025*.

Machado, B., Lautert, D., Kapelinski, C., and Kreutz, D. (2025). Structured extraction of vulnerabilities in openvas and tenable was reports using llms. In *XXII ERRC*.

Microsoft (2018). Presidio – data protection and de-identification SDK. `https://github.com/microsoft/presidio`. Accessed: 2026.

Nweke, L. O. and Wolthusen, S. (2020). Legal issues related to cyber threat information sharing. In *Proc. CyCon*. NATO CCDCOE.

Prasser, F., Kohlmayer, F., Lautenschlager, R., and Kuhn, K. A. (2014). ARX – a comprehensive tool for anonymizing biomedical data. *AMIA*, pages 984–993.

Severo, A., Lautert, D., Almeida, G., Kreutz, D., Rodrigo, G., Pereira Jr, L., and Bertholdo, L. (2025). LLMs e engenharia de prompt para classificação automatizada de incidentes em SOCs. In *XXV SBSeg*.

Slijepčević, D., Hein, D., Zec, M., and Kaltenbrunner, M. (2021). k-anonymity in practice: How generalisation and suppression affect machine learning classifiers. *Computers & Security*, 111:102488.

VulnCheck (2026). State of exploitation 2026. 884 KEVs identified in 2025; 28.96% exploited on or before CVE publication date.

Wagner, C., Dulaunoy, A., Wagener, G., and Iklody, A. (2016). MISP: The design and implementation of a collaborative threat intelligence sharing platform. In *ACM WISCS*.

Xu, H., Wang, S., Li, N., Wang, K., Zhao, Y., Chen, K., Yu, T., Liu, Y., and Wang, H. (2025). Large language models for cyber security: A systematic literature review. *ACM Transactions on Software Engineering and Methodology*.

A. Performance and Scaling Analysis (Dataset D1)

Performance metrics for 520 files ($N = 260$ runs/strategy) are summarized in Table 4. Due to file-size heterogeneity and right-skewed distributions (Shapiro-Wilk $p < 10^{-6}$), the median is used as a robust measure for D1, while the mean ($\bar{t} \pm \sigma$) is reserved for D2/D3. Across these runs, the `standalone` strategy consistently outperforms Presidio-based versions ($1.3\times$–$1.4\times$ faster) by bypassing orchestration overhead. Further scalability insights (Figure 2, Panel C) highlight an amortization effect: as file size increases, AnonShield_standalone throughput rises (peaking > 60 KB/s), effectively spreading fixed startup costs, whereas AnonLFI v2.0 shows degrading throughput, confirming its inefficiency for operational-scale workloads.

Table 4. Processing latency on D1 reports.

Format	Version/Strategy	Mean (s)	Median (s)	Max (s)	CV	Speedup	Cohen's d
XML	AnonLFI v2.0 (Baseline)	191.89	176.44	1717.71	0.96	1.00×	–
	AnonShield_presidio	15.62	13.69	85.58	0.60	12.28×	1.29 (Large)
	AnonShield_standalone	**11.62**	**10.02**	**72.34**	**0.63**	**16.51×**	**1.32 (Large)**
CSV	AnonLFI v2.0 (Baseline)	74.18	37.01	1442.08	1.75	1.00×	–
	AnonLFI v1.0	37.92	20.41	577.96	1.55	1.96×	0.35 (Small)
	AnonShield_presidio	10.65	9.08	60.62	0.53	6.97×	0.67 (Medium)
	AnonShield_standalone	**7.76**	**6.67**	**47.87**	**0.53**	**9.56×**	**0.70 (Medium)**
PDF	AnonLFI v2.0† (Baseline)	27.38	13.01	304.24	1.70	1.00×	–
	AnonShield_presidio	11.20	9.25	68.13	0.58	2.44×	0.47 (Small)
	AnonShield_standalone	**8.36**	**6.91**	**50.78**	**0.60**	**3.27×**	**0.56 (Medium)**
TXT	AnonLFI v2.0 (Baseline)	31.23	12.61	747.74	2.20	1.00×	–
	AnonLFI v1.0	20.28	10.98	354.91	1.56	1.54×	0.19 (Negligible)
	AnonShield_presidio	13.10	10.21	100.28	0.73	2.38×	0.36 (Small)
	AnonShield_standalone	**10.28**	**7.91**	**80.71**	**0.78**	**3.04×**	**0.42 (Small)**

† AnonLFI v2.0 PDF failed runs excluded due to out-of-memory errors. †† Presidio-based strategies are statistically similar ($p_{adj} > 0.46$), with AnonShield_presidio as representative. AnonLFI v1.0 does not support XML or PDF.

B. Converted Format Benchmarks (D1C)

To evaluate versatility, 130 reports were converted into XLSX, DOCX, JSON, and image-only PDF ($N = 260$ runs/strategy). Table 5 summarizes the results, with significant differences across strategies ($p < 10^{-5}$). AnonLFI v2.0 regressed on XLSX and DOCX due to additional recognizers without architectural optimizations, while the 2 failed AnonShield runs on image-only PDFs were caused by `PyMuPDF` parser errors triggered by malformed content streams.

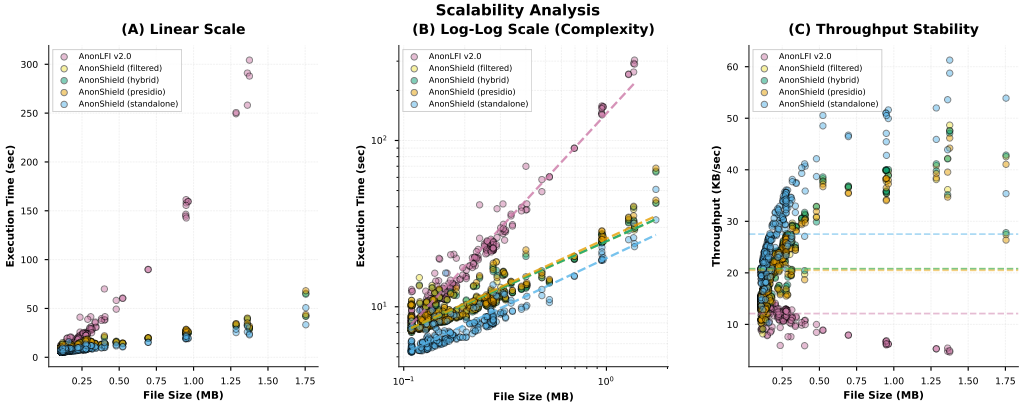

Figure 2. Scalability analysis on D1 PDF subset: (A) Linear scale, (B) Log-Log complexity ($\alpha \approx 1$ for AnonShield), and (C) Throughput stability.

Table 5. Processing latency on D1-converted formats.

Format	Version/Strategy	Mean (s)	Max (s)	CV	Speedup	Cohen's d
JSON	AnonLFI v2.0 (Baseline)	246.55	1861.01	1.16	1.00×	–
	AnonShield_presidio	20.15	110.78	0.72	12.23×	1.08 (Large)
	AnonShield_standalone	**10.61**	**49.02**	**0.60**	**23.24×**	**1.12 (Large)**
XLSX	AnonLFI v1.0 (Baseline)	35.51	320.73	1.37	1.00×	–
	AnonLFI v2.0	60.30	596.95	1.52	0.59×	−0.34 (Small)
	AnonShield_presidio	9.86	38.64	0.47	3.60×	0.73 (Medium)
	AnonShield_standalone	**7.11**	**27.95**	**0.47**	**5.00×**	**0.82 (Large)**
DOCX	AnonLFI v1.0 (Baseline)	19.58	186.55	1.34	1.00×	–
	AnonLFI v2.0	29.90	431.60	1.94	0.65×	−0.23 (Small)
	AnonShield_presidio	12.16	63.24	0.68	1.61×	0.36 (Small)
	AnonShield_standalone	**9.43**	**51.51**	**0.73**	**2.08×**	**0.52 (Medium)**
PDF-image†	AnonLFI v2.0 (Baseline)	59.15	757.63	1.85	1.00×	–
	AnonShield_presidio	37.85	362.81	1.46	1.56×	0.23 (Small)
	AnonShield_standalone	**36.04**	**348.63**	**1.54**	**1.64×**	**0.27 (Small)**

† Failures: 1 file (*openssh-server_images.pdf*) failed in both runs ($N = 2$) across all AnonShield strategies.
††`AnonShield_presidio` represents the Presidio cluster. Speedup for JSON/PDF-image vs. AnonLFI v2.0; XLSX/DOCX vs. AnonLFI v1.0.

C. Large-Scale Operational Benchmarks (D2 and D3)

AnonShield was evaluated against operational datasets D2 (Tenable, 550 MB) and D3 (Mock CVE, 444 MB). Due to the prohibitive slowness of AnonLFI v1.0 and v2.0, their runtimes were estimated based on D1 throughput. For CSV, these are **conservative lower bounds** ($\geq$121 hours), as v2.0's superlinear scaling significantly inflates costs as file sizes grow. For D3, an additional CPU-only benchmark (no GPU) was conducted to assess whether AnonShield's throughput gains depend on GPU hardware.

Analysis of Large-Scale Impact. Without schema-aware configuration (full NER inference active), the GPU variant is up to $\sim$6.6× faster than CPU on CSV and $\sim$5.1× on JSON; with config (inference bypass), both perform similarly ($\sim$8–9 s).

D. Accuracy Evaluation

Partial anonymizations were counted as 1 TP + 1 FN, and counts were taken from actual occurrences in the original text rather than from pseudonym counts in the output, to avoid double-counting fragmented or merged pseudonyms.

Table 6. D2 (Operational) processing times (70,951 records).

Mode	Version / Strategy	CSV (419.72 MB) Time (s)	CSV (419.72 MB) KB/s	JSON (550.54 MB) Time (s)	JSON (550.54 MB) KB/s
No config	AnonLFI v2.0 *(est.)*	$\geq$437,335	0.98	$\sim$334,472	1.69
	AnonShield_presidio	$2{,}520.7 \pm 68.0$	171	985.7 ± 73.7	575
	AnonShield_standalone	$\mathbf{588.5 \pm 30.7}$	**732**	$\mathbf{453.1 \pm 35.9}$	**1,250**
With config	AnonShield_presidio	13.42 ± 0.13	32,034	18.88 ± 0.16	29,855
	AnonShield_standalone	$\mathbf{12.55 \pm 0.74}$	**34,341**	$\mathbf{18.03 \pm 0.23}$	**31,272**
Speedup standalone vs. v2.0 (no config)		$\mathbf{\geq 743\times}$		$\mathbf{\sim 738\times}$	

AnonShield_standalone is significantly faster than Presidio-based strategies ($p_{adj} < 0.001$).

Table 7. D3 (Synthetic) processing times.

Mode	Version / Strategy	CSV (247.45 MB) Time (s)	CSV (247.45 MB) KB/s	JSON (444.56 MB) Time (s)	JSON (444.56 MB) KB/s
No config (GPU)	AnonLFI v2.0 *(est.)*	$\geq$257,835	0.98	$\sim$270,086	1.69
	AnonShield_presidio	318.0 ± 5.1	797	339.6 ± 14.4	1,343
	AnonShield_standalone	$\mathbf{73.0 \pm 1.6}$	**3,473**	$\mathbf{172.1 \pm 6.2}$	**2,647**
No config (CPU)	**AnonShield_standalone**	$\mathbf{481.5 \pm 8.9}$	**526**	$\mathbf{881.9 \pm 57.7}$	**518**
With config	AnonShield_presidio	8.89 ± 0.06	28,517	21.11 ± 0.22	21,571
	AnonShield_standalone	$\mathbf{7.96 \pm 0.08}$	**31,856**	$\mathbf{20.43 \pm 0.81}$	**22,314**
Speedup standalone (GPU) vs. v2.0		$\mathbf{\geq 3{,}532\times}$		$\mathbf{\sim 1{,}569\times}$	
Speedup standalone (CPU) vs. v2.0		$\mathbf{\geq 535\times}$		$\mathbf{\sim 306\times}$	

Standalone vs. Presidio comparisons are statistically significant in all cases ($p_{adj} < 0.001$).

Table 8. Accuracy on 67-record validation set (D1 OpenVAS CSV).

Version/Strategy	TP	FP	FN	Prec.	Rec.	F1
AnonLFI v1.0	108	225	466	32.4%	18.8%	23.8%
AnonLFI v2.0	283	67	412	80.9%	40.7%	54.2%
AnonShield_presidio	733	286	27	71.9%	96.4%	82.4%
AnonShield_filtered	733	63	27	92.1%	96.4%	94.2%
AnonShield_hybrid	733	63	27	92.1%	96.4%	94.2%
AnonShield_standalone	**730**	**66**	**30**	**91.7%**	**96.1%**	**93.8%**

False Negatives. All AnonShield strategies share 27 FNs from two sources: (1) 19 unrecognized organization names (*Oracle*, *ISC*, *Wiesemann & Theis*) in formulaic attribution strings, where context is insufficient for NER disambiguation; and (2) 8 partial URLs and hostnames embedded in descriptive prose where the analyzer's boundary detection fails (e.g., `prometheus-old.trabalho_vulnnet` in a Reverse-DNS table). `Standalone` adds 3 FNs from missed URLs (cf. Section 4).

False Positives. `AnonShield_presidio` produces 286 FPs, mainly from Presidio's `DATETIME` and `IP_ADDRESS` recognizers misclassifying version strings (e.g., `2.4.51` $\rightarrow$ `[DATE_TIME_...]`). `Filtered`, `hybrid`, and `standalone` suppress this to 63–66 via curated recognizer subsets; the residual FPs come from version strings still captured by the custom IP regex (e.g., `8.1.31` $\rightarrow$ `[IP_ADDRESS]`), the QoD field value (`80` $\rightarrow$ `PORT`), and isolated cases of `localhost` and file paths (`jquery-1.6.2.js`, `sysctl.conf`) detected as `HOSTNAME`. A systematic AnonLFI v2.0 misclassification of `Severity: Medium` as `ORGANIZATION` is fully resolved in AnonShield via `anonymization_config`.