R. Röhrig et al. (Eds.)
© 2021 The authors and IOS Press.
This article is published online with Open Access by IOS Press and distributed under the terms of the Creative Commons Attribution Non-Commercial License 4.0 (CC BY-NC 4.0).
doi:10.3233/SHTI210558

The Security State of the German Health Web: An Exploratory Study

Frederic HENN[a,1], Richard ZOWALLA[b,c,1], and Andreas MAYER[a,b,2]
[a] *Department of Software Engineering, Heilbronn University, Germany*
[b] *Department of Medical Informatics, Heilbronn University, Germany*
[c] *Center for Machine Learning, Heilbronn University, Germany*

Abstract. The internet has become an important resource for health information and for interactions with healthcare providers. However, information of all types can go through many servers and networks before reaching its intended destination and any of these has the potential to intercept or even manipulate the exchanged information if data's transfer is not adequately protected. As trust is a fundamental concept in healthcare relationships, it is crucial to offer a secure medical website to maintain the same level of trust as provided in a face-to-face meeting. This study provides a first analysis of the SSL/TLS security of and the security headers used within the health-related web limited to web pages in German, the German health web (GHW). **Methods:** *testssl.sh* and *TLS-Scanner* were used to analyze the URLs of the 1,000 top-ranked health-related web sites (according to PageRank) for each of the country-code top level domains: ".de", ".at" and ".ch". **Results:** Our study revealed that most websites in the GHW are potentially vulnerable to common SSL/TLS security vulnerabilities, offer deprecated SSL/TLS protocol versions and mostly do not implement HTTP security headers at all. **Conclusions:** These findings question the concept of trust within the GHW. Website owners should reconsider the use of outdated SSL/TLS protocol versions for compatibility reasons. Additionally, HTTP security headers should be implemented more consequently to provide additional security aspects. In future work, the authors intend to repeat this study and to incorporate a website's category, i.e. governmental or public health, to get a more detailed view of the GHW's security.

Keywords. health information seeking, internet, cyber security, data security, consumer health information, trust

1. Introduction

The internet has become an important resource for health information and for interactions with healthcare providers [1,2]. However, information of all types can go through many servers and networks before reaching its intended target destination. Along the communication path, attackers may eavesdrop, intercept or even manipulate the exchanged information if data's transfer is not adequately protected as the Snowden revelations showed [3]. As trust is a fundamental concept in healthcare relationships [1], it is crucial to offer a secure medical website to maintain the same level of trust as

[1] Authors contributed equally.
[2] Corresponding Author, Andreas Mayer, Department of Medical Informatics, Heilbronn University, Max-Planck-Str. 39, 74081 Heilbronn, Germany; E-mail: andreas.mayer@hs-heilbronn.de

provided in a face-to-face meeting. Therefore, Transport Layer Security (TLS) [4] is used to protect the confidentiality, integrity, and authenticity of the data transferred[3]. Duly deployed and accompanied with the usage of HTTP security headers [5], an adequate level of security may be achieved. While large-scale analyses of SSL/TLS deployments have already been made [6,7], to the best of the authors' knowledge, no study has been published about the security of the health-related web. We provide a first analysis of the SSL/TLS security of and the security headers used within the health-related web limited to web pages in German, the German health web (GHW) [8].

2. Methods and Material

Automated open-source security scanners, such as *testssl.sh* [9] or *TLS-Scanner* [10], are used by security researchers, penetration testers or administrators to perform a deep analysis of a given web server's configuration. These tools also allow for a subsequent analysis of SSL/TLS configuration and provided HTTP security headers. In addition, they are capable to detect common SSL/TLS vulnerabilities such as *Lucky13* [11], *Sweet32* [12] or *Breach* [13].

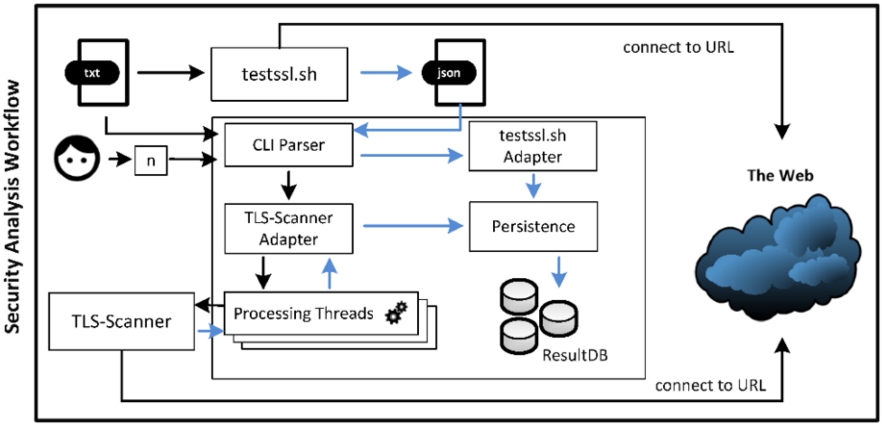

Figure 1. Workflow and architecture of the analysis software: URLs are provided via a TXT file. The *n* is a user-defined input parameter and specifies the amount of processing threads used for the *TLS-Scanne*r analysis. Black lines indicate the data flow related to unprocessed URLs; Blue lines indicate the result data flow.

In this study, we relied on *testssl.sh* (v3.0) and *TLS-Scanner* (v4.0.0) to mitigate a potential bias towards a certain scanner. These tools are used from a command line and perform a scan for only one URL at the time. Even in case they support bulk processing (*testssl.sh*), the resulting reports are difficult to interpret. For this reason, we decided to implement a software component written in Java, which is capable of processing and analyzing the information received from a scanner and to give a TLS/SSL security estimation of a given website. Based on the chosen scanner, the workflow differs in the early stages: In case of *testssl.sh,* the scanning is conducted in bulk processing mode and

[3] Secure Socket Layer (SSL) is the predecessor of TLS. Today, all SSL protocol versions are inherently insecure and must not be used.

is run independently. In contrast, *TLS-Scanner* does not offer out-of-the-box bulk processing.

For this reason, we added a component, which executes *TLS-Scanner* for multiple URLs in parallel. The results for each URL are then collected and analyzed by our software. For further analyses the results of both scanners are stored in a PostgreSQL (v10.6) database. Figure 1 depicts the system's workflow for the scanning and analysis process. For this study, we obtained the URLs of the 1,000 top-ranked health-related web sites (according to PageRank) for each of the country-code top level domains: ".de", ".at" and ".ch" from a crawl of the GHW conducted in 2020 [6]. The PageRank was computed on the graph representation of the GHW. The methodology used to obtain the GHW graph is described in detail in [6].

3. Results

The scans for all 3,000 URLs (n=1,000 per ccTLD) were run from December 17, 2020 to December 30, 2020 on a Ubuntu 20.04 LTS 64-bit virtual machine hosted in the university's datacenter. The *testssl.sh* scan took 13 hours and 40 minutes while *TLS-Scanner* took 7 days, 17 hours and 39 minutes to complete. In total, both scans contain 231 features per URL, ie certificates, signature algorithms, cipher suites and common security mechanisms, and extensions such as perfect forward secrecy, certificate transparency or DNS Certification Authority Authorization (CAA).

3.1. SSL/TLS configuration

Tables 1 depicts the distribution of offered SSL/TLS protocol versions. One web server usually supports several protocol versions for compatibility reasons. Interestingly, some websites offer deprecated protocol versions: SSLv2, SSLv3, TLS1.0, TLS1.1 [14,15].

Table 1. SSL/TLS protocol versions offered by the web servers for each ccTLD. [a] Unknown: URLs, which were (a) not reachable, (b) did offer SSL/TLS only after a temporary or permanent redirect to another domain, or (c) the scanners couldn't test SSL/TLS protocols for unknown reasons.

	testssl.sh (%)			**TLS-Scanner (%)**		
ccTLD Protocol	**de**	**at**	**ch**	**de**	**at**	**ch**
No SSL/TLS	31 (3.1)	63 (6.3)	34 (3.4)	17 (1.7)	49 (4.9)	22 (2.2)
SSLv2	0 (0.0)	0 (0.0)	0 (0.0)	1 (0.1)	0 (0.0)	0 (0.0)
SSLv3	4 (0.4)	8 (0.8)	4 (0.4)	6 (0.6)	9 (0.9)	3 (0.3)
TLS 1.0	338 (33.8)	385 (38.5)	325 (32.5)	345 (34.5)	362 (36.2)	328 (32.8)
TLS 1.1	354 (35.4)	429 (42.9)	345 (34.5)	357 (35.7)	401 (40.1)	350 (35)
TLS 1.2	954 (95.4)	913 (91.3)	936 (93.6)	960 (96)	888 (88.8)	955 (95.5)
TLS 1.3	433 (43.3)	441 (44.1)	534 (53.4)	443 (44.3)	443 (44.3)	549 (54.9)
Unknown[a]	15 (1,5)	23 (2,3)	30 (3)	16 (1,6)	46 (4,6)	23 (2,3)

3.2. HTTP Security Headers

Table 2 depicts the distribution of important and well-known HTTP security headers, which were *not* used on a given website. Although, implementing these headers is typically a trivial and well-proven task.

Table 2. Results of the HTTP security headers analysis: An entry means, that the given security header is **not** set for the given web site. [a] Unknown: URLs, which were (a) not reachable (b) did not offer SSL/TLS, (c) did offer SSL/TLS only after a temporary or permanent redirect to another domain, or (d) the scanners couldn't extract any security header at all.

	testssl.sh (%)			**TLS-Scanner (%)**		
ccTLD Security Header	**de**	**at**	**ch**	**de**	**at**	**ch**
X-Frame-Options	716 (71.6)	740 (74)	795 (79.5)	758 (75.8)	740 (74)	822 (82.2)
X-XSS-Protection	770 (77)	786 (78.6)	807 (80.7)	823 (82.3)	793 (79.3)	843 (84.3)
X-Content-Type-Options	639 (63.9)	710 (71)	750 (75)	667 (66.7)	705 (70.5)	776 (77.6)
Referrer Policy	868 (86.8)	848 (84.8)	888 (88.8)	900 (90)	845 (84.5)	916 (91.6)
Strict-Transport-Security	692 (69.2)	675 (67.5)	711 (71.1)	718 (71.8)	698 (69.8)	742 (74.2)
Content-Security-Policy	922 (92.2)	822 (82.2)	874 (87.4)	879 (87.9)	846 (84.6)	925 (92.5)
Unknown[a]	47 (4.7)	84 (8.4)	65 (6.5)	33 (3.3)	95 (9.5)	45 (4.5)

3.3. Vulnerabilities

In addition, each URL was automatically tested for known SSL/TLS vulnerabilities. Please note, that both scanners test slightly different and thus, the vulnerabilities tested differ between both scanners. According to *testssl.sh,* the following potential vulnerabilities were found in the given data set: *Lucky13* (n_{de}=837 (83.7%); n_{at}=791 (79.1%); n_{ch}=778 (77.8%)), *Breach* (n_{de}=671 (67.1%); n_{at}=577 (57.7%); n_{ch}=543 (54.3%)), *Beast* (n_{de}=337 (33.7%); n_{at}=385 (38.5%); n_{ch}=323 (32.3%)), *Sweet32* (n_{de}=152 (15.2%); n_{at}=172 (17.2%); n_{ch}=181 (18.1%)). *TLS-Scanner* revealed the following potential vulnerabilities in the given data set: *Breach* (n_{de}= 668 (66.8%); n_{at}= 550 (55%); n_{ch}= 559 (55.9%)), *Robot* (n_{de}= 85 (8.5%); n_{at}= 102 (10.2%); n_{ch}= 176 (17.6%)).

Overall, 2,406/3,000 (80.2%) websites are potentially vulnerable to *Lucky13.* According to *testssl.sh*, 1,791/3,000 (59.7%) websites (*TLS-Scanne*r: 1,777/3,000 (59.23%)) are potentially vulnerable to the *Breach* attack. In addition, 1,045/3,000 (34.83%) websites are potentially vulnerable to *Beast,* 505/3,000 (16.83%) to *Sweet32,* and 362/3,000 (12.1%) to *Robot.*

4. Discussion

Our study revealed that most websites in the GHW are potentially vulnerable to common SSL/TLS security vulnerabilities, offer deprecated SSL/TLS protocol versions and mostly do not implement HTTP security headers at all. Consequently, these findings question the concept of trust within the GHW. Website owners should reconsider the use of outdated SSL/TLS protocol versions for compatibility reasons. Additionally, HTTP security headers should be implemented to provide additional security aspects.

Several limitations apply for this study. First, a website's security cannot solely be judged on its SSL/TLS configuration and the HTTP security headers used. For example, factors such as the libraries used to build a certain website or a security-aware software development process also contribute to the overall security of a website. Second, the scanners may have produced some false positives. Finally, a website providing health-related information material may not have the same security needs as a website, which processes personal health-related data such as health insurance companies or online consultation services.

In future work, the authors intend to repeat this study and to incorporate a website's category, i.e. governmental or public health, to get a more detailed view of the GHW's security. Moreover, the level of protection for each individual website should be assessed in order to evaluate the criticality of a given vulnerability and to estimate the related impact. In addition, analyzing health-related websites regarding the software libraries and versions used could also shed light on the security state of the GHW. Moreover, only a fraction of the features captured by both scanners were analyzed in this paper. Due to this variety of available features, there exists a lot of potential data that can additionally be used to analyze further security aspects of a given website such as certificate chains as well as additional security mechanisms, e.g. perfect forward secrecy or DNS CAA.

Finally, the prevalence and usage of third-party cookies provided by such websites can give additional insights regarding privacy aspects.

5. Declarations

The authors declare, that there is no conflict of interest.

RZ, AM: conception of the work, FH, RZ: data acquisition and interpretation; FH, AM: data analysis and interpretation; FH, RZ: writing the manuscript, AM revising of the manuscript. All authors approved the manuscript in the submitted version and take responsibility for the scientific integrity of the work.

References

[1] Sbaffi L, Rowley J. Trust and credibility in web-based health information: a review and agenda for future research. J Med Internet Res 2017 Jun 19;19(6):e218; DOI: 10.2196/jmir.7579

[2] Prestin A, Vieux SN, Chou WS; Is online health activity alive and well or flatlining? Findings from 10 years of the Health Information National Trends Survey. J Health Commun 2015 Jul;20(7):790-798. DOI: 10.1080/10810730.2015.1018590

[3] Snowden E. J. Permanent Record. Metropolitan Books/Henry Holt, 2019

[4] Rescorla E. The Transport Layer Security (TLS) Protocol Version 1.3; RFC 8446; URL: https://tools.ietf.org/html/rfc8446 (03/25/2021)

[5] Averay A, Righetto D, Manico J. OWASP Secure Headers Project; URL: https://owasp.org/www-project-secure-headers/ (03/23/2020)

[6] Kotzias, P, Razaghpanah A., Amann J, et al. Coming of Age: A Longitudinal Study of TLS Deployment. In: In 2018 Internet Measurement Conference (IMC '18), October 2018, Boston, MA, USA. ACM, New York, NY, USA; DOI: 10.1145/3278532.3278568

[7] Hu, Q., Asghar, M. R., Brownlee, N. A large-scale analysis of HTTPS deployments: Challenges, solutions, and recommendations. Journal of Computer Security 2021, 29(1), 25–50. DOI:10.3233/jcs-200070

[8] Zowalla R, Wetter T, Pfeifer D. Crawling the German Health Web: Exploratory Study and Graph Analysis. J Med Internet Res 2020;22(7):e17853; DOI: 10.2196/17853

[9] Wetter, D: testssl.sh - Testing TLS/SSL encryption. URL: https://testssl.sh/ (03/18/2021)

[10] Merget R, Somorovsky J, Aviram N, et al. Scalable Scanning and Automatic Classification of TLS Padding Oracle Vulnerabilities. In: 28th USENIX Security Symposium (USENIX Security 19). Santa Clara, CA: USENIX Association; 2019:1029-1046.

[11] Al Fardan N J, Paterson K G. Lucky Thirteen: Breaking the TLS and DTLS Record Protocols, 2013 IEEE Symposium on Security and Privacy, Berkeley, CA, USA, 2013, pp. 526-540, DOI: 10.1109/SP.2013.42.

[12] Bhargavan K, Leurent G. On the Practical (In-)Security of 64-bit Block Ciphers: Collision Attacks on HTTP over TLS and OpenVPN. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS '16). Association for Computing Machinery, New York, NY, USA, 456–467. DOI: 0.1145/2976749.2978423

[13] Y. Gluck, N. Harris, A. Prado, BREACH: Reviving the CRIME attack, Black Hat USA, 2013.

[14] Bundesamt für Sicherheit in der Informationstechnik. Mindeststandard des BSI nach § 8 Abs. 1 Satz 1 BSIG zur Verwendung von Transport Layer Security (TLS) Version 2.1. URL: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Mindeststandards/Mindeststandard_BSI_TLS_Version_2_1.pdf (03/18/2021)

[15] U.S. National Security Agency. Eliminating Obsolete Transport Layer Security (TLS) Protocol Configurations; URL: https://media.defense.gov/2021/Jan/05/2002560140/-1/-1/0/ELIMINATING_OBSOLETE_TLS_UOO197443-20.PDF (03/22/2021)