Trust but Verify: An Assessment of Vulnerability Tagging Services

Szu-Chun Huang, Harm Griffioen, Max van der Horst,
Georgios Smaragdakis, Michel van Eeten, and Yury Zhauniarovich

Delft University of Technology

Abstract

Internet-wide scanning services are widely used for attack surface discovery across organizations and the Internet. Enterprises, government agencies, and researchers rely on these tools to assess risks to Internet-facing infrastructure. However, their reliability and trustworthiness remain largely unexamined. This paper addresses this gap by comparing results from three commercial scanners – Shodan, ONYPHE, and LeakIX – with findings from our independent experiments using verified Nuclei templates, designed to identify specific vulnerabilities through crafted benign requests. We found that the payload-based detections of Shodan are mostly confirmed. Yet, Nuclei finds many more vulnerable endpoints, so defenders might face massive underreporting. For Shodan's banner-based detections, the opposite issue arises: a significant overreporting of false positives. This indicates that banner-based detections are unreliable. Moreover, three commercial services and Nuclei scans exhibit significant discrepancies. Our work has implications for industry users, policymakers, and the many academic researchers who rely on the results provided by these attack surface management services. By highlighting their shortcomings in vulnerability monitoring, this work serves as a call for action to advance and standardize such services to enhance their trustworthiness.

1 Introduction

For over a decade, Internet-wide scanning tools have been used in industry and academia for attack surface monitoring and detection of vulnerable hosts. These tools typically employ two main approaches: one interprets banner data to identify vulnerable software versions, while the other uses crafted packets – containing specific payloads, custom headers, and other parameters – to directly probe hosts for vulnerabilities. The payload-based approach is seen as more reliable, but both techniques are used in practice. Services like Shodan [41], ONYPHE [35], LeakIX [26] and BinaryEdge [5], scan the Internet using these approaches and tag hosts with specific vulnerability labels, typically corresponding to CVE identifiers (Common Vulnerabilities and Exposures [34]). Later, network defenders – the consumers of these services – can run simple queries to find vulnerable Internet-facing hosts. For example, the United States Cybersecurity and Infrastructure Security Agency (CISA) employs the market-leader Shodan to monitor and assess Internet-facing network assets of other federal agencies and evaluate their vulnerability status [11]. Academic researchers have also relied on these services for collecting data (e.g., [18, 28]). For example, according to Google Scholar, Shodan is referenced in conjunction with CVE in around 1,500 research papers.

Given that many governments, companies and researchers use these services, it is important we gain a better understanding of the accuracy and completeness of their CVE detection. Earlier research has already exposed some of their limitations. They miss detections compared to more intrusive methods (e.g., Nmap scans [25]), privileged vantage points (e.g., Internet exchange traffic [3]), and more bespoke approaches (e.g., AI-based analysis of web interfaces [38]). Moreover, the absence of ground truth about which endpoints in the wild are actually vulnerable is another perennial problem for vulnerability scanning research. Our goal is to conduct an independent white-box measurement and compare the results to those of three industry scanners: Shodan, ONYPHE, and LeakIX. While they are a subset of all scanning services, they show up on most top-10 lists of scan engines for industry security researchers (e.g., [42]), so they represent a significant portion of the leading services in this market.

We will evaluate the banner-based and payload-based detections of those services by comparing them against each other and against our independent and synchronous payload-based measurements using Nuclei templates [32]. We repeatedly scanned 104,930 endpoints (IP:port pairs) that are all included in Shodan's scan results, so they are not blocking external scans. Our scans searched for 37 CVEs: 17 CVEs are included in Shodan's payload-based scans (tagged as 'verified' detections) and 20 CVEs are part of its banner-based scans (tagged as 'unverified' detections). While 37 CVEs might

seem like a small sample, the reality is that each service offers payload-based detections for only a limited set of CVEs. For example, Shodan tracks 38 CVEs and ONYPHE 54 CVEs, with only 8 in common. This limited overlap severely constrains how large the sample can be to compare the detections for the same vulnerability across different services. In sum, our 37 samples capture a significant portion of payload-based scanner results.

Shodan tracks a further 8199 CVEs and ONYPHE 64 CVEs using banner-based methods. Here, only 10 overlap, which again provides a small set for comparison. More importantly, these banner-based methods are reported by the services themselves as less reliable and thus are less suitable for evaluating accuracy. We do include them in the analysis, but do not claim that our small set is fully representative of the thousands of CVEs Shodan is tracking with banner-based methods.

We quantified the level of agreement among the results. Our Nuclei scans confirmed most of the payload-based detections from Shodan and ONYPHE. Yet, Nuclei detects many more vulnerable endpoints, raising concerns about false negatives. When comparing Nuclei to banner-based CVE detections, Nuclei finds that these are extremely noisy, to the point of being useless. Nuclei finds 95% of Shodan's banner-based detections to be false positives. Remarkably, Shodan implicitly confirms the unreliability of those detections, as for two CVEs it runs both banner-based and payload-based scans and gets *zero* overlap, confirming Nuclei's disconcerting findings. The community knows that banner-based detections are noisy, but our findings suggest the situation might be much worse than that. All in all, our findings raise substantial concerns about the accuracy of vulnerability-scanning services. In sum, we make the following contributions:

- We use Nuclei to conduct an independent evaluation of the vulnerability tagging performance of Shodan, ONYPHE, and LeakIX. We quantify the level of agreement among the four scanning solutions across 37 CVEs.
- For payload-based detections, we uncover large discrepancies. While there is a core set of agreed-upon detections, a much larger portion shows disagreement, highlighting substantial accuracy issues that affect enterprises relying on these services for attack surface monitoring and academic researchers using them to gather vulnerability data.
- For banner-based detections, we find that over 95% consists of false positives. Even using them as "starting points for further investigation", as Shodan describes, is highly questionable, since none of the payload-based detections fall inside the set of banner-based detections. This also suggests the widespread use of banner-based vulnerability detection in academic work is problematic.
- We complement our evaluation with a ground truth-based evaluation of 10 CVEs, exposing to the Internet Docker containers within three states: vulnerable, patched/mitigated, and non-vulnerable. We show that the precision of Nuclei scans reaches 100%, while Shodan and ONYPHE report only 86% and 80% correspondingly. At the same time, Nuclei's recall is only 79% – it misses 4 vulnerable hosts – while ONYPHE's result is perfect.
- During the ground truth experiment, we observed no crashes or service malfunctions as a result of Nuclei scans, supporting our earlier tests that verified templates are safe to employ.

2 Background and Related Work

There is a range of companies offering similar services for monitoring the attack surface of organizations [5, 9, 20, 26, 35, 50]. Typically, the data provided by these search engines includes general information about hosts (e.g., country, organization, etc.), which ports are open, what services are running on them, and whether certain vulnerabilities are present. While their toolchains are mostly proprietary, they rely on approaches that are also well-known in academic research on vulnerability scanning.

The first step is typically employing fast scanning tools to discover reachable hosts and open Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) ports [30]. During the second step, the service performs a full handshake either using the protocol based on the IANA-assigned services list [14] or by deducing the protocol dynamically [22]. Over the years, researchers [4, 45] have been attempting to understand the Shodan scanning behaviors since it mostly acts as a black box to users. They ran virtual machines worldwide and discovered that Shodan allocates different scanning ports and scanning priorities across its scanners. Each VM received, on average, 176 scans from Shodan per day [4]. Tundis et al. [45] ran 8 honeypots and observed that Shodan discovered all of their services within 31 days. They also pointed out that the longest scan interval between two scans is around 15 days. Bodenheim et al. [7] deployed four Internet-facing programmable logic controllers, and Shodan fully identified them within 19 days.

In the third step, the search engines analyze the obtained information with algorithms, which typically constitutes their "know-how". They annotate the hosts with *tags* or *annotations*, which can be later queried by users. Many of the services, including Shodan, provide information about possible vulnerabilities in the corresponding hosts.

Prior work developed two main approaches to getting from scan results to vulnerability detection: banner-based and payload-based methods.

Banner-based detection. This method employs the information from banners to extract the Common Platform Enumeration (CPE) [33] string or other identifiers, used to identify software and its version. This information is then checked against a list of CVEs associated with specific software versions. Some researchers collect the host metadata themselves,

and others leverage public scan data such as Censys [9]. Laštovička et al. [25] used NetFlow [13] and Nmap [29] to collect host information and create CPE labels for their university network. Then, they compared the CVEs inferred from CPE information with CVE tags from Shodan for the same target IPs. Shodan returned fewer CVE results than their scans.

There are some downsides for CPE-inferred CVEs, which may cause the CVE results to become inaccurate. For example, banner or service version information can form multiple CPE combinations due to different naming schemes among vendors and devices. Ushakov et al. [46] developed an algorithm to map software products to related CPE entries to obtain CVE lists and assess security risks. Sanguino et al. [37] proposed a tool to help users find suitable CPE strings for target software to decrease the potential wrong CVE mapping. Thomas et al. [44] studied vulnerabilities in Siemens Industrial Control Systems (ICS) related devices. They discussed the inconsistency of product and vendor information presented in CPE format and between CVE and CPE information. Among the selected 207 CVEs, there were 15% of CVEs had affected devices in their description but were not entirely in their CPE lists. This inconsistency may cause a false negative or positive for understanding any potential risk. Moreover, even a correct CPE may not be accurately associated with a CVE due to backporting. West et al. [48] used the Censys dataset [9] to observe OpenSSH software updates for enterprises. They pointed out that version information may not be accurate for measuring how outdated the software is since some of them may apply a backport version, i.e., a new patched software based on an old version.

Payload-based detection. Finding the existence of CVEs through actual payload exchange is more reliable than using CPE data. However, this CVE detection type is often highly specialized and can be impractical to use on a large scale [25].

Researchers studied various ways to detect vulnerabilities on the Internet through benign payload-based methods [16] [24] [2] [10]. Durumeric et al. [16] sent out crafted packets without payload or padding to discover the population of Heartbleed vulnerability in the IPv4 address space. Koot [24] studied the prevalence of *CVE-2019-11510* in the Netherlands. He tested the vulnerability by exploiting a crafted path. Antrobus et al. [2] developed a vulnerability scanner to identify multiple vulnerabilities ranging from weak cipher checks to Denial-of-Service tests. They conducted actual vulnerability checks, such as parsing URLs or testing the usage of unpredictable tokens. Gao et al. [10] studied vulnerability detection through Internet-wide scans for three CVEs. They used ZMap to detect open ports and sent special packets containing certain strings or objects to fingerprint the vulnerabilities. Similarly, Yu et al. [49] discussed Internet-wide vulnerability detection by sending special packets.

In our work, we do not rely on banner-based detections, but use the Nuclei toolchain [31] to conduct payload-based detections. Though prior work has compared scan results against Shodan, these were typically using different methods – more intrusive (e.g., Nmap scans [25]), using privileged vantage points (e.g., Internet exchange traffic [3]), and more bespoke approaches (e.g., AI-based analysis of web interfaces [38]). These are not apples-to-apples comparisons, however. These alternative approaches are not as scalable as banner-based and payload-based methods, so they are not substitutes. To the best of our knowledge, no prior work has compared different leading industry vulnerability monitoring services against each other and against an independent white-box measurement using the same type of toolchain.

3 Methodology

Figure 1 outlines the workflow of our study. There are two phases: (I) CVE Selection and (II) Vulnerability Scan. The goal of Phase I is to select a set of CVEs for comparison. This set is then scanned in Phase II.

Our goal is to compare three industry scanners (Shodan, ONYPHE, and LeakIX) to each other and to our independent scans using the Nuclei toolchain (more on this below). Selecting a representative sample of CVEs where we can compare the results is complicated, however. Each service tracks a different set of CVEs and the intersection of them is very small. Using payload-based methods, Shodan tracks 38 CVEs and ONYPHE 54 CVEs, with only 8 in common. Only 4 of these are tracked by LeakIX. In short, we cannot simply focus our analysis on the intersection of these three services, since the intersection is too small. An additional requirement is that we also need to be able to scan for those CVEs ourselves. This makes us dependent on the availability of Nuclei scanning templates for those CVEs and they are not always available. So this adds a fourth scanning solution to the intersection, making it even smaller.

Instead of focusing on the intersection of all three services plus the set of available scanning templates, we developed a broader sample of CVEs that allows us to make comparisons across two or more scanning solutions (meaning the three industry services and our own Nuclei setup). To bootstrap this process, we started with the market leader Shodan.

3.1 Phase I: CVE Selection

Step 1a. Select CVEs from Shodan: We employed the `stats` Application Programming Interface (API) provided by Shodan to extract all CVE-IDs tracked by the platform. In total, Shodan tracks 8,237 CVEs. Of these, just 38 are tagged as *verified*, so based on payload-based scans, while 8,199 are tagged as *unverified*, so based on banner data (see first row in Table 1).

Step 1b. Select CVEs from Nuclei: The next step is to check for which of these CVEs we can do our own scans. Our scanning toolchain is based on Nuclei [31], a fast and customizable

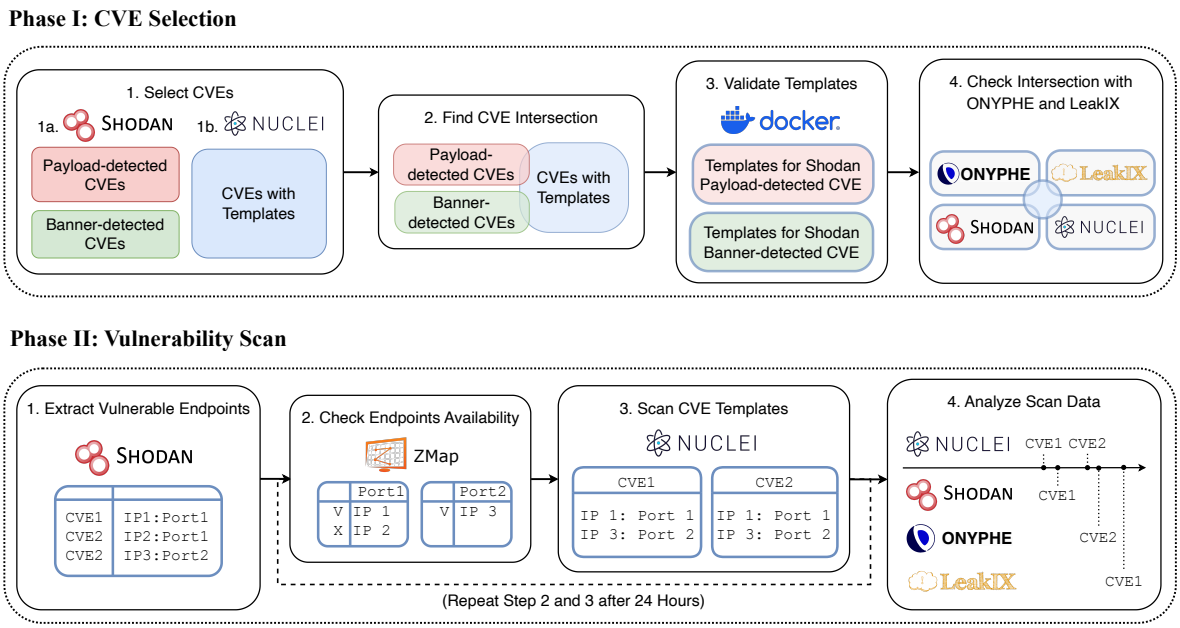

Figure 1: Research Workflow.

Table 1: CVE coverage by different scan types.

Category	Payload	Banner	Total
# Shodan CVE	38	8,199	8,237
# CVE with Templates	20	83	103
# Validated Templates	3	20	23
# Unvalidated Templates	14	0	14
# CVE Selected	17	20	37

vulnerability scanner. The procedure for performing a scan and analyzing the obtained data is specified in a *template*. Due to the simple YAML-based Domain Specific Language (DSL) used to develop templates, it has become very popular among security researchers. The templates are contributed by a large community of volunteers and stored in the official *nuclei-templates* repository [32]. We retrieved a list of all Nuclei Templates from the repository on May 10, 2024 and extracted the corresponding CVE-IDs from the *cves.json* file in the repository, which specifies the CVEs that these templates support. As of May 2024, the Nuclei repository contained 2,437 templates for detecting CVEs.

Step 2. Find CVE Intersection: We computed the intersection between both sets of CVE-IDs. There are 103 CVE-IDs present on both platforms. Of these, 20 CVEs are payload-based detections by Shodan, while 83 are banner-based (see the second row in Table 1).

Step 3. Validate Templates: Given that Nuclei templates are community-developed, it is crucial to validate their effectiveness in detecting the specified vulnerabilities. To accomplish this, we utilized Docker [15] to set up isolated, portable environments known as *containers* against which we can test the Nuclei templates. To speed up the process of finding relevant Docker environment specifications, we utilized the Vulhub [47] data. Vulhub is a GitHub repository that stores a collection of pre-built vulnerable Docker environments for specific CVEs. As of June 2024, it provided environment-building specifications for 204 CVEs.

Out of the 83 Shodan CVE detections based on banner data, the Vulhub project provides Docker environments for 25 of them. We deployed these 25 Docker environments and manually validated the performance of the Nuclei templates. A total of 6 Nuclei templates failed to confirm the presence of vulnerabilities in their respective vulnerable environments. We conducted a detailed investigation of these cases. For 4 CVEs, we identified the cause and made modifications to improve the templates' detection capabilities. For *CVE-2014-3704*, *CVE-2018-7600*, and *CVE-2019-3396*, we adjusted the headers either in the sent requests or in the matching conditions. In the case of *CVE-2018-18778*, we did not alter the template itself but instead modified the input format. The original template executed HTTP requests but did not revert to HTTPS when an error occurred. To ensure both protocols were checked, we manually adjusted the input file to specify both HTTP and HTTPS explicitly. The template for *CVE-2018-19518* failed to detect the associated vulnerability and no straightforward remediation was available, so we excluded it. Another template, for *CVE-2018-7602*, was excluded because it required knowing the username and password for the target service. So

this left us with 23 validated templates. We excluded an additional 3 templates (for *CVE-2022-0543*, *CVE-2022-24706*, and *CVE-2020-1938*) because they required additional interactions with target hosts, potentially resulting in indefinite scan durations. Ultimately, we selected 20 CVEs from the Shodan banner-based category, for which the Nuclei templates were deemed valid and effective.[1]

Among the 20 Shodan payload-based CVE detections, only 2 had corresponding Nuclei templates with compatible Docker environments available from Vulhub. To expand the number of templates we could validate, we explored official Docker repositories and identified suitable environments for additional 4 templates. We then tested these 6 templates. Among them, 4 passed the validation process. We excluded the template for *CVE-2023-33246* from the validated set because it required additional interactions with target hosts, potentially leading to indefinite scan durations. Notably, the *CVE-2015-2080* template failed due to an illegal character in the header field, intended to trigger the vulnerability. This illegal character prevented Nuclei from sending the correct packets. This issue has been reported to the Nuclei development team for further investigation. The remaining 14 templates are not validated because the associated software, e.g., Microsoft Exchange Server, is extremely difficult to set up in a Docker environment. However, we decided to include these CVEs in our final set, so as to have a broader comparison of payload-based CVE scans, even though we were unable to independently validate them. We assume their performance is comparable to the ones we were able to validate. In the end, for Shodan payload-based CVE scans, we selected 3 CVEs with validated templates and 14 with not-validated templates.

In conclusion, our final set of CVEs includes 20 Shodan banner-based CVEs and 17 Shodan payload-based CVEs. Table 4 in Appendix A provides a detailed overview of these 37 CVEs and their characteristics. The CVEs span a range of severity levels, with CVSS scores varying from 5.3 to 10.0, indicating a broad spectrum of risk from moderate to critical vulnerabilities. The average CVSS score among these CVEs is 8.9, reflecting a predominance of high-severity vulnerabilities that pose security risks across the affected endpoints.

Step 4. Check Intersection with ONYPHE and LeakIX: We further examined the intersection of the selected CVEs with those tracked by ONYPHE[2] and LeakIX[3], as detailed in Table 4. In total, 11 CVEs from our list intersect with those tracked by ONYPHE, while 8 intersect with LeakIX. And 7 CVEs are tracked by all four scanners.

[1]We enhanced the *CVE-2021-41773* template by adding a URL path based on Vulhub's checks. This modification enabled the detection of 200 more vulnerable hosts compared to the original template in our scans.

[2]The complete list of CVEs tracked by ONYPHE, along with their CVE selection policy, is available at `https://www.onyphe.io/docs/dorkpedia/vulnscan-cve-list`.

[3]LeakIX uses plugins to detect potential security issues and CVEs, we included the relevant plugin names capable of detecting these CVEs. The list of plugins is available at `https://leakix.net/plugins`.

3.2 Phase II: Vulnerability Scan

To collect data points to compare CVE detections of Shodan, ONYPHE, LeakIX, and Nuclei, we followed four steps.

Step 1. Extract Vulnerable Endpoints: We need to build a set of endpoints – where endpoints are defined as `IP:port` tuples – that have results in the industry scanners while also be scannable with Nuclei. We build our set of endpoints from recent Shodan scan results – thereby ensuring that Shodan can reach and scan these endpoints. If we were to choose random IPv4 addresses or networks, it would likely contain some IP space where Shodan is blocked by the network operator, while our ad hoc Nuclei scans might have normal access. This would bias the comparison against Shodan. Given that Shodan is the most well-known scanner, if it is not blocked for certain endpoints, we assume that the network administrators also did not block ONYPHE and LeakIX.

To build the set of endpoints, we used Shodan's `search` API to extract the set of endpoints that it had detected as vulnerable for any of the 37 selected CVEs in the first three weeks of July 2024, just before the start of our scanning period. Shodan performs scans irregularly. According to Tundis et al. [45], the longest scan interval is around 15 days. So we employed a slightly longer period of 21 days to maximize the number of endpoints that have an updated scan result.

Some CVEs are detected at a large scale: hundreds of thousands of vulnerable endpoints. For instance, we collected 956,543 `IP:port` tuples for *CVE-2017-15715* and 721,310 instances for *CVE-2021-40438*. As we want to perform our scans in a reasonable time, we restricted number of selected endpoints to 15,000, by randomly sampling them from all results for that CVE. Then, we combined all selected endpoints into a single set, obtaining a total of 105,232 entries. We excluded the entries with IPv6 addresses, since ZMap only accepts IPv4 addresses. This generates the final dataset containing 104,930 endpoints (unique IP:port combinations), which we call the *superset*. Across the set, there are 1,244 unique ports and 94,265 IPv4 addresses.

We observed a small proportion of honeypots within the superset.[4] Honeypot services can introduce bias when evaluating the safety level of a system, as their primary purpose is to study attacker behavior. However, since these services inherently present vulnerable environments, we opted to retain them, treating them as true positives for CVE detection scans.

Step 2. Check Endpoint Availability: We used ZMap [17] to scan the IPs and ports from the superset obtained in the previous step. We used the ZMap version that has not yet got the multi-port scanning functionality, therefore, we grouped our dataset by a port number and run the ZMap scan for each group and port number. We ran 15 instances of ZMap in parallel, limiting the speed of each scanning process to 1,000

[4]Out of the 69,369 IPv4 addresses collected for 17 Shodan payload-based CVEs, only one IP address was identified as a honeypot by Shodan.

packets per second. It took us approximately 14 minutes to check all the endpoints from our dataset. The results of the scan were merged into one resulting file in a random order.

Step 3. Scan CVE Templates: From July 28 to August 17, 2024, we scanned all endpoints obtained during the previous step with all 37 Nuclei templates. Since the maximum scan interval for Shodan is 15 days [45], our three-week collection period would ensure we would capture a fresh Shodan scan result for each endpoint to compare our scan results against. We applied these templates sequentially, meaning that at each point in time we only scanned one vulnerability for all available endpoints. We configured Nuclei to analyze 150,000 endpoints in parallel (async mode). It took 1.5 hours on average to complete the scan of all available endpoints for one CVE, and about 3 days and 4 hours to finish scanning for all selected CVEs. Once the scans for all 37 CVE templates were completed, we let our scanner rest for 24 hours before resuming the process from *Step 2*. This procedure was repeated 5 times during our data collection period. Note that our Nuclei scans did not lead to crashes or service malfunctioning in an additional experiment that we report in Section 5.

Step 4. Analyze Scan Data: Any scan is a snapshot in time. If the scan result of Nuclei is different from that of other scanners, this might be caused by a difference in timing. In between the scans, the situation at the endpoint might have been changed, e.g., it might be patched. To address this timing factor, we conducted consecutive Nuclei scans to ensure consistent results. By verifying that subsequent Nuclei scans produce stable outcomes, we minimize the likelihood that discrepancies with other scanners are caused by sudden changes at the endpoint itself. Following the completion of our Nuclei data collection, we retrieved scan data from Shodan, ONYPHE, and LeakIX for all endpoints in our superset, collected within the same time frame as the Nuclei scans. These datasets were then systematically compared to evaluate the level of agreement across scanners.

4 Measurement Results

In this section, we present the results of our measurements that aim to assess the reliability of CVE detections of scanning services, including Nuclei, Shodan, ONYPHE, and LeakIX.

4.1 Scan Performance

Each of the 104,930 endpoints in our set is scanned 5 times with all 37 CVE templates. Thus, in total, we collected 19,412,050 scan records. Figure 2 illustrates the statistics across these scans, visualizing the number of scanning records that have ZMap-detected not reachable results[5] and the Nuclei detection outcomes, which are categorized into cases where

[5]On average, each scan contains 28,250 unresponsive endpoints, accounting for 26.58% of the total IP addresses.

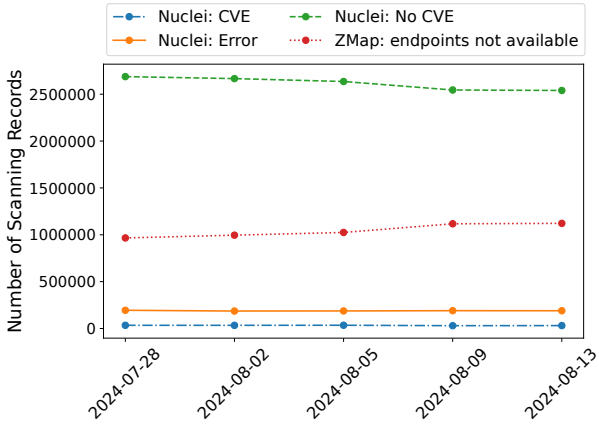

Figure 2: Summary of scanning records by scan.

some error occurred and cases with a clear outcome (CVE and No CVE).

We present the categorization of scan records for each CVE in Figure 3. This figure illustrates the overall composition of scan results for each CVE across the entire scanning period, including the number of records where ZMap identified endpoints as unresponsive, the instances where Nuclei returned errors (indicating indeterminate CVE status), and the records where Nuclei successfully completed the scan and returned a result of CVE detected or not detected. As shown in the figure, all CVEs share a very similar amount of ZMap "unavailable endpoint" records (141,251 scanning records). For the majority of CVEs, the predominant outcome category is "No CVE," with an average of 353,338 records per CVE falling into this category. The highest count of "No CVE" detected records is observed for *CVE-2020-7247*, with 383,310 records, while *CVE-2018-18778* has the fewest, with 69,933 records.

Interestingly, Nuclei error accounts for only a small fraction of all results, an average of 25,629 records per CVE. However, specific CVEs, such as *CVE-2018-18778* (a vulnerability in ACME mini_httpd) and *CVE-2020-5902* (a vulnerability affecting F5 BIG-IP-related products), exhibit a disproportionately large number of Nuclei error records, with 313,464 and 263,972 records, respectively. This indicates that Nuclei could not conclusively determine the presence of these CVEs due to issues such as fallback failures. *CVE-2018-18778* also encountered errors related to malformed HTTP responses or incorrect status codes, while *CVE-2020-5902* experienced errors associated with the failure to parse the response header. While the number of scan records indicating a detected CVE presence is relatively small compared to other categories, these records provide critical insights into the vulnerability landscape as observed by Nuclei. The most frequently detected CVE among our superset of endpoints is *CVE-2015-1635*, a vulnerability related to Microsoft Windows that carries a CVSS score of 10 (HIGH), underscoring its prevalence and severity. Of the 37 CVEs evaluated, seven have more than 10,000 scan records

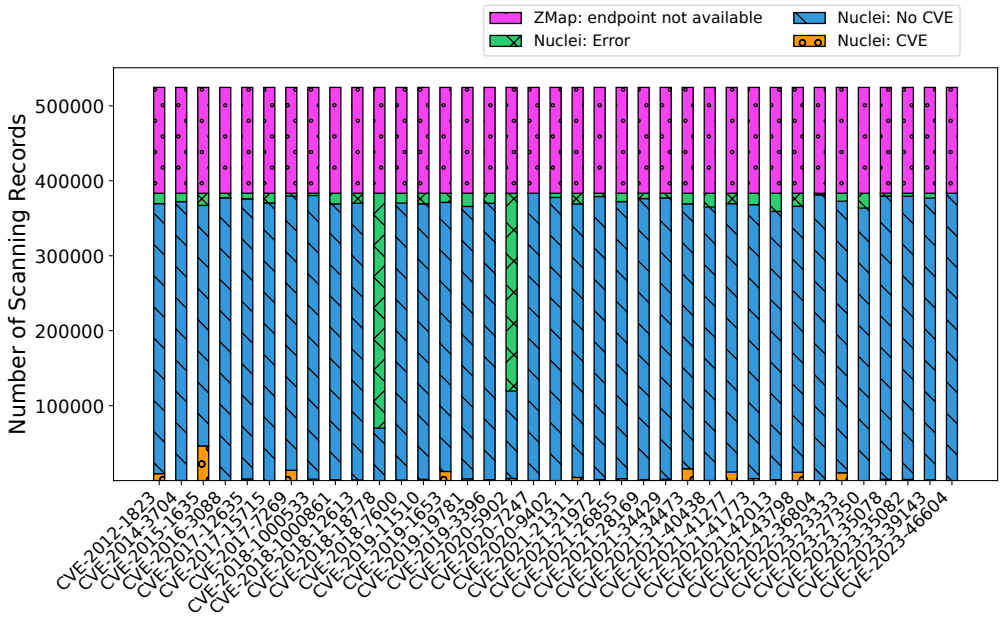

Figure 3: Categorization of CVE scan records by failure types and detection outcomes in ZMap and Nuclei scans.

indicating a confirmed CVE presence.

From the 19,412,050 scan records, we dropped the scan records with *ZMap: endpoint not available* and *Nuclei: Error*. Then, we conducted a detailed analysis for each CVE to determine the number of endpoints with consistent vulnerability detection results versus those with inconsistent results. Visualized in Figure 11 in Appendix C, the distribution of consistent and inconsistent results across CVEs shows that only a small fraction of endpoints exhibit inconsistencies. To ensure accurate analysis, we exclude endpoints with inconsistent results due to their fluctuating vulnerability status. Such variability makes it difficult to draw reliable conclusions or compare them with data from other scanning sources, as these endpoints may change their status over time, and different scanners may yield varying results. In total, we collected 13,196,431 scan records with consistent scan results for 81,681 endpoints.

4.2 Agreement between Shodan and Nuclei

We further queried Shodan for data on whether the endpoints from our superset were vulnerable to any of the 37 selected CVEs during the same period as the Nuclei scans (July 28 to August 17, 2024). To align Shodan's data format with that of the Nuclei scan results, we transformed Shodan's aggregated data, which tags multiple CVEs to a single endpoint, into a detailed dataset. Specifically, each endpoint with multiple CVE tags was expanded into individual scan records, each comprising the IP address, port, and a single CVE tag. In total, we obtained 40,087 scan records for 34,351 endpoints.

We now analyze to what extent our Nuclei scans agree with Shodan on the vulnerability status of an endpoint. We present the results for Shodan's payload-based scans separately from the banner-based ones since those methods are very different, and the latter is seen as less reliable.

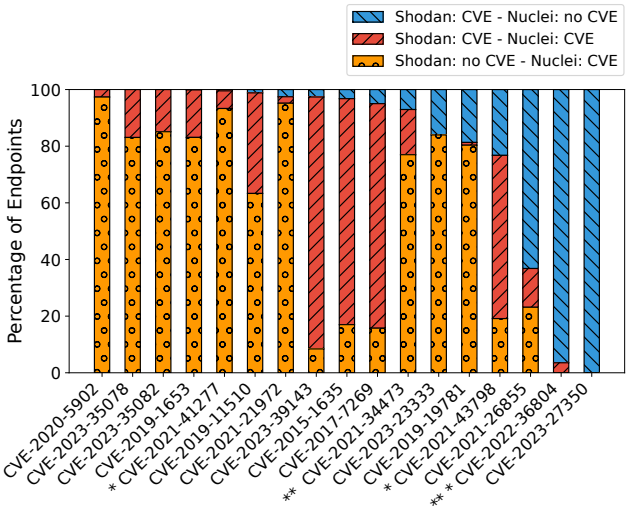

Figure 4: Percentage of endpoints classified as vulnerable or non-vulnerable by Nuclei and/or Shodan for payload-detected CVEs.

Figure 4 shows the Nuclei results for all Shodan's payload-based detections. The results show noticeable variation in agreement across different CVEs. Some CVEs exhibit high agreement between Shodan and Nuclei (areas marked in red).

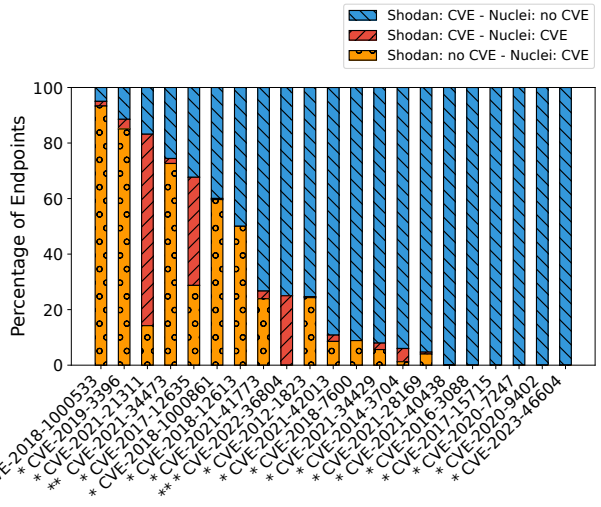

Figure 5: Percentage of endpoints classified as vulnerable or non-vulnerable by Nuclei and/or Shodan for banner-detected CVEs.

In others, Nuclei confirms the detections of Shodan, but finds many more vulnerable endpoints that Shodan missed (orange). Finally, for other CVEs Nuclei contradicts the detections of Shodan, finding no vulnerability (blue).

For 11 out of 17 CVEs (65%), we find that Nuclei does not substantially contradict the Shodan results, with less than 10% of the Shodan detections not confirmed by Nuclei. That said, for 10 CVEs, Nuclei finds many more vulnerable endpoints than Shodan: a factor of 2-36 more vulnerable endpoints depending on the CVE. Across all results, Nuclei reports 13,275 more detections than Shodan. This raises the question of whether Shodan verified results suffer from a large false-negative rate. Since there is no ground truth, we cannot ascertain that this is the case. The alternate explanation is that the Nuclei detections are incorrect.

These differences seem unrelated to whether the template is validated or not. There are three validated templates (marked by a single asterisk at the beginning of the CVE-ID). In the first case, Nuclei confirms Shodan's detections but discovers a factor of 10 more vulnerable endpoints (*CVE-2021-41277*). The second case reports a majority of consistent results (*CVE-2021-43798*), while the third case reports overwhelmingly contradictory results, where Nuclei disagrees for almost all endpoints with Shodan's detection for *CVE-2022-36804*.

We further investigate the underlying factors contributing to the varying levels of detection agreement among these three CVEs. For *CVE-2021-41277*, it is a vulnerability in Metabase with GeoJSON map support that allows potential local file inclusion via specific path queries, such as path with string: */api/geojson?url*. Similarly, *CVE-2021-43798* is a directory traversal vulnerability affecting Grafana services, detectable by probing paths including string: */public/plugins/<"plugin-id">/* (Nuclei template for this CVE uses "alertlist" as the plugin-id). Nuclei scans for these CVEs both rely on targeting paths associated with default service configurations. In contrast, detecting *CVE-2022-36804* in Bitbucket requires access to a public repository or read permissions for a private repository. As noted by Bitbucket Support, temporary mitigation involves globally disabling public repositories: "If you're unable to upgrade Bitbucket, a temporary mitigation step is to turn off public repositories globally..." [6]. Nuclei's template attempts to access the latest project within the targeted repository; however, the detection rate may be low because vulnerable services can be quickly mitigated by disabling public repository access. This makes confirming the presence of the vulnerability challenging, despite its high severity (CVSS 8.8). We speculate that the disagreement for this CVE is due to differing detection methods used by the two scanning tools.

Next, we compare Shodan's banner-based CVE detections with Nuclei detections (Figure 5). In this comparison, almost all Nuclei templates were validated (as marked by an asterisk at the beginning of each CVE-ID) except one template (*CVE-2021-34473*). The first thing we can see is that Nuclei disagrees with the bulk of the Shodan detections (marked by the blue area). Only for 3 out of 21 CVEs is there a meaningful degree of agreement, meaning that Nuclei confirms some portion of Shodan's banner-based detection: *CVE-2021-21311*, *CVE-2017-12635*, and *CVE-2022-36804*. For 18 out of 21 CVEs, Nuclei contradicts over 95% of the Shodan banner-based detections. While it is well-known that banner-based detections are less reliable, these findings suggest that they consist almost completely out of false positives.

There are two CVEs (*CVE-2022-36804* and *CVE-2021-34473*) where Shodan conducts both types of scans: payload-based and banner-based. So its search engine reports two separate sets of scan results for each CVE: one labeled as verified and one as unverified. Stunningly, we find *zero* overlap in endpoints between the verified and unverified results. So even Shodan's own scans contradict completely the unverified CVE detections.

Now, one might reasonably expect that banner-based detections are noisy. Indeed, this is why Shodan explains they "can have significant false positives [...] They should be seen as a starting point for further investigation". In other words, their value is that some fraction of them will be true positives. Following this logic, Shodan's own "verified" detections should fall within the set of "unverified" detections. Yet, as said, not a single banner-based detection contains the actual vulnerability, even according to Shodan's own scans. We see the same pattern for our Nuclei detections. Yet that is not what we see. The bulk of the Nuclei detections are not detected by Shodan's banner-based scans (the areas in orange), so they are not a subset of the banner-based detections. This means that the banner-based detections do not seem a useful "starting point for further investigation". For researchers using Shodan's unverified CVE detection as data, the problem is arguably even

worse. The doubts about data quality seem to render these detections unusable for scientific research.

Overall, banner-based detection appears to be highly problematic, likely due to several limitations inherent in using version information to identify CVEs. These errors can arise from inaccuracies in mapping banner information to CPE strings [37], incorrect associations between CPEs and CVEs [44], or backporting practices where patched services are built on older software versions [48], all of which contribute to inaccurate CVE identification.

Take-Away. In sum, for Shodan payload-based detections, defenders might face massive underreporting of vulnerable endpoints, while for Shodan banner-based detections they face massive overreporting. Given that payload-based methods are more reliable than banner-based, it seems the overwhelming majority of banner-based detections are false positives. Even Shodan itself confirms this implicitly, since for two CVEs their verified and unverified detections have no overlap whatsoever, which means their own results imply a 100% false positive rate. This corroborates our finding from Nuclei that banner-based detection is extremely unreliable. In the Discussion (Section 6), we will reflect on the implications of these findings for both security professionals as well as academic researchers relying on these detections.

4.3 Comparison with ONYPHE

We now extend our comparison by including ONYPHE [35], a competitor of Shodan. ONYPHE focuses on the CVEs in the CISA KEV (Known Exploited Vulnerabilities) Catalog [12]. This means they prioritizing CVEs that are exploited at scale. They employ both banner-based (`tag:vulnerableversion`) and payload-based (`tag:vulnerable`) methods to identify vulnerable endpoints, covering a total of 115 CVEs.[6] One important limitation to acknowledge is that the superset is derived from Shodan hits, i.e., endpoints scanned by Shodan for specific CVEs. However, ONYPHE conducts scans based on a substantially different set of CVEs, focusing on those deemed critical and actively exploited through their proprietary threat intelligence. Also, ONYPHE targets specific ports, resulting in limited overlap between their scans and Shodan's scans.

We plot the comparison between our Nuclei results and the ONYPHE results in the same way as in our Shodan comparison. Figure 6 presents the results. The pattern looks remarkably similar. For 8 out of 10 CVEs, our scans do not contradict ONYPHE's detections. Less than 5% of the detections are not confirmed by Nuclei. Yet, just as with Shodan, for those 8 CVEs Nuclei detects substantially more vulnerable endpoints: ranging from 1.1 to 104 times more (as measured by dividing the area marked in orange by the areas in red and blue). Also, for *CVE-2023-27350*, we find no detections in Nuclei, while ONYPHE found three. The same result occurred in the comparison to Shodan, where Nuclei found none, while Shodan found 51 vulnerable endpoints (Figure 4). This strongly suggests that this Nuclei template is not valid. Since this pattern is consistent across both Shodan and ONYPHE, it raises the question of whether our detection has a higher false positive rate. Perhaps a commercial provider is more likely to report conservatively and avoid false positives to its clients? Still,

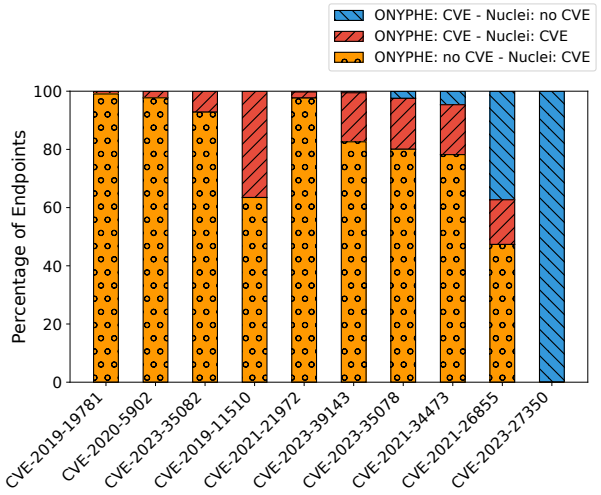

Figure 6: Percentage of endpoints identified as vulnerable or non-vulnerable by Nuclei compared to ONYPHE.

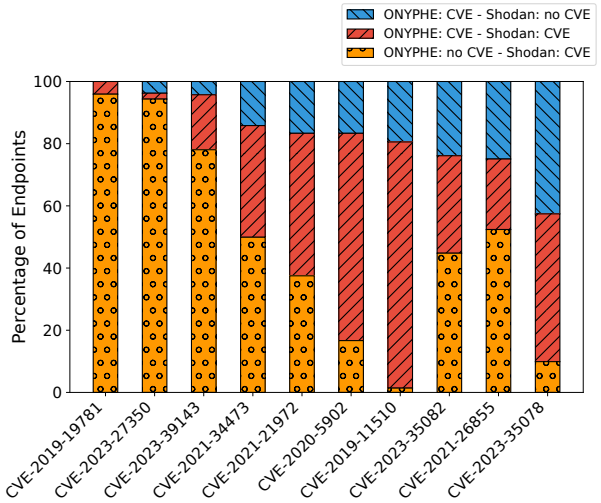

Figure 7: Percentage of endpoints identified as vulnerable by Shodan compared to ONYPHE.

[6] Unlike Shodan, ONYPHE performs weekly scans for their targeting 115 critical CVEs. We contacted ONYPHE and they generously shared their detection results with us, after we provided them with our superset of 104K endpoints and CVE list. These detections were collected in the same period where we performed our experiments: July 28 to August 17, 2024. This dataset contains a total of 16,714 detections across 1,053 endpoints. They contain results from both banner-based and payload-based vulnerability detection methods employed by ONYPHE. These are detailed in Table 4 in Appendix A.

since companies rely on these services for attack surface monitoring, our findings do confirm the urgency of investigating the potential for high false negative rates in future work.

We performed a direct comparison between ONYPHE and Shodan by analyzing all of Shodan's results alongside ONYPHE's CVE detection data during the same scanning period and against the same superset IP list. Both platforms identified scan results for 10 common CVEs. The results are depicted in Figure 7. Surprisingly, the same pattern emerges, where both services see a large amount of false negatives in the other service. For 8 out of 10 CVEs, both services agree on fewer detections than they disagree on. The portion of the detections that both services agree on (the red area) is a bit higher than the portion of detections agreed between Nuclei and each of the services separately, but disagreement is still the dominant pattern.

Take-Away. Overall, three datasets exhibit varying levels of agreement depending on the CVE and the detection methods. Even among the two commercial services, each sees a significant amount of false positives and false negatives in the other service. This raises serious concerns about the accuracy of vulnerability detection in attack surface monitoring services.

4.4 Comparison with LeakIX

To further enhance our analysis, we incorporated comparisons with LeakIX [26], a platform designed to identify leaks and security issues through its plugin-based framework. We queried all endpoints in our superset and filtered the results to include only scans conducted within the same timeframe as the Nuclei scans. This process produced a dataset consisting of 102 CVE detections spanning 46 unique endpoints.

We compare Nuclei with LeakIX (Figure 8), Shodan with LeakIX, and ONYPHE with LeakIX. As all comparisons exhibit similar trends, the detailed results of the Shodan-LeakIX and ONYPHE-LeakIX comparisons are provided in Appendix B. For all intersected CVEs, LeakIX reported CVEs for fewer than 20% of the endpoints. The average agreement of CVE detections between LeakIX and the other three scanning datasets is 1.86%. We attribute this low level of agreement to LeakIX's primary focus on detecting security flaws beyond CVE identification, which likely leads to fewer CVEs being reported than by other scan services.

Take-Away. Disagreement between scanners remains the dominant trend. As LeakIX primarily focuses on broader security risks, it generates relatively few CVE-related reports. Consequently, its level of agreement with the other three scan services is significantly low.

5 Ground Truth-Based Evaluation

This section evaluates the CVE detection performance of the four scan services – Nuclei, Shodan, ONYPHE, and LeakIX – using controlled environments as ground truth data.

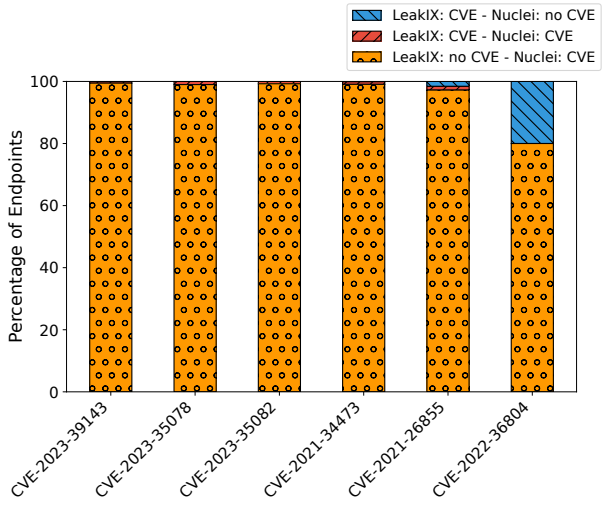

Figure 8: Percentage of endpoints identified as vulnerable by Nuclei compared to LeakIX.

5.1 Docker Environment Deployment

We selected 10 CVEs[7], comprising five Shodan payload-based CVEs and five Shodan banner-based CVEs. For each CVE, we deployed Docker containers with two vulnerable versions and two non-vulnerable versions. Additionally, for three CVEs, we included patched or mitigated versions that retain the same version information as their corresponding vulnerable versions. Specifically, for *CVE-2015-2080*, we included a patched version from the official release. For *CVE-2022-36804* and *CVE-2024-23897*, we followed official guidelines to apply mitigations to the vulnerable software, resulting in one non-vulnerable (mitigated) version for each. Note that *CVE-2021-41773* affects only one vulnerable version, *Apache 2.4.49*. Consequently, this version was included as the sole vulnerable case for the CVE. Since the default configuration of *Apache 2.4.49* is not impacted by the vulnerability, we also included this configuration as the non-vulnerable (with default setting) version. In total, we deployed 43 Docker containers of the corresponding software, comprising 19 vulnerable and 24 non-vulnerable versions. Table 5 in Appendix D lists the selected CVEs, along with their associated software versions.

Docker containers were deployed locally to evaluate Nuclei templates for one week. Simultaneously, the same containers were hosted on a separate machine with public IP addresses,

[7]Our selection prioritized CVEs detectable by at least two scan services for which we were able to run Dockerized environments. From 17 Shodan payload-based CVEs in our list, we included three CVEs that have buildable Docker environments. Additionally, we incorporated two Shodan payload-based CVEs—*CVE-2015-2080* and *CVE-2024-23897*—that were initially excluded due to Nuclei template malfunctions. These two CVEs were chosen because they have buildable Docker environments, and one (CVE-2024-23897) is tracked by all three commercial scan services. We further selected five CVEs from 20 Shodan banner-based CVEs in our list. Among these, *CVE-2023-46604* is also tracked by all four scan services.

enabling commercial scan services to scan them over the same duration. Each container was assigned a unique public IP address and configured to operate on default ports.

5.2 Performance Metrics Evaluation

Following the 7-day exposure of Docker instances, we collected scan data from each service and computed their performance metrics, including true positives (TP), false positives (FP), true negatives (TN), false negatives (FN), accuracy, recall, precision, and F1 score, as summarized in Table 2.

The evaluation for each scan service is based on its ability to provide accurate CVE detection for the CVEs that it actually tracks. In the Nuclei and Shodan cases, we evaluated them with 19 vulnerable and 24 non-vulnerable Docker instances for 10 CVEs they both track. Whereas since ONYPHE detects only 2 out of the 10 selected CVEs, the evaluation is only based on these CVEs, encompassing a total of 9 Docker instances. LeakIX was excluded from the evaluation as we were not able to identify the LeakIX probes in our infrastructure, yet it provided three detections for *CVE-2024-23897*. It did not detect any of the other CVEs however.

Table 2: Evaluation of CVE detection performance across Nuclei, Shodan, and ONYPHE.

Metric	Nuclei	Shodan	ONYPHE
True Positive	15/19	12/19	4/4
False Negative	4/19	4/19	0/4
True Negative	24/24	16/24	4/5
False Positive	0/24	2/24	1/5
No Detection	0/43	9/43	0/9
Accuracy	90.70%	82.35%	88.89%
Precision	100.00%	85.71%	80.00%
Recall	78.95%	75.00%	100.00%
F1 Score	88.24%	80.00%	88.89%

5.2.1 Nuclei

During the 7-day period, we executed Nuclei scans on local Docker instances every 10 hours. Simultaneously, we continuously monitored the Docker instances and network traffic to ensure stability, observing no crashes or service malfunctions as a result of these scans. Nuclei successfully identified 39 out of 43 Docker instances, with exceptions for the 4 vulnerable instances associated with *CVE-2015-2080* and *CVE-2024-23897* due to template malfunction. For *CVE-2023-46604*, Nuclei exhibited minor inconsistencies in detection. Specifically, for two of the vulnerable Docker instances, 5% of scan records reported no CVE found, while 95% accurately detected the CVE. Despite this inconsistency, we classify Nuclei's detection of these instances as True Positive results, as the error rate is negligible and does not significantly affect the overall evaluation. We report it because it does illustrate that even under relatively controlled conditions, stochastic events can cause incorrect scan results.

5.2.2 Shodan

During the experiment, we initiated on-demand scan requests twice: on the 2^{nd} day and on the 5^{th} day. For the 5^{th} day, for undiscovered endpoints, we specified IP addresses and ports. Alongside, we queried the scan results daily using Shodan's *host* search interface to monitor the detection status of each endpoint. The collected scan results were analyzed to evaluate Shodan's CVE detection performance throughout the experiment period. As shown in Table 2, Shodan correctly detected more than half of the vulnerable (12 out of 19) and non-vulnerable (16 out of 24) Docker instances.

For the two False Positive results, Shodan incorrectly assigned CVE detections to non-vulnerable versions that shared the same version information as the vulnerable versions. These errors occurred for one payload-based CVE (*CVE-2015-2080*) and one banner-based CVE (*CVE-2021-41773*). The incorrect detection for the payload-based CVE is particularly noteworthy, as it raises concerns about the reliability of CVE detection results provided by black-box commercial CVE detection mechanisms.

For the four False Negative cases, Shodan failed to provide correct CVE results for three banner-based CVEs (*CVE-2017-12635*, *CVE-2018-1000861*, and *CVE-2023-46604*) and one payload-based CVE (*CVE-2022-36804*). These failures occurred for two reasons: Shodan either failed to provide the correct CVE tags despite correctly identifying the software version information of the endpoints (for both banner-based *CVE-2017-12635* and *CVE-2018-1000861*), or it failed to detect the services running on the default open ports (for one banner-based *CVE-2023-46604* and one payload-based *CVE-2022-36804*). Although Shodan indicates it tracks these CVEs, the scanner did not reliably report the correct CVE information from the collected banner data or detect the appropriate services running on the open ports.

Regarding scan frequency, Shodan demonstrated an inconsistent scanning rate across endpoints, despite on-demand scans being requested uniformly for all endpoints. For example, among four Docker instances running Apache HTTP Servers (for *CVE-2021-41773*) on port 80, one endpoint was scanned five times, whereas the others were scanned only once or twice. In another case, we deployed nine containers with different Jenkins software versions (for *CVE-2024-23897* and *CVE-2018-1000861*) configured with ports 8080 and 50000 open. Over the 7-day period, port 50000 received a total of 28 scan results, while port 8080 received only 10. Notably, for three instances deployed for *CVE-2022-36804*, where Shodan does payload-based detections, the service consistently reported 'No information available' throughout the entire experiment. This emphasizes the importance of understanding the scanning schedule of the Shodan scanner and its

role in capturing the real-time attack surface on the Internet.

5.2.3 ONYPHE

As ONYPHE employs a network-neutral scanning approach for host discovery, it does not accommodate on-demand scan requests. Within the 10-CVE evaluation set, ONYPHE employs banner-based methods to detect the two CVEs (*CVE-2024-23897* and *CVE-2023-46604*). Among the 4 vulnerable and 5 non-vulnerable instances, ONYPHE correctly identified 8. Due to the use of banner-based detection methods, ONYPHE incorrectly identified the only non-vulnerable (mitigated) version of *CVE-2024-23897* as vulnerable.

Beyond the nine Docker instances included in the previous evaluation, ONYPHE also discovered eight additional instances and associated them with correct but different CVEs outside of our evaluation. For example, ONYPHE correctly identified the presence of *CVE-2024-23897* for three instances we deployed to test *CVE-2018-1000861*.[8]

5.2.4 LeakIX

We contacted LeakIX on the second day of our experiment to request scans of our endpoints, as their platform does not support on-demand scan requests. As detailed in Table 5, LeakIX tracks *CVE-2022-36804*, *CVE-2024-23897*, and *CVE-2023-46604*. However, the limited probing activity of LeakIX during our experiment period resulted in CVE detection results being available only for *CVE-2024-23897*. Specifically, it correctly identified two vulnerable instances and one non-vulnerable (non-mitigated) instance. Given the limited data points, conducting a comprehensive evaluation of LeakIX's CVE detection capabilities is not feasible. It highlights the significant influence of scanning activity levels on effective attack surface monitoring.

5.2.5 Take-Away.

Overall, the payload-based detection employed by Nuclei templates achieved the highest accuracy. Both Shodan and ONYPHE also demonstrated accuracy and precision rates exceeding 80%, effectively detecting the majority of vulnerable and non-vulnerable instances across our 43 deployed Docker instances. However, both Shodan and ONYPHE exhibited cases where banner-based CVE detection methods produced False Positive results, supporting our earlier observation that banner-based detection is unreliable. Of course, our small-scale evaluation cannot confirm that banner-based detections are sometimes so unreliable to the extent of producing only false positives.

[8] Shodan detected *CVE-2024-23897* for two instances associated with *CVE-2018-1000861*. However, Shodan did not provide CVE detection results for the remaining two instances, due to no scan data for port 8080.

5.3 Scanning Traffic Characteristics and Implications for CVE Detection

During the experiment, we received, on average, 6,767 packets per day from Shodan, and 19,225 packets from ONYPHE. The traffic is collected in raw PCAP files and filtered to contain only traffic from Shodan, ONYPHE, LeakIX, or the local Nuclei scans.[9] From the PCAP files, we extract all *tcp sessions*, which we use to identify requests and responses.

During the experiment, we counted the number of unique requests and replies made by a scanning service against our infrastructure. I.e., if we observe that a service uses a differently crafted request to test our infrastructure not spotted before, the corresponding number is increased.

Table 3 reports the results for Shodan, ONYPHE, and Nuclei,[10] showing a substantial difference in scanning behavior between Nuclei and Shodan/ONYPHE, especially in the number of unique request packets that we observe. While our infrastructure exposes multiple CVEs, the number of unique requests performed by Shodan and ONYPHE is low, and we did not observe any CVE-specific scanning behavior. For instance, for port 61616, Shodan only completed a TCP handshake and grabbed a banner without any further requests. This leads to misclassifications of services that appear vulnerable but are already patched, and vice-versa.

Shodan and ONYPHE mainly differ in how they handle certain ports. For port 8080, ONYPHE only requests the webpage and sends non-HTTP traffic afterwards and appears to focus on identifying responses that could indicate vulnerabilities tied to specific CVEs. In contrast, Shodan's methodology is strictly confined to HTTP path requests on port 8080. For port 80 we see the opposite, where Shodan sends non-HTTP requests. However, we cannot identify whether these requests are aimed at specific CVEs, or used in the fingerprinting of malware services. We identify for at least two of the requests, binary requests containing the string `Gh0st`, that is aimed at identifying a malware Command and Control server [39]. For Nuclei, we observe many requests that do either not adhere to a protocol specification or are very clear attempts to trigger a vulnerability (e.g., a path-traversal attempt). We do not see such clear evidence for other scanning services. On all ports apart from 80 and 8080, Shodan and ONYPHE have only sent requests to identify the service.

Take-Away. Overall, our analysis shows that Shodan and ONYPHE scale their scans through the entire Internet, and do not extensively probe individual hosts. The number of unique requests made to potentially vulnerable hosts is therefore limited, leading to more False Positive and False Negative

[9] We identified Shodan scans by mapping IP addresses to domain names associated with `shodan.io`. Scanning IP ranges for ONYPHE were obtained from their official website (`http://hina.probe.onyphe.net/ip-ranges.txt`), while LeakIX scanning IPs were sourced from their platform (`https://scan.leakix.net`).

[10] As LeakIX did not extensively contact our infrastructure, we omit the results for this service in our analysis.

Table 3: Unique requests (Req.) and replies (Rep.) per scanner over the duration of the experiment. LeakIX traffic is omitted because it only targeted one port at the time of the experiment.

Port	Nuclei		Shodan		ONYPHE	
	Req.	Rep.	Req.	Rep.	Req.	Rep.
80	113	88	8	14	3	18
3000	177	116	3	3	4	7
5984	83	133	2	4	3	10
7990	680	299	2	6	-	-
8080	1774	560	8	32	11	130
50000	-	-	5	19	1	1
61616	1	1	- *	1	1	1

*Shodan collected a banner for port 61616 and did not send any request payload.

classifications compared to Nuclei.

6 Discussion

Many countries are adopting more stringent cybersecurity regulations. In the EU, for example, NIS2 (Network and Information Security Directive 2) [19] requires patch management policies to be implemented by all medium to large-sized organizations operating in sectors deemed essential or important. This will increase the demand for vulnerability tagging services. Already, such services are very widely used by organizations to monitor their attack surface for vulnerability management programs.

One example is market leader Shodan's Monitor service, which lets defenders register their IP ranges to be monitored for security-relevant events, like the presence of CVEs. A related use case is when oversight bodies or sectoral CERTs (computer emergency response teams) rely on these services to support their constituents. In the US, CISA relies on Shodan and other tools for vulnerability scanning as part of the "Cyber Hygiene services" it offers to other federal agencies [11] [1].

While the promise of Shodan is to "gain complete visibility into what you have connected" [40], in practice the professionals relying on these services will expect some inaccuracy. Yet they have no way of gauging its performance. As far as we know, there has been no independent testing of the CVE detections of multiple industry attack surface monitoring services. Our study found different patterns for payload-based versus banner-based scans. Starting with the evaluation of the payload-based detections, we observed that our measurements confirmed the bulk of the detections of Shodan and ONYPHE, though less so for LeakIX. To illustrate: Only 10% of the Shodan detections were not corroborated by our findings. However, our scans found many more vulnerable endpoints than the other services – signaling the problem of false negatives. For example, compared to Shodan, for 10 out of 17 CVEs, we found a factor of 2-36 more vulnerable endpoints. This raises questions about the promised "complete visibility". Even if clients take that with a grain of salt, they are unlikely to expect this level of potential inaccuracy. For enterprise clients, the question is how critical false negatives are for their use case. If they rely on Shodan detections for keeping their attack surface secure, false negatives are omissions that leave vulnerabilities in place, until they are detected via other solutions – or are compromised. In terms of false positives, our analysis suggests that the rate is low for payload-based detection, which avoids burdening IT staff.

The situation is quite different for banner-based detections. For 18 out of 21 CVEs, Nuclei contradicts over 95% of the Shodan banner-based detections. Shodan says that these "unverified" detections are known to contain "significant false positives," yet their value is to serve as a "starting point for further investigation". Our results severely question this use case. Nearly all of our CVE detections fall outside of Shodan's banner-based detections. So investigating the noisy Shodan CVE tags won't lead you to discover the vulnerabilities in your attack surface. Shodan's own detections provide even stronger evidence for the extremely low quality of banner-based detections: we found zero overlap between the Shodan banner-based and payload-based detections for the same CVE. This means if you give these detections to your analysts as starting points for further investigation, then you are wasting scarce and costly resources. Of course, we should be careful when generalizing these specific findings, since we only analyzed small set of CVEs that were collected using banner-based methods. Shodan collects thousands more. While our findings are cause for concern and further investigation, it is likely that some banner-based detections are more accurate and might still have value.

We have focused our analysis on Shodan, but these patterns are not unique to that service. We confirmed them by an analysis of ONYPHE's CVE detections for the same set of endpoints. Even when comparing the two commercial services directly, each sees a significant amount of false positives and false negatives in the results of the other service. Our ground-truth evaluation also confirms that these services face the similar issues, with LeakIX being a bit of a special case operating with a different approach.

The limited overlap among the results of Nuclei, Shodan, ONYPHE, and LeakIX is reminiscent of a different area that has seen a lot of industry and academic effort: threat intelligence. Numerous studies have looked at the indicators of compromise (IoCs) that are provided by different commercial and free sources. A consistent finding in that literature has been that there is very little overlap among the IoCs detected by each provider [27]. This even holds true for the high-end market leaders who claim to be tracking the same threat actors groups. One study found an overlap of less than 4% in the IoCs detected by these firms [8]. In other words, every threat intelligence provider sees only a limited slice of the attacker ecosystem. This has led to an industry practice where enter-

prises feel the need to acquire, on average, seven different threat intelligence services, to have a bit more confidence in their coverage [36]. We certainly found significantly higher overlap among the CVE tagging services than is found among threat intelligence providers, but it does seem a single service might not be sufficient for adequate attack surface monitoring.

Overall, vulnerability monitoring services play an increasingly important role. The concerns that are raised by our findings are not meant to imply these services should be abandoned. They are indispensable in light of increasingly complex IT infrastructures and perennial problems like 'shadow IT'. Rather, we argue for improved transparency about performance and for awareness among practitioners and regulators about what these services actually provide. Our findings also argue for a strong commitment to improvement. Community-wide efforts to adopt state-of-the-art solutions and standardization will improve the quality of vulnerability tagging. Regulatory intervention may be required to give incentives to adopt best practices and improve the quality of vulnerability tagging, e.g., via certification under the EU's Cyber Resilience Act. Cyber insurance companies might also help establish which vulnerability tagging services are more accurate, based on their claims data around breaches.

Finally, our findings also serve to caution academic researchers. Hundreds, if not thousands, of papers rely on Shodan for detection of one kind or another. Though the CVE tags do not seem to be widely used in papers, our findings also question the use of banner metadata for CVE detection – which is a much more widespread practice (e.g., [43, 48]). There would be great value in a future study to better understand under what limited conditions banner-based detection is accurate enough for scientific purposes.

7 Limitations

Our study faces several limitations. First and foremost, we have no ground truth on the presence of the CVEs at the endpoints. This limitation is faced by all research on Internet-wide scans for vulnerabilities. Only direct contact with the administrators of the affected systems would get closer to ground truth [18], but that obviously does not scale. So, the best anyone can do is to provide white-box implementations of scanning methods that can be independently validated, replicated, and compared against the results of other toolchains.

A second limitation is that our evaluation is based on 37 CVEs. While this set contains variation in terms of the affected services and severity of the vulnerabilities (CVSS scores range: 5.3 — 10.0, average: 8.9), there may be other factors that become visible if the comparison is based on a larger set of CVEs.

Another limitation is that we can only speculate about why results for some CVEs were very similar between Nuclei, Shodan, ONYPHE, and LeakIX, while for others they were wildly different. We do know that small changes in the scanning templates can make a significant difference (Section 3.1). This issue is implicitly present in every vulnerability scanning study, though rarely surfaced. Future work might do sensitivity analysis to quantify the impact of certain changes.

Finally, payload-based scanning methods may not always accurately determine the absence of a CVE. Even if a scan does not detect a CVE, the vulnerability could still be present (e.g., the software is unpatched). However, other security measures, such as application firewall rules or network configurations, might prevent the payload from reaching the vulnerable code path. This potential discrepancy highlights that a "no CVE" result does not necessarily confirm the absence of a vulnerability but rather indicates that it was not detectable given the current conditions. Future work should assess the impact of these external factors on vulnerability scan assessments.

8 Conclusion

Enterprises, government agencies, and academics increasingly rely on attack surface monitoring services to assess the risk level of Internet-facing computing infrastructures. However, such reports are typically used as ground truth without scrutinizing their accuracy. In this paper, we perform independent experiments to assess the trustworthiness of such attack surface monitoring services. We compare the reports of a market leader in attack surface monitoring, the Shodan Search Engine, with the reports of our synchronous experiments that use carefully crafted Nuclei templates tailored to target requests based on specific vulnerability checks and payloads for a given vulnerability. Our analysis shows that for banner-based detections, Shodan users face massive overreporting. It is also noticeable that 52.07% of the vulnerable endpoints identified by our experiments were not reported by Shodan. Also, according to our study, for payload-based detections, Shodan users are massively under-reporting vulnerability hosts. Our work shows that the above-mentioned shortcomings do exist when comparing our results with different vulnerability detecting services, ONYPHE and LeakIX. Furthermore, discrepancies in CVE detection results among these commercial scanners, coupled with their inconsistent scanning frequencies, undermine the trustworthiness of their CVE reports.

Our findings have significant implications for industry users, policymakers, and security researchers, as they challenge the trustworthiness of vulnerability reports used for operational and regulatory decisions. With this study, we would like to make the different stakeholders aware of the limitations of current attack surface monitoring services and open a debate on concrete steps needed to advance and standardize such services based on best current practices and community efforts to make them more trustworthy in the future.

Ethical Considerations

This research adheres to the Menlo Report's ethical principles of Respect for Persons, Beneficence, Justice, and Respect for Law and Public Interest [23].

Respect for Persons: We respected individual privacy by scanning only publicly accessible systems without collecting personally identifiable information (PII). Our research project involves processing sensitive information about vulnerable systems. We obtained approval from our Institutional Review Board (IRB), which also required a Data Management Plan to ensure the data was stored on a secured server with access limited to the researchers involved.

Beneficence: To maximize benefits and minimize harm, we first examined all selected Nuclei templates and ruled out any presence of malicious exploits or otherwise harmful payloads. Additionally, during our ground truth experiment, we confirmed that Nuclei scans do not lead to crashes or service malfunctions of the containers running vulnerable and non-vulnerable software.

As it is infeasible for Internet-wide scans to obtain prior consent from system owners, we did adopt the prior research best practice of running a web page on port 80 of our scanning source IP address. The page allowed system owners to identify and contact us. The page explained our scan purpose and provided contact information for opting out of future scans. We did not receive any opt-outs or complaints.

Scans were conducted with minimal impact; on average, each endpoint is probed once every 90 minutes. While scans might trigger alerts that network operators have to deal with, we assess this impact as modest, since they experience thousands of daily scans. Also, our dataset consists completely of endpoints present in the Shodan search engine. In other words, the operators of these networks have not blocked or opted out of Shodan scans. Given that our scans are based on the same techniques as Shodan, we assume that our five scan runs do not add a substantial additional burden.

We considered notifying the owners of the endpoints we detected as vulnerable but that are not tagged as vulnerable in Shodan. In the end, we decided against it for two reasons. First, our findings show that each scan approach (Nuclei, Shodan, ONYPHE) yields substantially different results. In the absence of ground truth, it is not clear which of the Nuclei detections we can completely trust. Sending out potential false positives undermines the effectiveness of the vulnerability notification ecosystem as a whole. Of course, the issue of potential false positives affects Shodan and ONYPHE as well. But those services do not proactively send out vulnerability notifications. Network operators have to solicit this data from the service itself. Second, it is critical that vulnerability notifications are based on fresh scan results. Notifying about scans that are more than 24 hours old is already seen as bad industry practice and not helpful. We discovered the discrepancy between our detections and Shodan's detections more than a week after the scans were completed. To then notify the endpoint owners would require additional scans to obtain fresh results. This would increase the impact on the network, which we also wanted to minimize.

In the end, the benefit of this research is to support network operators, CSIRTs and others who rely on commercial scanning services by providing independent insights on their accuracy and value for network defense.

Justice: We ensured a fair distribution of risks and benefits by focusing on publicly accessible enterprise systems and disseminating findings to benefit the broader community, including researchers and system owners.

Respect for Law and Public Interest: While we followed best practices established in prior research, we agree with [21] that strictly speaking this kind of work operates in a legal 'grey zone' because of the many jurisdictions and unclear frameworks. Our research is set up to comply with the EU General Data Protection Regulation (GDPR). Some earlier jurisprudence has considered all IP addresses a form of personally-identifiable information (PII), but this has been superseded by new jurisprudence. Since we scanned for enterprise software on servers, we do not collect PII, except for some edge cases. The GDPR does provide legal grounds for collecting PII for statistical purposes, provided that the processing adheres to the principles of lawfulness, fairness, transparency, and data protection safeguards as outlined in the regulation.

Open Science Policy Compliance

In Section 3, we describe in detail our methodology. This knowledge can be used to obtain the initial set of vulnerable endpoints. So as we provide the exact list of CVEs and the exact dates of data collection, the interested readers can extract the same set. Although for some CVEs with a large number of vulnerable endpoints, we took a subset of them, we do not expect this to change our final results as the subsets were selected randomly. Unfortunately, we cannot share the exact set of endpoints used in this study because the corresponding hosts might be put in danger of being compromised.

This study used three public information sources about vulnerable hosts: *Shodan*, *ONYPHE*, and *LeakIX*. Given the set of picked endpoints and the selected dates, one has enough details to extract the information about the identified vulnerabilities there. We cannot share this set for the same reasons as in the previous case. Additionally, we share updated Nuclei templates and Docker environment files used in Section 5 as artifacts.[11] Using the described methodology, the interested readers can verify the correctness of these files. At the same time, it is impossible to validate the results of our scanning because of the agile Internet and the impossibility of performing the scan in the past.

[11] The updated Nuclei templates and Docker environment files are available at https://doi.org/10.5281/zenodo.14732150

Acknowledgments

We would like to express our sincere gratitude to Patrice Auffret, founder & CTO of ONYPHE, as well as Danny Willems and Gregory Boddin, founders of LeakIX, for their invaluable datasets and support throughout this work. This research was funded by the Dutch Research Council (NWO) via project THESEUS (grant nr. NWA.1215.18.006). It was also supported by the European Commission under the Horizon Europe Programme as part of the projects SafeHorizon (Grant Agreement #101168562) and RECITALS (Grant Agreement #101168490). The content of this article does not reflect the official opinion of the European Union. Responsibility for the information and views expressed therein lies entirely with the authors.

References

[1] America's Cyber Defense Agency. Cyber Hygiene Services. https://www.cisa.gov/cyber-hygiene-services, 2024.

[2] Rob Antrobus, Benjamin Green, Sylvain Frey, and Awais Rashid. The forgotten I in IIoT: A vulnerability scanner for industrial Internet of Things. In *Living in the Internet of Things*, IoT 2019, pages 1–8, 2019.

[3] Giovanni Barbieri, Mauro Conti, Nils Ole Tippenhauer, and Federico Turrin. Assessing the use of insecure ICS protocols via IXP network traffic analysis. In *International Conference on Computer Communications and Networks*, ICCCN '21, pages 1–9, 2021.

[4] Christopher Bennett, A Abdou, and Paul C van Oorschot. Empirical scanning analysis of Censys and Shodan. In *Workshop on Measurements, Attacks, and Defenses for the Web*, 2021.

[5] BinaryEdge. BinaryEdge: We gather data for you. https://www.binaryedge.io/, 2024.

[6] Bitbucket. Bitbucket Server and Data Center Advisory 2022-08-24. https://confluence.atlassian.com/bitbucketserver/bitbucket-server-and-data-center-advisory-2022-08-24-1155489835.html, 2022.

[7] Roland Bodenheim, Jonathan Butts, Stephen Dunlap, and Barry Mullins. Evaluation of the ability of the Shodan search engine to identify Internet-facing industrial control devices. *International Journal of Critical Infrastructure Protection*, 7(2):114–123, 2014.

[8] Xander Bouwman, Harm Griffioen, Jelle Egbers, Christian Doerr, Bram Klievink, and Michel van Eeten. A different cup of TI: The added value of commercial threat intelligence. In *USENIX Security Symposium*, pages 433–450, 2020.

[9] Censys. The Leading Internet Intelligence Platform for Threat Hunting and Attack Surface Management. https://censys.com/, 2024.

[10] Gao Chuan, Han-bing Yan, and Jia Zi-Xiao. Method for measuring the Internet devices and applications based on the features. In *International Conference on Computer Networks and Communication Technology*, CNCT 2016, pages 36–46. Atlantis Press, 2016.

[11] CISA. CY2021 administrative subpoena for vulnerability notification year in review. https://www.cisa.gov/sites/default/files/2023-01/CY2021_Admin_Subpoena_Summary_Factsheet_FINAL.pdf, 2024.

[12] CISA. Known Exploited Vulnerabilities Catalog. https://www.cisa.gov/known-exploited-vulnerabilities-catalog, 2024.

[13] Cisco. Introduction to Cisco IOS NetFlow - A Technical Overview. https://www.cisco.com/c/en/us/products/collateral/ios-nx-os-software/ios-netflow/prod_white_paper0900aecd80406232.html, 2012.

[14] Michelle Cotton, Lars Eggert, Dr. Joseph D. Touch, Magnus Westerlund, and Stuart Cheshire. Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry. RFC 6335, August 2011.

[15] Docker. Docker Builds: Now Lightning Fast. https://www.docker.com, 2024.

[16] Zakir Durumeric, Frank Li, James Kasten, Johanna Amann, Jethro Beekman, Mathias Payer, Nicolas Weaver, David Adrian, Vern Paxson, Michael Bailey, et al. The matter of Heartbleed. In *ACM Internet Measurement Conference*, pages 475–488, 2014.

[17] Zakir Durumeric, Eric Wustrow, and J. Alex Halderman. ZMap: Fast Internet-wide scanning and its security applications. In *USENIX Security Symposium*, pages 605–620, August 2013.

[18] Aksel Ethembabaoglu, Rolf van Wegberg, Yury Zhauniarovich, and Michel van Eeten. The unpatchables: Why municipalities persist in running vulnerable hosts. In *USENIX Security Symposium*, August 2024.

[19] European Commission. NIS 2 Directive. https://eur-lex.europa.eu/eli/dir/2022/2555/oj, 2022.

[20] FOFA. FOFA. https://en.fofa.info/, 2024.

[21] Florian Hantke, Sebastian Roth, Rafael Mrowczynski, Christine Utz, and Ben Stock. Where are the red lines? towards ethical server-side scans in security and privacy research. In *IEEE Symposium on Security and Privacy*, pages 4405–4423, 2024.

[22] Liz Izhikevich, Renata Teixeira, and Zakir Durumeric. LZR: Identifying unexpected Internet services. In *USENIX Security Symposium*, pages 3111–3128, August 2021.

[23] Erin Kenneally and David Dittrich. The Menlo Report: Ethical Principles Guiding Information and Communication Technology Research. *Available at SSRN 2445102*, 2012.

[24] Matthijs Koot. Field note on CVE-2019-11510: Pulse connect secure SSL-VPN in the netherlands. *Digital Threats: Research and Practice*, 1(2):1–7, 2020.

[25] Martin Laštovička, Martin Husák, and Lukáš Sadlek. Network monitoring and enumerating vulnerabilities in large heterogeneous networks. In *IEEE/IFIP Network Operations and Management Symposium*, pages 1–6, 2020.

[26] LeakIX. LeakIX. https://leakix.net/, 2024.

[27] Vector Guo Li, Matthew Dunn, Paul Pearce, Damon McCoy, Geoffrey M Voelker, and Stefan Savage. Reading the Tea leaves: A comparative analysis of threat intelligence. In *USENIX Security Symposium*, pages 851–867, 2019.

[28] Naif Mehanna, Walter Rudametkin, Pierre Laperdrix, and Antoine Vastel. Free proxies unmasked: A vulnerability and longitudinal analysis of free proxy services. In *Workshop on Measurements, Attacks, and Defenses for the Web*, pages 1–12, February 2024.

[29] Nmap. Nmap: the Network Mapper - Free Security Scanner. https://nmap.org/, 2021.

[30] Nmap. UDP Scan (-sU). https://nmap.org/book/scan-methods-udp-scan.html, 2024.

[31] Nuclei. Nuclei: Fast and customisable vulnerability scanner based on simple YAML based DSL. https://github.com/projectdiscovery/nuclei, 2024.

[32] Nuclei. Nuclei Templates. https://github.com/projectdiscovery/nuclei-templates, 2024.

[33] NVD. NATIONAL VULNERABILITY DATABASE: Official Common Platform Enumeration (CPE) Dictionary. https://nvd.nist.gov/products/cpe, 2024.

[34] NVD. NATIONAL VULNERABILITY DATABASE: Vulnerabilities. https://nvd.nist.gov/vuln, 2024.

[35] Onyphe. Onyphe: Cyber Defense Search Engine. https://www.onyphe.io/, 2024.

[36] Ponemon Institute. The Value of Threat Intelligence: Annual Study of North American & United Kingdom Companies. https://stratejm.com/wp-content/uploads/2019/08/2019_Ponemon_Institute-Value_of_Threat_Intelligence_Research_Report_from_Anomali.pdf, 2019.

[37] Luis Alberto Benthin Sanguino and Rafael Uetz. Software vulnerability analysis using cpe and cve. *arXiv preprint arXiv:1705.05347*, 2017.

[38] Takayuki Sasaki, Akira Fujita, Carlos H. Gañán, Michel van Eeten, Katsunari Yoshioka, and Tsutomu Matsumoto. Exposed infrastructures: Discovery, attacks and remediation of insecure ICS remote management devices. In *IEEE Symposium on Security and Privacy*, pages 2379–2396, 2022.

[39] Shodan. Malware Hunter. https://malware-hunter.shodan.io/, 2024.

[40] Shodan. Shodan Monitor. https://monitor.shodan.io/dashboard, 2024.

[41] Shodan. Shodan: Search Engine for the Internet of Everything. https://www.shodan.io/, 2024.

[42] SOCRadar. Top 10 search engines for pentesters and bug bounty hunters. https://socradar.io/top-10-search-engines-for-pentesters-and-bug-bounty-hunters/, 2024.

[43] Carlotta Tagliaro, Martina Komsic, Andrea Continella, Kevin Borgolte, and Martina Lindorfer. Large-scale security analysis of real-world backend deployments speaking IoT-focused protocols. *arXiv preprint arXiv:2405.09662*, 2024.

[44] Richard J Thomas, Joseph Gardiner, Tom Chothia, Emmanouil Samanis, Joshua Perrett, and Awais Rashid. Catch me if you can: An in-depth study of CVE discovery time and inconsistencies for managing risks in critical infrastructures. In *The Joint Workshop on CPS&IoT Security and Privacy*, pages 49–60, 2020.

[45] Andrea Tundis, Eric Marc Modo Nga, and Max Mühlhäuser. An exploratory analysis on the impact of Shodan scanning tool on the network attacks. In *International Conference on Availability, Reliability and Security*, pages 1–10, 2021.

[46] Roman Ushakov, Elena Doynikova, Evgenia Novikova, and Igor Kotenko. CPE and CVE based technique for software security risk assessment. In *IEEE International Conference on Intelligent Data Acquisition and*

Advanced Computing Systems: Technology and Applications, volume 1, pages 353–356. IEEE, 2021.

[47] Vulhub. Vulhub. https://github.com/vulhub/vulhub, 2024.

[48] Jonathan Codi West and Tyler Moore. Longitudinal study of Internet-facing OpenSSH update patterns. In *International Conference on Passive and Active Network Measurement*, pages 675–689. Springer, 2022.

[49] Miao Yu, Jianwei Zhuge, Ming Cao, Zhiwei Shi, and Lin Jiang. A survey of security vulnerability analysis, discovery, detection, and mitigation on IoT devices. *Future Internet*, 12(2):27, 2020.

[50] Zoomeye. Zoomeye. https://www.zoomeye.hk/, 2024.

A CVE Information

In this section, we present the 37 selected CVEs used in this study, as detailed in Table 4. For each CVE, we provide the CVE ID, CVSS score, application type, Nuclei template validation status, and the detection methods employed by each scanning service.

B Comparisons with LeakIX

This section presents the comparative scan results for Shodan with LeakIX in Figure 9 and ONYPHE with LeakIX in Figure 10.

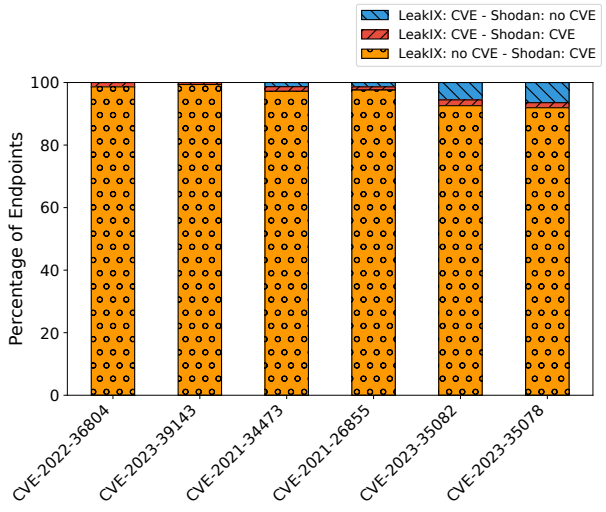

Figure 9: Percentage of endpoints identified as vulnerable by Shodan compared to LeakIX.

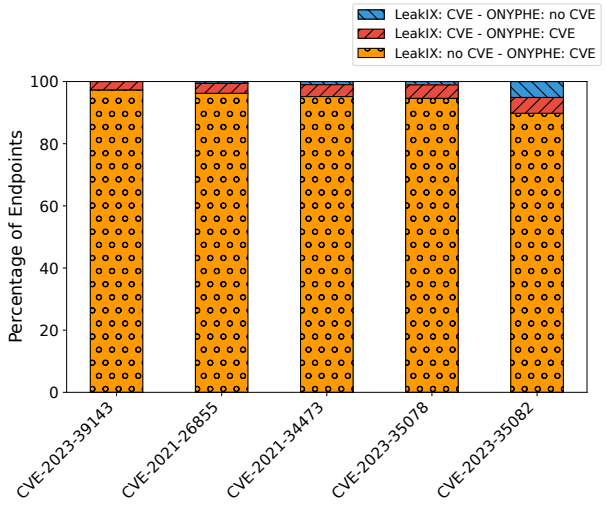

Figure 10: Percentage of endpoints identified as vulnerable by ONYPHE compared to LeakIX.

C Consistency in Nuclei CVE Detection

This section presents the distribution of consistent and inconsistent Nuclei scan results across CVEs in Figure 11. Notably, *CVE-2018-18778* and *CVE-2020-5902* exhibit the lowest number of consistently responsive endpoints, primarily due to a significant portion of their scan records being categorized as *Nuclei: Error* (see Figure 3). Among all the CVEs, the Nuclei templates for *CVE-2017-15715*, *CVE-2016-3088*, *CVE-2020-7247*, *CVE-2020-9402*, *CVE-2023-46604*, and *CVE-2023-27350* did not identify any CVEs.

D Docker Environment Information

In this section, we list the 10 selected CVEs used as ground truth data to evaluate the CVE tag performance of scan services, including Nuclei, Shodan, ONYPHE, and LeakIX, as summarized in Table 5. For each CVE, we provide details on the corresponding vulnerable and non-vulnerable software versions utilized in this assessment.

Table 4: **Selected CVEs Details.** *CVSS3* – CVSS 3 score (asterisks mean that CVSS2 score is used since the CVSS3 is unavailable); *Application* – vulnerable application/service; *Shodan Detection* – if the corresponding CVE is verified on Shodan (✔- verified, ✘- unverified); *Nuclei Detection* – if the corresponding Nuclei template is validated (✔- validated, ✘- not validated); *ONYPHE Detection* – CVE detection methods adopted by ONYPHE (★ - payload-based, ☆ - banner-based); *LeakIX Detection* – Plugin names used by LeakIX.

CVE-ID	CVSS3	Application	Shodan Detection	Nuclei Detection	ONYPHE Detection	LeakIX Detection
CVE-2015-1635	10*	Microsoft Windows	✔	✘	-	-
CVE-2017-7269	9.8	Microsoft Internet Information Services	✔	✘	-	-
CVE-2019-11510	10	Pulse Secure Pulse Connect Secure	✔	✘	★	-
CVE-2019-1653	7.5	Cisco Small Business RV320 and RV325 Routers	✔	✘	-	-
CVE-2019-19781	9.8	Citrix Application Delivery Controller and Gateway	✔	✘	★	-
CVE-2020-5902	9.8	F5 BIG-IP	✔	✘	★	-
CVE-2021-21972	9.8	VMware vCenter Server and VMware Cloud Foundation	✔	✘	★	-
CVE-2021-26855	9.8	Microsoft Exchange Server	✔	✘	★ ☆	ExchangeVersion
CVE-2021-34473	9.8	Microsoft Exchange Server	✔✘	✘	★ ☆	ExchangeVersion
CVE-2021-41277	7.5	Metabase	✔	✔	-	-
CVE-2021-43798	7.5	Grafana (Open Source)	✔	✔	-	-
CVE-2022-36804	8.8	Atlassian Bitbucket Server and Data Center	✔✘	✔	-	BitbucketPlugin
CVE-2023-23333	9.8	SolarView Compact	✔	✘	-	-
CVE-2023-27350	9.8	PaperCut	✔	✘	☆	PaperCutPlugin
CVE-2023-35078	9.8	Ivanti Endpoint Manager Mobile (EPMM)	✔	✘	☆	MobileIronCorePlugin
CVE-2023-35082	9.8	Ivanti Endpoint Manager Mobile (EPMM)	✔	✘	☆	MobileIronCorePlugin
CVE-2023-39143	9.8	PaperCut NG and PaperCut MF	✔	✘	☆	PaperCutPlugin
CVE-2012-1823	7.5*	PHP	✘	✔	-	-
CVE-2014-3704	7.5*	Drupal core	✘	✔	-	-
CVE-2016-3088	9.8	Apache ActiveMQ	✘	✔	-	-
CVE-2017-12635	9.8	Apache CouchDB	✘	✔	-	-
CVE-2017-15715	8.1	Apache httpd	✘	✔	-	-
CVE-2018-1000533	9.8	klaussilveira GitList	✘	✔	-	-
CVE-2018-1000861	9.8	Jenkins	✘	✔	-	-
CVE-2018-12613	8.8	phpMyAdmin	✘	✔	-	-
CVE-2018-18778	6.5	ACME mini_httpd	✘	✔	-	-
CVE-2018-7600	9.8	Drupal	✘	✔	-	-
CVE-2019-3396	9.8	Atlassian Confluence Server	✘	✔	-	-
CVE-2020-7247	9.8	OpenSMTPD	✘	✔	-	-
CVE-2020-9402	8.8	Django	✘	✔	-	-
CVE-2021-21311	7.2	Adminer	✘	✔	-	-
CVE-2021-28169	5.3	Eclipse Jetty	✘	✔	-	-
CVE-2021-34429	5.3	Eclipse Jetty	✘	✔	-	-
CVE-2021-40438	9	Apache HTTP Server	✘	✔	-	-
CVE-2021-41773	7.5	Apache HTTP Server	✘	✔	-	-
CVE-2021-42013	9.8	Apache	✘	✔	-	-
CVE-2023-46604	9.8	Java OpenWire protocol marshaller	✘	✔	☆	ApacheActiveMQ

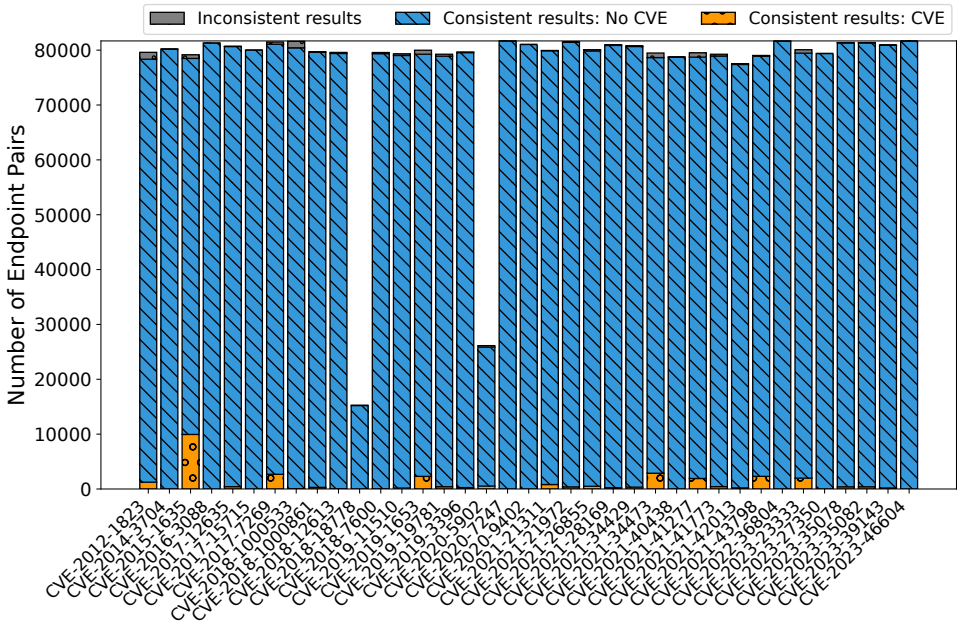

Figure 11: Classification of endpoints by consistent and inconsistent responses and their vulnerability status over time.

Table 5: **Docker-Buildable Software Versions for CVE Assessment.** The CVE detection methods employed by the scan services reuse the notations defined in Table 4. Versions marked with * are also affected by *CVE-2024-23897*, but excluded from CVE tag evaluation to simplify the analysis. † indicates mitigated vulnerabilities, and ‡ indicates patched vulnerabilities.

CVE-ID	Shodan Detection	Nuclei Detection	ONYPHE Detection	LeakIX Detection	Vulnerable Version	Non-vulnerable Version
CVE-2021-41277	✔	✔	-	-	metabase:v0.40.0 metabase:v0.40.4	metabase:v0.40.5 metabase:v0.40.7
CVE-2021-43798	✔	✔	-	-	grafana:8.0.1 grafana:8.2.0	grafana:8.0.7 grafana:8.2.7
CVE-2022-36804	✔✘	✔	-	BitbucketPlugin	bitbucket-server:7.0.0 bitbucket-server:7.7.0	bitbucket-server:7.6.17 bitbucket-server:7.17.10 bitbucket-server:7.0.0†
CVE-2015-2080	✔	✘	-	-	jetty:9.2.3.v20140905 jetty:9.2.7.v20150116	jetty:9.2.30-jre8-openjdk jetty:9.3.30-jre8-openjdk jetty:9.2.3.v20140905‡
CVE-2024-23897	✔	✘	☆	JenkinsVersionPlugin	jenkins:2.441 jenkins:2.426.2	jenkins:2.442 jenkins:2.426.3 jenkins:2.441†
CVE-2021-21311	✘	✔	-	-	adminer:4.7.8 adminer:4.7.7	adminer:4.7.9 adminer:4.8.1
CVE-2017-12635	✘	✔	-	-	couchdb:2.1 couchdb:1.6.1	couchdb:3.4.2 couchdb:3.1.2
CVE-2023-46604	✘	✔	☆	ApacheActiveMQ	activemq:5.18.0 activemq:5.18.2	activemq:5.18.3 activemq:5.18.6
CVE-2018-1000861	✘	✔	-	-	jenkins:2.138.1* jenkins:2.138.2*	jenkins:2.426.3-lts jenkins:2.427*
CVE-2021-41773	✘	✔	-	-	httpd:2.4.49	httpd:2.4.62 httpd:2.4.57 httpd:2.4.49 (Default)