One Does Not Simply Score a Website: Evaluating Website Security Scoring Algorithms

Daan Vansteenhuyse
DistriNet, KU Leuven
3001 Leuven, Belgium

Victor Le Pochat
DistriNet, KU Leuven
3001 Leuven, Belgium

Gertjan Franken
DistriNet, KU Leuven
3001 Leuven, Belgium

Lieven Desmet
DistriNet, KU Leuven
3001 Leuven, Belgium

firstname.lastname@kuleuven.be

***Abstract*—With the increasing importance of cybersecurity on the Web, an uptake in the use of scoring algorithms arises which combine security properties of a website to a single score. These algorithms make it convenient to compare security postures of websites and are used in settings like risk assessment. In this paper we present a comparative analysis of three scoring algorithms by employing them on the most popular domains and find several shortcomings. We find that the investigated scoring algorithms give higher scores to more popular websites yet lack agreement on what the most secure sites are. Additionally, we show that it requires minimal effort to fool a security algorithm into providing a high score to an unprotected website, allowing them to appear more secure. Our analysis reveals that 1,019 sites either intentionally or unintentionally inflated their security score. We list recommendations that could prevent such score inflation like score limitating or content-depended scoring.**

***Index Terms*—scoring algorithm, web security, security scoring**

1. Introduction

As the Web continues to grow and attract more users [15], cybercriminals increasingly exploit it for malicious activities. In response, companies are combatting this by continuously improving their cybersecurity defenses. While companies can hire external penetration testers or white-hat hackers for this purpose, publicly available automated tools offer a more accessible alternative.

One type of automated tools are security scoring algorithms. These algorithms quantify a website's security by assigning a score to it. Security scoring algorithms are widely used in both industry and literature to compare websites with each other. We see those algorithms being used to quantify risk [17], [19] but also to provide a general security overview of websites [8], [14]. It can give feedback to developers showing potential gaps in their security but also compare their security with other websites [8], [16].

Scoring algorithms allow ordering websites based on employed security measures, facilitating convenient comparison. This enables businesses, for instance, to assess the security of various software vendors and make informed decisions on whether to adopt their software if their security underperforms [19]. This indicates potential high impact of security scoring algorithms. Despite this, these algorithms have not been studied in the literature.

In this paper, we conduct a comparative analysis of three different scoring algorithms to shed light on this ecosystem: ImmuniWeb [9], SSL Labs [16] and an algorithm based on CWSS [18]. We investigate how these scoring algorithms operate by investigating two parameters. Specifically, we study the correlation between a website's popularity and its security score. Additionally, we investigate the agreement among the highest-scoring websites across different algorithms.

Furthermore, we demonstrate that it is possible to manipulate a website to achieve an inflated security score despite having minimal protection. This challenges the reliability and effectiveness of these scoring algorithms.

To summarize, the main contributions of this paper are the following:

- We perform a comparative analysis of three web security scoring algorithms, focussing on two aspects: The similarity between a website's popularity and security score, and the level of agreement among scoring algorithms.
- We demonstrate how a scoring algorithm can be manipulated to assign a high security score to an unprotected website. Additionally, we find websites in the wild that, possibly unknowingly, use scoring inflation techniques to gain a higher score.

2. Background and related work

Security scoring algorithms all score websites using their own method, yet they all function by investigating the presence and/or configuration of different security features. The final score is an aggregation of subscores of various security features. This aggregated score is obtained by combining all subscores using certain heuristics, most commonly the sum of all subscores. Moreover, most security scoring frameworks sort nominal scores into letter-designated buckets. For example, a website scoring 100 could be given the letter A, indicating a good security posture. An insecure website could be assigned a letter F to indicate further improvement is needed.

Table 1 and the remainder of this section provide an overview of all security scoring algorithms discussed in this paper and the different checks they employ to calculate a score.

Both industry players and researchers in literature have developed scoring algorithms, not always limited to only

DOI 10.1109/EuroSPW67616.2025.00028

TABLE 1. EXISTING SCORING ALGORITHMS AND THE SECURITY CHECKS THEY USE TO CALCULATE A SCORE

	Security Scorecard [17]	Vanta [19]	**ImmuniWeb [9]**	HTTP Observatory [14]	Security Headers [8]	**CWSS based** [18]	CSP Evaluator [20]	**SSL Labs [16]**
Exploitable CVE	x	x	x					x
Presence security headers	x		x	x	x	x		
Content security headers	x		x	x			x	
DNS records	x							
Legal Compliance		x						
Cookie security			x	x		x		
TLS/SSL setup								x

the security of a website. One of the industry providers of such an algorithm is **Security Scorecard** [17]. They calculate a rating to reflect an entire organization's security posture. This helps with determining the likelihood an organization is sustaining a breach. This can be used both for self-evaluation and for assessing the security posture of suppliers. Similarly, **Vanta** specializes in vendor risk management which assesses third-party dependencies and assigns risk scores to them [19]. This is used by organizations to monitor their suppliers and gain insights in potential security issues that could harm their own software. Based on risk scores assigned by Vanta, a company might decide to not include insecure dependencies in their own software.

ImmuniWeb developed a web security testing platform that assigns a score to a website based on their general security posture [9]. They run over 3 million website tests yearly. ImmuniWeb's testing platform evaluates a website by looking for the employment of certain security properties such as cookie attributes and HTTP headers. Starting with a score of 100, the algorithm adds or subtracts subscores based on the presence of certain properties. Finally, this aggregated score is reduced to a letter bucket for easy comparison. Additionally, ImmuniWeb provides clients with recommendations to enhance their website security and offers insights into how their website compares to others.

A similar method is applied by Mozilla's **HTTP Observatory** [14], which reduces the given score derived from subscores to a letter bucket. They aim to help developers strengthen their web security by analyzing their header configuration and cookies and did so for 6.9 million websites since its release in 2016.

Security Headers also checks for HTTP header configuration, but does not take cookies into account [8]. This project, developed by Scott Helme and now part of Probely, only checks the security headers of a website and assigns a score based on the presence of certain headers.

Another nameless scoring algorithm is designed by Van Goethem et al. for their large-scale security analysis, which focussed on the security posture of popular European websites [18]. Here, the researchers developed a security scoring algorithm **based on the Common Weakness Scoring System (CWSS)** to facilitate website comparison [13]. Leveraging the CWSS framework, they assigned scores to security mitigations such as HSTS. Similar to Security Headers, HTTP Observatory and ImmuniWeb, this algorithm assigns subscores when a certain security property is present. The final score is determined by the sum of all subscores.

Whilst the previously discussed scoring algorithms focus on the general security of a website, some algorithms only focus on a specific aspect of a website. One such scoring method is **CSP-evaluator**, a publicly available tool developed by Weichselbaum et al., to assess how well a website's the Content Security Policy (CSP) protects against Cross-Site Scripting attacks (XSS) [6], [20]. They analyze the employed CSP and give advice to the developer on how to improve it. A CSP is a policy to be set by a website that controls what resources may be loaded and from where they can originate. Each fault that is found is awarded a severity ranging from $HIGH$ to $INFO$, where $HIGH$ indicates a known vulnerability to bypass the employed CSP and $INFO$ indicates a deprecated directive or minor fault.

SSL Labs is a scoring algorithm developed by Qualys that investigates the secure connection made by a client to the web server [16]. It helps developers strengthen their connection security by giving concrete advice on how to improve. Qualys tests the connection based on three properties: Protocols, Encryption keys and Ciphers. For each property the assigned score is the mean of the most and least secure aspect. This means that if a website supports both SSL 2.0 and TLSv1.2 the protocol score will be lower than a website that only supports the newer TLSv1.2 protocol. The final score is a weighted average of all subscores. Qualys runs around 134,000 tests each month. To investigate this scoring algorithm we use an open-source implementation called `testssl.sh` [21].

3. Methodology

In this section, we elaborate on the selected scoring algorithms for our comparative analysis and define the properties to be studied. Finally, we discuss the details related to our experimental setup.

3.1. Scoring algorithms

The evaluated algorithms are highlighted in bold in Table 1: ImmuniWeb, the CWSS-based algorithm and SSL Labs. The reason for picking these algorithms is twofold. First they were selected for the diversity they offer. Both ImmuniWeb and the CWSS-based algorithm study the entire security posture of a website, with the former originating from industry and the latter academic research. SSL Labs varies in how it only focuses on one security aspect. Secondly these algorithms were picked due to their availability of information and accessibility.

Figure 1. Cumulative distribution function of scores of all three scoring algorithms

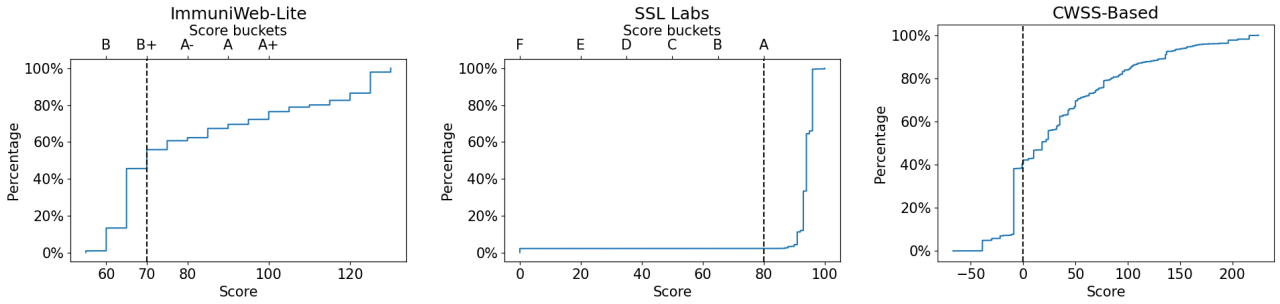
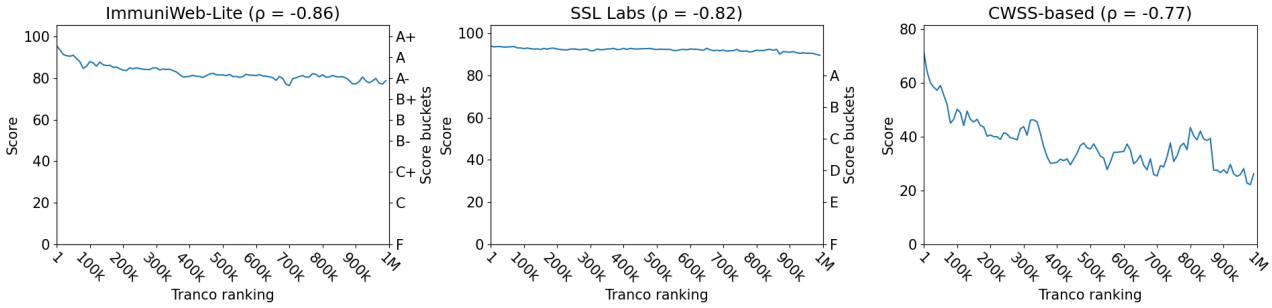


Figure 2. Security score in function of the popularity as defined by its Tranco ranking showing that popular domains on average get a higher score.



3.2. Ranking properties

The properties used in our analysis were chosen to gain insight in how security scoring algorithms behave. The properties were partly inspired by previous research on toplists [11], [22].

The first property we study is the relation between security and popularity. Since more popular websites attract more visitors, they most likely have more resources to invest in security. Hence, we expect more popular websites to be scored as more secure.

Secondly, we investigate the agreement between the scoring algorithms to find out how similar the assigned scores generated by the algorithms are. Considering the highest scoring domains of each scoring algorithm, we expect to find high agreement between all algorithms.

3.3. Experimental setup

We used the domains from the Trancolist of 26 December 2023 as our dataset[1] by crawling them using a headless variant of Chromium [4]. After filtering out all unresponsive sites (i.e. websites not returning a HTTP status code of 200), the dataset contains 846.709 entries. Here, SSL Labs can only be evaluated for the 738.894 sites that employ SSL. The crawled websites are fed to all three security scoring algorithms to generate a score.

Although the selected scoring algorithms provide public information, not all of them are fully transparent about their scoring methodology. To better understand the these algorithms, we reached out to the authors of ImmuniWeb and the CWSS-based algorithm to request additional information regarding certain methodological choices. This was not necessary for SSL Labs, since their implementation is open-source and frequently updated [21]. The scoresheet for the algorithm based on CWSS can be found in Appendix A. ImmuniWeb provides a public API to calculate scores, but due to API limits we were not able to make use of it for our large dataset. Therefore, we replicated their algorithm using the available documentation[2] and implemented all properties for which they made their methodology public. Appendix B shows which subset of properties are studied from the full ImmuniWeb algorithm. We call this limited version ImmuniWeb-Lite.

4. Comparative analysis

4.1. Score distribution

To get a first grasp on how scoring algorithms behave in our dataset, Figure 1 shows the distribution function of all scores in the dataset. We see that in SSL Labs the lower buckets are seldom used, 98% of the scores are in score bucket A showing that SSL Labs deems most sites to have a well-secured SSL/TLS connection. In the CWSS-based algorithm, 40.5% of sites receive a negative score due to only having a secure connection as implemented security and having at least one vulnerability such as leaking information or mixed-content inclusion. Similarly, ImmuniWeb-Lite scores 45.5% of the dataset below B+ due to not having any security properties except a secure connection.

1. https://tranco-list.eu/list/83JJV

2. https://www.immuniweb.com/websec/scoring

4.2. Popularity

We analyze the security scores of our dataset and compare it with the popularity of each domain according to Tranco. With this comparison we try to find out if the popularity of a website can indicate a certain security level. To be more secure, a website should strive to get a high score. Figure 2 plots the security score for each domain in function of its popularity. Each data point on the plot is the mean score of 10,000 websites. The right y-axis shows the score buckets for ImmuniWeb-Lite and SSL Labs. Additionally, the Pearson correlation coefficient (ρ) between the Tranco ranking and the average score is shown, the associated p-values are all less than 10^{-21}.

If we compare all three graphs, we see that all scoring systems follow a similar downwards trend. This is especially notable for the CWSS-based algorithm. The difference in score between the highest and lowest popularity is lower for ImmuniWeb-Lite and SSL Labs. By calculating the correlation coefficient we further confirm the less popular a domain is, the lower a score it receives. This matches our hypothesis and previous research stating that popular domains are more secure [5], [10], [18]. Nevertheless, it also shows that the choice of scoring algorithm impacts the results. The correlation between popularity and security score differs between each algorithm.

Since all scoring algorithms give high scores to popular domains, we expect that the highest scored websites in each scoring algorithm mostly contains popular domains.

4.3. Agreement

We study the agreement between the security scoring algorithms to better understand if they give a higher score to the same websites. When studying the popularity we saw that on average popular websites got a higher score. We thus expect all scoring algorithm to mostly give popular websites the highest score.

As a first comparison, we take the 10,000 highest scored websites by each algorithm and compare how much of these websites also appear in the others their top 10k. We only consider websites that had a score in all three scoring algorithms. This is plotted in Figure 3 where the overlap between the top 10,000 websites of all scoring algorithms is shown. We find that this overlap is rather limited. Only 37 sites appear in the top 10,000 of all three scoring algorithms.

To broaden the view, we extend the overlap between all the scoring algorithms in Figure 4 where it is plotted for any given amount of websites instead of just the top 10,000. If we enlarge the dataset to 100,000 sites the overlap remains limited at 2.5%, at 200,000 sites the overlap is 10.0% indicating a small agreement between the three scoring algorithms.

The minimal agreement indicates that a domain implementing strong security on a single parameter (e.g. SSL/TLS) does not necessarily indicate that the domain also has a strong general security posture. It also further stresses the need for businesses using security rankings, to pick their algorithm well. A different algorithm can lead to completely different results.

Figure 3. Overlap between the 10,000 highest scoring websites of each scoring algorithm their ranking

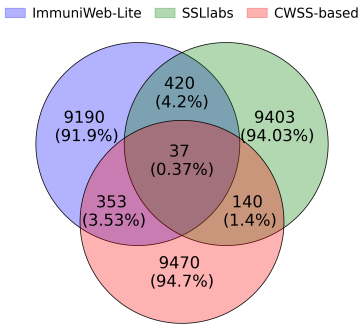
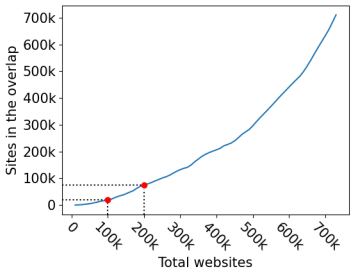


Figure 4. Evolution of the overlap between the N highest scoring websites of all three scoring algorithms



4.4. Transparency

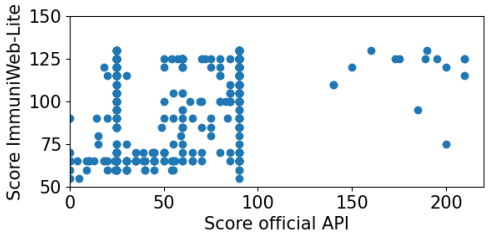
To assess the transparency of security scoring algorithms, we analyze ImmuniWeb's algorithm by comparing the score received from the official API with our ImmuniWeb-Lite.

Figure 5 shows for 540 sites how they score in ImmuniWeb-Lite and how they score in the official ImmuniWeb. The first finding we can see on this figure is that the range of given scores of ImmuniWeb-Lite is limited between 50 and 130 whilst the full implementation gives scores from 0 to 210. This discrepancy is expected, as the full ImmuniWeb implementation evaluated additional properties, such as the presence of a web application firewall or vulnerable JavaScript libraries. Since ImmuniWeb does not disclose the exact methodology behind these checks, we were unable to include them in our implementation.

Additionally, we see that there is a cutoff of scores in the ImmuniWeb API around scores 90, 60 and 20 indicating some sites are being limited in their final score. According to ImmuniWeb's documentation, the cutoff at 20 is due to a constraint regarding vulnerabilities in a CMS. However, we could not find any mention of score limitations around 90 and 60. This again highlights the limited transparency of the ImmuniWeb scoring algorithm.

As such, ImmuniWeb does not fully disclose its approach, unlike other scoring algorithms such as SSL Labs and the CWSS-based algorithm, which are more transparent by publishing their methodology or providing open-source implementations. Though, ImmuniWeb publishes part of their algorithm's scoresheet, key aspects of their methodology remain undisclosed. This lack of transparency hinders reproducibility, making independent evaluation more challenging.

Figure 5. Difference in between ImmuniWeb-Lite and the official API for 540 sites



5. Score manipulation

In order to be representative, scores must be robust against adversarial manipulation. In the case of security scoring algorithms, the generated score should properly reflect the security instated by the domain as to not give a faulty image of the security posture. This section shows how scoring algorithms can be manipulated by creating an unprotected website which still receives favorable scores.

We optimize this process by analyzing which security properties contribute to the final score for each algorithm. Additionally, we evaluate how transferable our score inflation techniques are by testing it on the closed-source, full version of ImmuniWeb and other scoring algorithms.

5.1. Subscore analysis

We infer the impact of security properties on the final score by consulting the scoresheets of each scoring algorithm.

According to ImmuniWeb's scoresheet HSTS has the highest score impact. The presence of this header gives a website 25 points, whilst the absence removes 20 points meaning this header alone can have an impact of 45 points which could be the difference between an A+ and a B-grade. Similarly, the impact of a CSP is quite high. By just setting the associated header to a non-empty value the score difference can be 40. By further setting directives, the total score difference can lead up to 105, though, ImmuniWeb limits this to a maximum of 60.

Similarly, CSP employment leads to the highest score impact for the CWSS-based algorithm, giving a score bonus of 58.93. When a website only receives positive subscores, the CSP will contribute for 26% to their final score. The second highest subscore is awarded when the X-Frame-Options header is present, awarding a score of 45.21, followed by HSTS which leads to an additional 33.52 points.

In SSL Labs, the property with the most impact is less easy to determine. Scores get awarded based on the best and worst property present. For example a website supporting both SSL2.0 and TLSv1.3 will get a protocol subscore of 50. If it only supported TLSv1.3, the score would be 100. A similar mechanic is in place for evaluating the ciphers and keys. Since the final score is a weighted average of these three properties, the best score in SSL Labs is achieved by only supporting the most secure protocol, key and cipher. A score can, however, be capped to a lower score if certain connection vulnerabilities, like Heartbleed [1] or ROBOT [3] are present. SSL Labs does however not check for more recent vulnerabilities, such as issues with session tickets [7] or the Raccoon attack [12]. Since those checks are missing it is theoretically possible to construct a TLS connection with an optimal score which can be bypassed by an attacker employing those recent vulnerabilities.

5.2. ImmuniWeb manipulation

Leveraging the insights discussed in the previous section, we optimize the ImmuniWeb-Lite score using different security settings, while keeping the website unprotected. We aim to achieve a score of A+ which equals to a nominal value of at least 100.

Even without any security features added, a newly created website via Apache2 receives a score of 75 or B+.

We are able to inflate this score by adding an empty CSP and a cookie without any content but with the secure, HttpOnly and SameSite attributes set. This gives us a score of 119 which is the highest achievable bucket of A+. Although HSTS contributes a lot to the score, we find that adding it to the website contributes negatively to the final score. ImmuniWeb subtracts points if the HSTS configuration does not fulfill certain requirements.

5.3. Transferability

To evaluate transferability, we evaluate the manipulation techniques discussed in the previous section by using the full ImmuniWeb algorithm through their API. Here, our unprotected website also performs well with a score of 135. This shows that even with limited access to a scoring's methodology, still, manipulation can lead to high score inflation with minimal effort.

Similarly, the same website configuration leads to a good score of 79.39 with the CWSS-based algorithm, even outperforming 79% of the websites in our comparative analysis. However, we could not inflate the scores received by the Security Headers and HTTP Observatory algorithms in the same way. Here, the algorithms effectively consider the content of security properties, which makes manipulation less straightforward and makes Security Headers and HTTP Observatory more robust from manipulation attempts.

Some scoring algorithms, however, are not necessarily intended to be used as a comparison tool. For example, SSL Labs focuses on providing a benchmark for individual sites rather than ranking them against others, making susceptibility to manipulation less of an issue. In such cases, inflating a score offers no real benefit beyond self-assessment. However, this can create a false sense of security if developers rely solely on these scores, believing their site is secure based on a high rating, even when its actual security remains weak.

5.4. Prevalence in the wild

To grasp how common scoring manipulation techniques are used on the Web, we scan our dataset for websites employing security policies which have no effect on their security posture yet do result in a higher score. We search for sites having a CSP, HSTS or Permissions

TABLE 2. AMOUNT OF SITES IMPLEMENTING A SECURITY PROPERTY WITHOUT EFFECTIVE CONTENT LEADING TO AN INCREASED SECURITY SCORE

Security property	Amount	Score impact ImmuniWeb	Score impact CWSS-based
Cookies	371	+10	+60.05
CSP	327	+40	+58.93
Permissions Policy	314	+15	0
HSTS	20	+15	+33.52

Policy header with no security effect. Since these headers have a predefined set of valid directives, we mark a header as having no security effect if they are either empty or do not contain any valid directive. Additionally, we look for cookies without content yet with security attributes set. We choose these security properties due to their high possible impact on scores.

This way, we find $1,019$ unique sites out of the total $846,709$ that receive inflated scores, either as a result of manipulation or misconfiguration with Tranco ranks ranging from 314 to $998,643$. Table 2 shows the distribution of sites that use a certain property and their score impact for both ImmuniWeb and the CWSS-based algorithm. Here, 12 sites have more than one ineffective policy. It is however, possible more sites employ score inflation techniques using more advanced methods we did not detect using our approach. Currently, with only $1.2‰$ of the dataset implementing a detectable form of score manipulation, the total impact on scoring algorithms is limited.

6. Discussion

Considering the fact that we managed to receive high security scores for meaningless security measures, the evaluated scoring algorithms are straightforward to deceive. To counter this, we believe security scoring algorithms should not only take into account the employed security policies, but the content hosted on the website as well. Manipulation or misconfiguration leading to inflated security scores would then have to be more complex, in contrast to the contentless website and empty security policies we employed. However, developing scoring algorithms that take into account content as well is not always feasible due to the vast amount of different policies and security properties available. For instance, it would have to account for all HTML elements that affect CSP, and which CSP directives would be able to prevent a XSS attack. To the best of our knowledge, none of the scoring algorithms discussed in this paper consider website content.

Another way to solve the identified score manipulation issue is by instating mandatory security properties, for which absence leads to score capping. SSL Labs already does this by capping the assigned grade if certain vulnerabilities are found. ImmuniWeb could for example also employ this technique by capping the grade if no secure connection is used. This approach does have a downside, however: When a score is capped based on one property, all other properties are not taken into account that could potentially make a big security impact.

Throughout our paper additional shortcomings with specific scoring algorithms came to light. ImmuniWeb's most notable issue is the lack of transparency. Its public scoresheet does not completely disclose their methodology for checking certain properties making it hard to study their scoring algorithm. Conversely, SSL Labs is completely transparent in their methodology, yet is not up-to-date with the latest vulnerabilities that should impact their given score. For instance, their algorithm does not check for newer vulnerabilities in TLS connections that could bypass the encryption.

7. Conclusion and future work

In this paper, we conducted an initial analysis of the security scoring ecosystem. By comparing three diverse security scoring algorithms, we gained insights into their methodologies and identified several shortcomings. Using a dataset of 846,709 websites, we observed that popular websites generally receive higher scores, as expected. However, we also found a lack of agreement between scoring algorithms: only 0.37% of all websites appeared in the top 10,000 highest-scoring sites across all three algorithms, a trend that continues when enlarging the set of websites. This highlights that the choice of scoring algorithm significantly impacts results, and developers should be aware of these discrepancies. Additionally, we demonstrated that it is trivial to inflate security scores of two scoring algorithms, even with limited knowledge of their methodology. By simply adding empty security headers without regard for their effectiveness, we achieved the highest possible score (A+) without actually improving security.

Based on our findings, we identify key gaps in current scoring methodologies. We recommend that scoring algorithms move beyond merely checking for the presence of security properties and instead assess their actual impact. This requires considering the content of these properties and, ideally, the context of the website they aim to protect. Seeing this as a challenging task, it presents an opportunity for future research, to incorporate deeper contextual analysis, ensuring that scores more accurately reflect a website's true security posture.

8. Limitations

In this paper, we investigated three scoring algorithms to provide an initial overview of the security scoring ecosystem. As our analysis was based on a small sample, the weight of our generalized conclusions is limited. Furthermore, we faced constraints due to API limits imposed by ImmuniWeb. As a result, we were unable to run their full scoring algorithm on our entire dataset and instead implemented it ourselves using their publicly available scoresheet. However, we found that ImmuniWeb's methodology was not fully transparent, which limited the completeness of our analysis.

9. Ethical considerations

We used publicly accessible data and crawled only public information about each website. To limit traffic we visited the sites at most twice and complied with the Menlo Report [2]. We adhered to the rate limitations imposed by API services used.

Acknowledgment

The authors would like to thank the developers of the various security scoring algorithms for their valuable insights. This research is partially funded by the Research Fund KU Leuven, Internal Funds KU Leuven, and by the Cybersecurity Research Program Flanders. This research was supported by the Research Foundation - Flanders (FWO) through a junior postdoctoral fellowship (1298825N) held by Victor Le Pochat. Victor Le Pochat is currently employed by the European Commission. The views and opinions expressed herein are personal and do not necessarily reflect those of the European Commission or other EU institutions.

References

[1] "Heartbleed Bug." [Online]. Available: https://heartbleed.com/

[2] M. Bailey, D. Dittrich, E. Kenneally, and D. Maughan, "The menlo report," *IEEE Security And Privacy*, vol. 10, no. 2, pp. 71–75, 2012.

[3] H. Böck, C. Young, and T. Vert, "Return Of Bleichenbacher's Oracle Threat (ROBOT)."

[4] Chrome, "Chrome Headless mode." [Online]. Available: https://developer.chrome.com/docs/chromium/headless

[5] G. Franken and V. Vanderlinden, "The 2024 Web Almanac: Security," HTTP Archive, Tech. Rep., Nov. 2024, issue: 13 Publication Title: The 2024 Web Almanac Volume: 6. [Online]. Available: https://almanac.httparchive.org/en/2024/security

[6] Google, "CSP Evaluator." [Online]. Available: https://csp-evaluator.withgoogle.com/

[7] S. Hebrok, S. Nachtigall, M. Maehren, N. Erinola, R. Merget, J. Somorovsky, and J. Schwenk, "We Really Need to Talk About Session Tickets: A Large-Scale Analysis of Cryptographic Dangers with TLS Session Tickets."

[8] S. Helme, "Analyse your HTTP response headers." [Online]. Available: https://securityheaders.com/

[9] Immuniweb, "Website Security Test." [Online]. Available: https://www.immuniweb.com/websec/

[10] G. Karopoulos, D. Geneiatakis, and G. Kambourakis, "Neither Good nor Bad: A Large-Scale Empirical Analysis of HTTP Security Response Headers," in *Trust, Privacy and Security in Digital Business*, S. Fischer-Hübner, C. Lambrinoudakis, G. Kotsis, A. M. Tjoa, and I. Khalil, Eds. Cham: Springer International Publishing, 2021, pp. 83–95.

[11] V. Le Pochat, T. Van Goethem, S. Tajalizadehkhoob, M. Korczynski, and W. Joosen, "Tranco: A Research-Oriented Top Sites Ranking Hardened Against Manipulation," in *Proceedings 2019 Network and Distributed System Security Symposium*. San Diego, CA: Internet Society, 2019.

[12] R. Merget, M. Brinkmann, N. Aviram, and J. Somorovsky, "Raccoon Attack: Finding and Exploiting Most-Significant-Bit-Oracles in TLS-DH(E)."

[13] MITRE, "CWE - Common Weakness Scoring System (CWSS)."

[14] Mozilla, "HTTP Header Security Test - HTTP Observatory | MDN." [Online]. Available: https://developer.mozilla.org/en-US/observatory

[15] L. Pelchen, "Internet Usage Statistics In 2024," Mar. 2024, section: Internet. [Online]. Available: https://www.forbes.com/home-improvement/internet/internet-statistics/

[16] Qualys, "SSL Server Rating Guide." [Online]. Available: https://github.com/ssllabs/research/wiki/SSL-Server-Rating-Guide

[17] SecurityScoreCard, "How SecurityScorecard calculates your scores," May 2024. [Online]. Available: https://support.securityscorecard.com/hc/en-us/articles/8366223642651-How-SecurityScorecard-calculates-your-scores

[18] T. van Goethem, P. Chen, N. Nikiforakis, L. Desmet, and W. Joosen, "Large-Scale Security Analysis of the Web: Challenges and Findings," in *Trust and Trustworthy Computing*, T. Holz and S. Ioannidis, Eds. Cham: Springer International Publishing, 2014, pp. 110–126.

[19] Vanta, "How to score third-party vendor risks in 3 easy steps." [Online]. Available: https://www.vanta.com/collection/tprm/vendor-risk-scores

[20] L. Weichselbaum, M. Spagnuolo, S. Lekies, and A. Janc, "CSP Is Dead, Long Live CSP! On the Insecurity of Whitelists and the Future of Content Security Policy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. Vienna Austria: ACM, Oct. 2016, pp. 1376–1387. [Online]. Available: https://dl.acm.org/doi/10.1145/2976749.2978363

[21] D. Wetter, "testssl/testssl.sh," Jan. 2025, original-date: 2014-07-01T11:55:26Z. [Online]. Available: https://github.com/testssl/testssl.sh

[22] Q. Xie and B. Liu, "Building an Open, Robust, and Stable Voting-Based Domain Top List," 2022.

Appendix A.
CWSS-based algorithm scoresheet

TABLE 3. CWSS-BASED SCORING ALGORITHM FROM VAN GOETHEM ET AL.

	Present
Content-Security-Policy	+58.93
X-Frame-Options	+45.21
Strict-Transport-Security	+33.52
HttpOnly Cookie	+28.21
Secure Cookie	+31.84
Iframe Sandboxing	+19.95
X-content-Type-Options	+8.02
SSL stripping	-30.16
X-XSS-Protection	-28.33
Mixed-Content Inclusion	-13.42
Information Leakage	-9.43

Appendix B.
ImmuniWeb properties

Table 4 contains the scoresheet we used to determine the ImmuniWeb-Lite score. It is a reduced version of the full scoring algorithm as not all parameters were public. If a certain attribute is present, we add its score, otherwise we subtract.

TABLE 4. PROPERTIES STUDIED IN IMMUNIWEB-LITE SCORING ALGORITHM

	Present	Absent
HttpOnly cookies	0	-5
Secure cookies	+5	0
SameSite cookies	+5	-1
__Secure cookieprefix	+5	0
__Host cookieprefix	+5	0
Permissions policy	+15	0
HSTS	+25	-20
HSTS age $<$ 6 months	-10	0
X-Frame-Options	+15	0
X-Content-Type-Options	+15	0
CSP	+20	-20
CSP default-src is 'none' or 'self'	+5	0
CSP has '*' in default-src	-10	0
CSP has '*' in another directive	-10	0
CSP has frame-ancestors without '*'	+10	0
CSP has frame-ancestors with '*'	+5	0
CSP has reflected-XSS directive	+5	0
CSP has upgrade-insecure-requests or block-all-mixed-content	+5	0
Server-Info	-5	0
X-Powered-By	-5	0
X-AspNet-Version	-5	0