SHOGUN: An Integrated Dashboard for Vulnerability Analysis via Internet-Wide Search Engines

Open Source

Isabelle Matos[1] **Lucas M Ponce**[2] **Igor Cunha**[1] **Pedro Almeida**[2]
Elverton Fazzion[1,2] **Ítalo Cunha**[2] **Cristine Hoepers**[3] **Klaus Steding-Jessen**[3]
Marcelo H. P. C. Chaves[3] **Dorgival Guedes**[2] **Wagner Meira Jr.**[2]

[1]DCOMP/UFSJ [2]DCC/UFMG [3]CERT.br/NIC.br

{isabelle.mattos25,igoraserpac}@aluno.ufsj.edu.br fazzion@ufsj.edu.br

{lucasmsp,pedromeireles,cunha,dorgival,meira}@dcc.ufmg.br

{cristine,jessen,mhp}@cert.br

Abstract. *The increasing exposure of digital assets on the Internet demands efficient solutions for identifying and continuously monitoring vulnerabilities. While search engines like Shodan provide a targeted retrieval of exposure data, longitudinal risk assessment requires extensive data preparation. To bridge this gap, we present SHOGUN, an interactive dashboard engineered for the multidimensional analysis of internet-wide vulnerability data. Built upon a decoupled architecture, our tool abstracts raw search engine dumps into structured analytical datasets. The tool features a reactive web interface that combines geospatial and statistical visualizations with a cross-view drill-down navigation model, allowing security analysts to transition from macro-level network overviews to granular, asset-level forensic investigations.*

1. Introduction

The integration of smart systems and automated services into daily life has driven an unprecedented increase in the number of devices connected to the Internet [Abuserrieh and Alalfi 2024]. While this technological expansion fosters innovation and operational efficiency across multiple domains, it also significantly amplifies the global attack surface, with ransomware attacks growing by 42% worldwide in 2025 [Telesíntese 2026]. As the volume of exposed digital assets grows, traditional security monitoring and network control approaches struggle to keep pace with the increasing frequency and scale of contemporary cyber threats [Mousavi et al. 2020]. Consequently, identifying and monitoring vulnerabilities in this interconnected landscape has become a critical challenge for organizations and network operators alike. To address this challenge, network reconnaissance platforms such as Shodan [1], Censys [2], and ZoomEye [3] have become widely adopted for discovering and characterizing exposed devices at Internet scale [Li et al. 2020, Ponce et al. 2024, Huang et al. 2025].

Despite their utility, these tools are primarily designed for fast, targeted information retrieval. Although systems such as Shodan provide some interfaces for longitudinal dashboards, they are still fairly basic, offering limited information derived from

[1]Shodan: https://shodan.io
[2]Censys: https://censys.io
[3]ZoomEye: https://www.zoomeye.org

static data.[4] As analytical requirements evolve toward more sophisticated or longitudinal investigations, the need for extensive data preparation, such as large-scale cleaning, normalization, and integration with external datasets, becomes more pronounced [Abuserrieh and Alalfi 2024, Ponce et al. 2024]. Therefore, moving beyond basic queries requires the development of additional processing pipelines, which highlights the need for frameworks capable of streamlining these complex analytical workflows.

To address these challenges, this paper introduces SHOGUN,[5] an interactive and scalable framework for the multidimensional analysis of security vulnerabilities. The framework is designed to streamline the processing of large-scale monitoring data, providing a structured platform for integrated queries and risk assessment. The platform's architecture enables both geo-spatial and statistical visualization of exposed devices, facilitating the evaluation of security configurations and supporting analytical workflows in research and operational environments.

Given these capabilities, the main objective of this paper is to present the design, architecture, and practical application of SHOGUN as a framework that bridges the gap between raw exposure data and actionable cybersecurity intelligence. Specifically, this work aims to (i) describe the abstraction layer that simplifies the integration of Shodan data with external vulnerability knowledge bases such as EPSS (Exploit Prediction Scoring System), NIST/NVD (National Vulnerability Database), and CISA KEV (Known Exploited Vulnerabilities Catalog); (ii) detail the interactive, drill-down navigation model that connects macro-level risk overviews to granular, asset-level investigations; and (iii) demonstrate the framework's practical applicability through a case study using real-world exposure data. The remainder of this paper is organized as follows: Section 2 discusses the framework architecture; Section 3 describes the setup; Section 4 details the performance evaluation; Section 5 presents a demonstration of the framework using real-world data; and Section 6 concludes the paper with final remarks and directions for future work.

2. Architecture

The SHOGUN dashboard utilizes an architecture based on Front-end, Back-end, and Data Store to provide mechanisms for modularity, security, and the scalability of its components. As shown in Figure 1, by separating the user-facing interface from the data processing layer, the structure preserves the application's responsiveness, enabling the efficient exploration of large volumes of exposed devices. In the input flow, Nginx manages incoming requests, forwarding them to a WSGI server where Flask acts as the central application orchestrator. From there, tasks are delegated: queries for summarized data are processed by Dash, while detailed analysis of individual records is rendered using standard web technologies, such as HTML and JavaScript.

2.1. Abstraction and Analysis Engine

The Back-end of SHOGUN acts as the core orchestration and data processing layer, designed to handle large-scale vulnerability monitoring data efficiently. The ingestion pipeline monitors external internet-wide search engines.[6] Platforms such as Shodan pro-

[4] An example is available at: https://exposure.shodan.io/#/BR
[5] Available at: https://github.com/lucasmsp/shogun-dashboard
[6] The current version focuses on Shodan data because it contains more detailed vulnerability information.

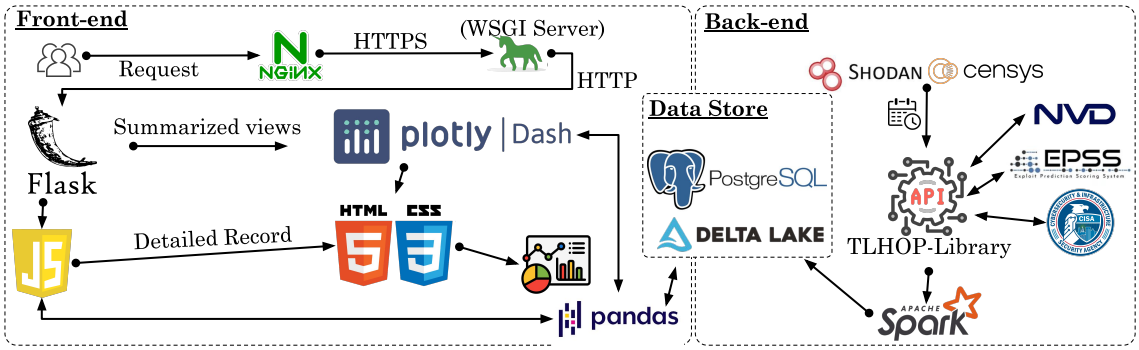

Figure 1. Tool architecture.

vide comprehensive metadata regarding open ports, active services, and exposed digital assets. However, because of the massive volume of this type of source, using search engines to perform longitudinal analytical workflows requires extensive filtering, cleaning, and structured aggregation to generate actionable intelligence, making traditional processing approaches prohibitive. To bridge the gap between raw data and high-level analysis, SHOGUN leverages a data abstraction layer powered by the TLHOP-Library [Ponce et al. 2024], an Apache Spark-based library capable of handling significant daily volumes of raw data.

For our purposes, the library was extended with a new vulnerability assessment module that serves as an operational roadmap for generating context-aware datasets. This extension implements a pipeline to filter records with vulnerabilities identified by the search engines and aggregate results within the specific context of the Brazilian Internet infrastructure. To ensure that risk metrics remain precise and up to date, we use the TLHOP-Library functionality to download new information provided by the National Vulnerability Database (NVD), the Exploit Prediction Scoring System (EPSS), and the Known Exploited Vulnerabilities catalog (CISA KEV).

Our tool leverages a hybrid Data Store approach: (i) a PostgreSQL instance handles user preferences, saved items, and configuration tables; (ii) Delta Lake files[7] serve as the high-performance analytical storage consumed by the Front-end. The output of the Spark pipeline is stored as versioned, highly compressed Delta tables. Currently, the framework generates seven distinct Delta files, which function as relational tables, as summarized in Table 1. The primary file is the GENERAL RECORDS table, which consolidates all vulnerable hosts discovered by the search engine. As discussed in Section 4, generating this daily report takes 2 minutes and 39 seconds on average. Subsequently, all other auxiliary files are derived using this primary dataset as a starting point, requiring an average of only 30 seconds to process all remaining aggregated tables combined.

The ingestion and analysis engine is fully automated through a system CRON scheduler. Alongside data generation, the pipeline runs automated data lifecycle routines, performing data pruning via Delta Lake native functionalities to remove old records (retaining a rolling window of 7 days by default).

[7]Delta Lake: https://delta.io

Table 1. Summary of each table and their key fields.

TABLE	KEY FIELDS
SUMMARY	Grouping by EPSS rank: # CVEs, organizations, IPs, ports, and ASNs.
ORGS	Impact by organization: data of the most critical CVE (maximum EPSS), total number of IPs, affected products, and CVE list.
IPS	Impact by IP: vulnerable product/version, details of the most critical CVE (maximum EPSS), and general list of CVEs affecting the IP.
CVES	Vulnerability catalog: individual details of each CVE (CVSS, EPSS, CWE, vendor/product, CISA) and the total number of organizations, IPs, ports, and ASNs affected by it.
ASES	AS statistics: count of vulnerable IPs and organizations, lists of cities and CVEs (including CISA KEV), along with CVSS/EPSS metrics.
PORT	Analysis by ports: list of CVEs and products on the port, CVSS/EPSS metrics, and the number of known exploits.
GENERAL RECORDS	General report per probe: IP, ASN, port, Organization, other banner information, as well as complete lists of CVEs, CVSS scores, EPSS, and CISA KEV metadata.

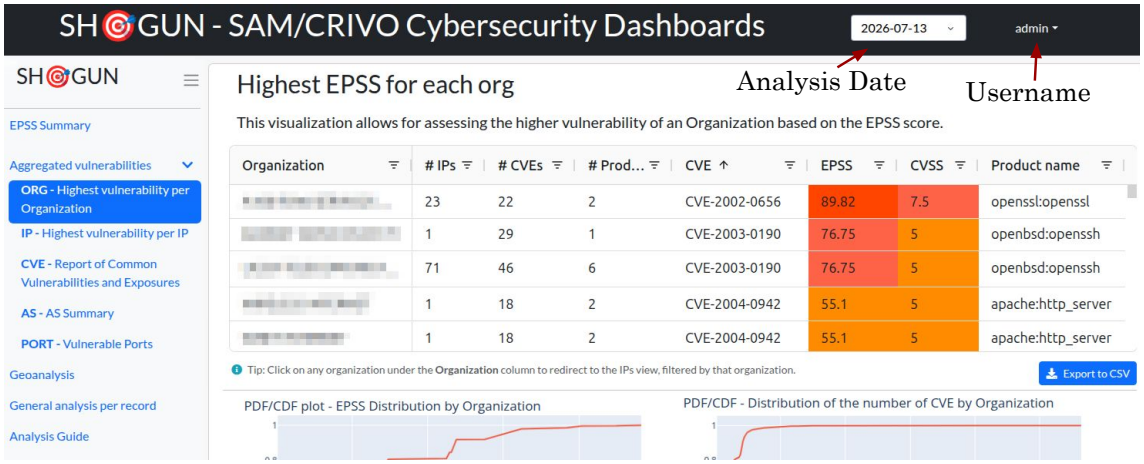

Figure 2. Example of a SHOGUN's view. In the figure, organization names were anonymized.

2.2. Visualization Interface

The front-end is built using the Dash framework, providing a modular environment for security analysis. Its architecture is organized into distinct views, each corresponding to one of the seven Delta tables described in Table 1, allowing analysts to navigate from macro-level infrastructure overviews to individual, probe-level records. An example of our interface is shown in Figure 2. This dashboard view displays the maximum vulnerability levels identified for each organization on a selected analysis date.

A key feature of the interface is its reactive behavior: when a user interacts with a table, such as by applying a filter, the underlying Pandas DataFrame is updated in real time, automatically triggering a refresh of all associated visualizations. Also, the table provides a search bar, allowing users to isolate records by specific field values, such as an IP address or an EPSS score threshold. When a filter is applied to focus on a particular subset of organizations, the corresponding graphs are immediately updated to reflect the

new criteria, enabling precise and granular forensic examination.

In each table provided by SHOGUN, specific fields act as interactive hooks. Clicking on these elements redirects the user to other relevant views, with the filtering context automatically propagated through the application's global state store. For instance, by clicking on an organization in Fig. 2, the page redirects to a second view tailored for IP-level data analysis, displaying only the hosts associated with the selected organization. Such a design allows security analysts to seamlessly transition from macro-level trends to individual endpoint details without manual reconfiguration. This cross-view navigation capability is demonstrated in practice in Section 5 through a concrete use case.

3. Setup

The tool provides a Docker Compose file in its repository to simplify the deployment process. If users choose to run the tool directly on the operating system, they must first ensure that the TLHOP-Library is installed from its official repository,[8] as well as install the Python dependencies. It is also necessary to configure PostgreSQL (responsible for storing user information, preferences, and saved items).

Table 2. Environment variables exposed in SHOGUN.

PARAMETER	DESCRIPTION
`CRON_EXPRESSION`	Specification of how often new dumps are checked
`SPARK_VCORES`	Number of VCPUs available for data processing
`SPARK_MEMORY`	Amount of RAM available for data processing
`SHODAN_FOLDER`	Directory where the Shodan dumps will be monitored
`RESULT_FOLDER`	Directory where the processing results will be saved
`TLHOP_DATASETS_PATH`	Directory used by TLHOP-Library to save auxiliary files

The tool's parameterization is presented in Table 2. By default, the `CRON_EXPRESSION` environment variable is set to "0 3 * * *" (meaning, each day at 3 AM). The Spark parameters are important to ensure adequate performance based on the available hardware resources. The last three parameters represent, respectively, the directories where the input data, output data, and auxiliary files (such as the NIST/NVD, CISA KEV, and FIRST EPSS datasets) will be saved and consumed.

4. Performance Evaluation

This section presents a comprehensive evaluation of the proposed tool's performance across several dimensions. In our experiments, we employed a Debian 12 server equipped with an Intel Xeon Gold 5318N processor (48 cores, 2.10 GHz) and 503 GiB of RAM. However, to simulate a more constrained and realistic environment, we restricted the Apache Spark (version 3.5.4) execution to 10 cores and 20 GiB of RAM. The evaluation of our approach utilizes Shodan data provided by Brazil's Center for Studies, Response, and Treatment of Security Incidents (CERT.br) under a partnership to map vulnerabilities within the Brazilian IPv4 address space. It is noteworthy that these data are publicly accessible via Shodan, meaning the level of information required to develop this work is subject to the same conditions available to any user of the platform.

[8]TLHOP-Library: https://github.com/lucasmsp/tlhop-library

The consolidated results of the experiments are presented in Table 3, which details the mean execution times along with input and output data sizes obtained from a 15-day experimental run (from July 1 to July 15, 2026). To ensure statistical representation, the results are accompanied by their respective 95% confidence intervals (CI).

Table 3. 95% confidence intervals (CI) for processing times and data sizes.

METRIC	MEAN	95% CI
Report (s)	159.2	(144.9; 173.5)
Aggregated (s)	29.2	(27.3; 31.2)
Total (s)	188.5	(172.4; 204.5)
Input (GB)	14.6	(12.9; 16.4)
Output (MB)	110.9	(93.7; 128.2)

On average, the pipeline processes 14.6 GB of input data per day (CI 95%: [12.9; 16.4] GB). Most of the processing time is consumed during the first stage (report generation). This stage is responsible for reading the raw JSON input files from the search engine and performing all necessary conversions, parsing, and cleaning. These operations include filtering Brazilian IP addresses, selecting records with identified vulnerabilities, and pruning unnecessary fields. Once this stage is completed, the resulting dataset, stored in a Delta format, is used by subsequent methods to generate auxiliary tables (represented by the "Aggregated" time in Table 3). Due to extensive filtering and the high compression efficiency of the Delta format, the generated output files are orders of magnitude smaller than the input, averaging only 110.9 MB (all output files combined). These optimized files enable the frontend to serve data rapidly using traditional Python web servers. Figure 3 and Figure 4 complement this analysis.

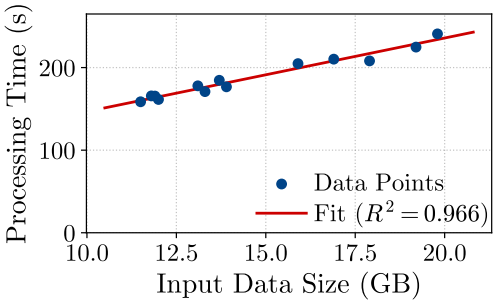

Figure 3. Linear regression of processing time versus data size.

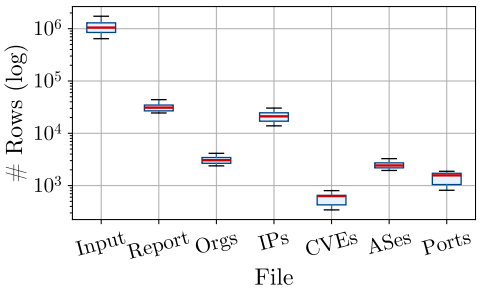

Figure 4. Box plot of row distribution.

Figure 3 illustrates the linear regression of the total processing time as a function of the input data size. This relationship is modeled by the equation $f(x) = 8.9x + 57.7$, yielding a high coefficient of determination (0.966), which indicates strong predictability. Conversely, Figure 4 presents a box plot detailing the distribution of data volume across the different stages of the experiment. We omit the EPSS summary data from this plot because its structure remains constant, always containing exactly 5 rows representing each EPSS probability tier (0–20%, 20–40%, 40–60%, 60–80%, and 80–100%). As shown in the box plot, a vast majority of the raw Shodan data is filtered out during the report generation phase. At the median, only 2.9% of the probed records actually correspond

to Brazilian IPs with confirmed or inferred vulnerabilities, translating to a median of approximately 31,000 records per day. Over the 15-day sample period, our system identified a daily average of 3,059 unique organizations with active vulnerabilities, spanning 21,082 unique IP addresses and 1,569 open ports. On median, 631 distinct CVEs were detected daily, with the vulnerable hosts distributed across 2,424 Autonomous Systems (ASes).

5. Demonstration and Discussion

The SHOGUN framework consolidates vast volumes of data on exposed devices, transforming complex information into actionable cybersecurity intelligence. For our demonstration, the system will run locally on the authors's laptop using the containerized infrastructure provided via Docker Compose. The planned demonstration will guide attendees through a practical workflow for vulnerability analysis and multidimensional data exploration, as illustrated in the following case study.[9]

Our demonstration begins with the "EPSS Summary" view (Fig. 5), which synthesizes global vulnerabilities categorized by their probability of exploitation (EPSS). For instance, when analyzing the July 13, 2026 data, most organizations fall into lower-risk brackets. Conversely, a critical subset within these metrics demands immediate attention: 103 active IP addresses, spanning 51 distinct organizations, exhibit an EPSS rank greater than or equal to 80%, signaling imminent exploitation risks. Furthermore, by toggling between different periods, such as between July 13 and July 14, users can quantify variations in the risk ecosystem. For example, one can observe an increase from 103 to 171 critical IPs, accompanied by a growth in the number of impacted organizations and associated CVEs. This quantification capability is fundamental for assessing the effectiveness of mitigation policies and promptly detecting newly exposed attack surfaces.

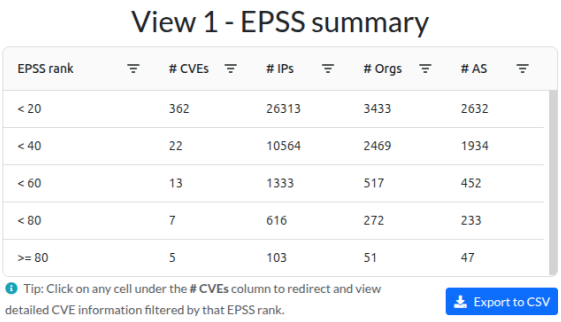

Figure 5. EPSS Summary view.

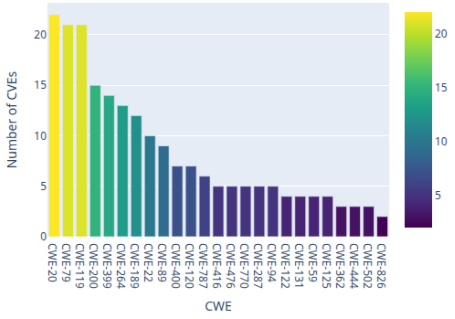

Figure 6. Number of CVEs by CWE category.

Building upon this macro-level overview, the SHOGUN framework enables a fluid transition to more granular investigations through its interactive design. For instance, by clicking on a specific cell within the "# CVEs" column in the table of Fig. 5, the user is automatically redirected to the "Report of Common Vulnerabilities and Exposures (CVE)"view (which leverages the corresponding dataset detailed in Table 1), pre-filtered according to the selected EPSS rank.

This interface acts as a centralized knowledge hub for each identified CVE. For example, on July 14, eight CVEs with an EPSS score $\geq$ 80% were detected by Shodan

[9]Our tool demonstration plan is also available at: https://youtu.be/-utJ5qEnlC4

within the Brazilian IP space. All of them are well-known vulnerabilities published between 2002 and 2006, such as `CVE-2006-3747`. Furthermore, to assist in identifying systemic security trends, the view includes nine comprehensive statistical visualizations, such as the distribution of CVEs by CWE shown in Figure 6. These encompass a variety of metrics, including scatter plots correlating EPSS and CVSS scores, line charts tracking the impact on organizational infrastructure, and bar charts highlighting the prevalence of specific weakness categories and ransomware campaigns.

Expanding the current analysis, clicking on the "# IPs" field for `CVE-2006-3747` redirects the user to a detailed view of affected assets, focusing on host IPs and their respective organizations (see Table 1). Within this view, clicking on any specific IP address affected by the targeted CVE redirects to the "IP Details" view, as illustrated in Figure 7. This page aggregates critical intelligence for the selected target, including geolocation data, CPE identifiers, and a full list of associated vulnerabilities complete with their respective EPSS and CVSS metrics. Furthermore, it provides technical context such as HTTP response headers, server information, and raw banner data. This level of granular visibility is essential for technical teams to validate exposure and determine the exact nature of the risk posed by an individual network endpoint.

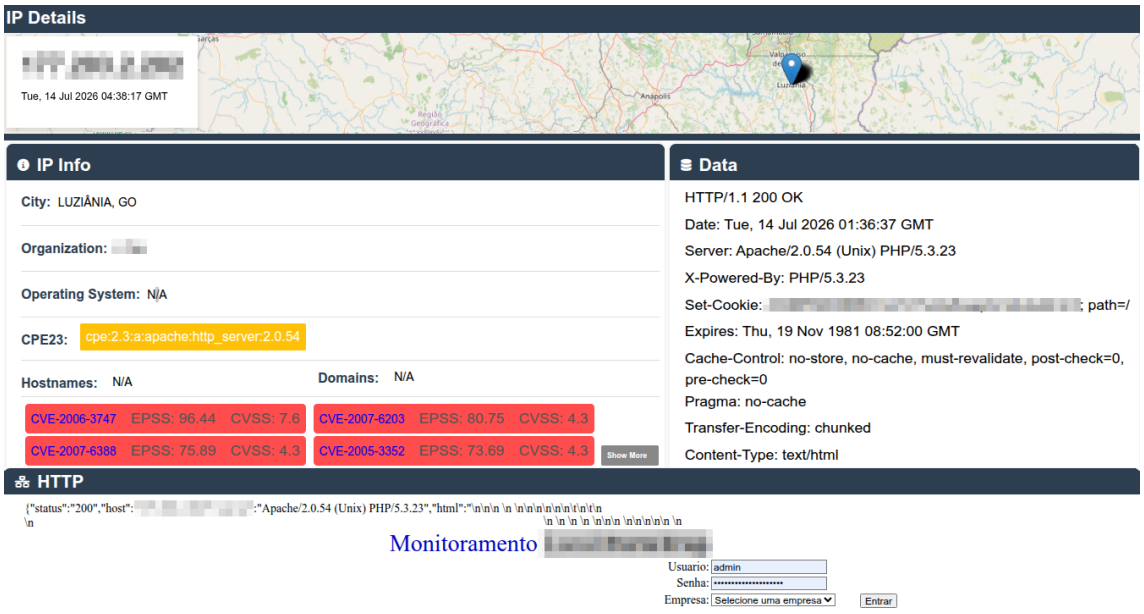

Figure 7. IP Details view. Sensitive information has been anonymized.

6. Conclusions and Future Work

In this work, we present SHOGUN, a dashboard tool for longitudinal vulnerability analysis utilizing search engine data. The tool allows network operators and security analysts to explore device scanning data, such as Shodan's, in a fast and interactive manner. We believe the SHOGUN has the potential to reveal insights regarding the security of a network space through its various tables, charts, and maps, while also enabling targeted analyses on specific IPs. The tool can be expanded in several contexts, such as by incorporating data from other search engines, adding new visualizations, further enriching information with external sources and the relationships among existing tables, as well as conducting additional studies to evaluate and enhance the system's usability.

AI Usage Statement

Artificial Intelligence tools were used exclusively for editorial support. All research decisions, methods, analyses, and interpretations were carried out entirely by the authors, without these tools influencing the scientific content or the conclusions of the work.

Acknowledgments

This work was partially supported by NIC.br, RNP, FAPEMIG, CNPq, CAPES, MASWEB, INCT-Cyber, and IAIA (INCT for AI).

References

Abuserrieh, L. and Alalfi, M. H. (2024). A Survey on Verification of Security and Safety in IoT Systems. *IEEE Access*, 12:138627–138645.

Huang, S.-C., Griffioen, H., van der Horst, M., Smaragdakis, G., van Eeten, M., and Zhauniarovich, Y. (2025). Trust but verify: an assessment of vulnerability tagging services. In *Proceedings of the 34th USENIX Conference on Security Symposium*, SEC '25, pages 823–842, Berkeley, CA, United States. USENIX Association.

Li, R., Shen, M., Yu, H., Li, C., Duan, P., and Zhu, L. (2020). A Survey on Cyberspace Search Engines. In Lu, W., Wen, Q., Zhang, Y., Lang, B., Wen, W., Yan, H., Li, C., Ding, L., Li, R., and Zhou, Y., editors, *Cyber Security*, pages 206–214, Singapore. Springer Singapore.

Mousavi, S., Khansari, M., and Rahmani, R. (2020). A fully scalable big data framework for Botnet detection based on network traffic analysis. *Information Sciences*, 512:629–640.

Ponce, L., Cunha, I., Matos, I., Ítalo Cunha, Fazzion, E., Hoepers, C., Steding-Jessen, K., Chaves, M., Guedes, D., and Jr., W. M. (2024). Arcabouço Multi-motor para Detecção de Vulnerabilidades na Internet Brasileira. In *Anais do XLII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 197–210, Porto Alegre, RS, Brazil. SBC.

Telesíntese (2026). Ransomware cresce 42% globalmente e amplia pressão sobre redes. URL: https://t.ly/2Tjui (visited on 2026-06-22).