A Hard Lesson: Assessing the HTTPS Deployment of Italian University Websites[*]

Stefano Calzavara, Riccardo Focardi, Alvise Rabitti, and Lorenzo Soligo

Università Ca' Foscari Venezia
name.surname@unive.it

Abstract

In this paper we carry out a systematic analysis of the state of the HTTPS deployment of the most popular Italian university websites. Our analysis focuses on three different key aspects: HTTPS adoption and activation, HTTPS certificates, and cryptographic TLS implementations. Our investigation shows that the current state of the HTTPS deployment is unsatisfactory, yet it is possible to significantly improve the level of security by working exclusively at the web application layer. We hope this observation will encourage site operators to take actions to improve the current state of protection.

1 Introduction

HTTP is the workhorse protocol of the Web, yet it does not provide any confidentiality, integrity or authenticity guarantee by default. Standard HTTP traffic is just unauthenticated plaintext, which can be read, modified and forged by attackers who are in control of the network, e.g., rogue access points and malicious Internet service providers. Luckily, this shortcoming of HTTP can be overcome by the adoption of its secure counterpart HTTPS, which runs HTTP on top of cryptographic protocols like TLS. HTTPS is phenomenally popular nowadays, to the point that the amount of HTTPS traffic has finally surpassed the amount of HTTP traffic [12]. Yet, previous research showed that the correct deployment of HTTPS is particularly tricky and things can go wrong at many different levels [3, 4, 14, 16, 18]. This means that the adoption of HTTPS is not just a binary checkbox, but rather multiple factors must be taken into account for a realistic security assessment.

In this paper we carry out a systematic analysis of the state of the HTTPS deployment of the most popular Italian university websites. In particular, we study:

- the adoption and activation of HTTPS at the (web) application layer (Section 3). The lack of HTTPS support and the use of unsafe practices in the HTTPS activation can entirely void security against network attackers, since communication might run unencrypted;
- the use of best practices in HTTPS certificates (Section 4). The incorrect management of HTTPS certificates might unduly expose users to phishing attempts or even lead to the disclosure of the cryptographic keys used to protect communication;
- the correct cryptographic implementation of the TLS protocol (Section 5). Cryptographic flaws in the TLS deployment can reveal cryptographic keys to network attackers, leading to various confidentiality and integrity breaches.

In our security assessment we use numeric scores to measure the compliance with respect to security best practices, where higher scores stand for better security. At the end of our study, we assign a final numeric score to all sites and show that the current state of the HTTPS deployment is unsatisfactory, yet it is possible to significantly improve the level of security by working exclusively at the web application layer (Section 6). We hope this observation will encourage site operators to take actions to improve the current state of protection.

2 Preliminaries

2.1 HTTPS and TLS

HTTPS is an encrypted variant of HTTP based on the TLS cryptographic protocol. At a high level, TLS can be described as follows:

1. The client initiates a handshake with the server by proposing a TLS version and a list of supported ciphersuites;

2. The server chooses the lower between its highest supported TLS version and the TLS version proposed by the client. It then picks a supported ciphersuite from the proposed list and sends to the client an X.509 certificate, which contains information about the server's identity, the server's public key and the issuing certification authority;

3. The client confirms the validity of the X.509 certificate by checking that it was issued by a trusted certification authority to the hostname to which it is trying to connect, thus getting a proof of authenticity;

4. The client and the server take appropriate actions to generate a fresh session key, which is used to protect the communication by means of symmetric encryption, thus ensuring its confidentiality and integrity.

The session key establishment can be implemented in different ways and takes advantage of the server's public key.

2.2 Domains and Sub-Domains

On the Web, servers are typically identified by a *fully qualified domain name* (FQDN), i.e., a dot-separated sequence of labels terminated by a *top-level domain* (TLD) from a fixed list. For example, `www.unive.it` is a FQDN under the TLD `.it`. Domain registration typically operates at the granularity of TLD+1: this means that an organization can register a domain name like `unive.it` and then create arbitrary *sub-domains* like `www.unive.it` and `idp.unive.it`. It is common practice to create different sub-domains for different services, e.g., `www.unive.it` to serve the university website and `idp.unive.it` for authentication.

The security of sub-domains plays an important role on web application security, most notably because sub-domains can share *cookies*. For example, `www.unive.it` and `idp.unive.it` can both set cookies with the `Domain` attribute set to `.unive.it`: such cookies, called *domain cookies*, are sent to both services. Though this practice is useful and popular, e.g., to implement authentication across different sub-domains, it also means that attacking `www.unive.it` might break the confidentiality and integrity of cookies at `idp.unive.it` and vice-versa [8].

2.3 Threat Model

We assume an active network attacker who is able to add, remove or modify messages sent between a client and a server. The attacker also controls a malicious website, which is navigated by the attacked client. By means of the website, the attacker can inject scripts in the client from an attacker-controlled origin, which is relevant for a subset of the considered attacks. However, the attacker can neither break the Same Origin Policy (SOP), nor exploit any bug in the browser. We assume the attacker cannot exploit timing side-channels, since the feasibility of such attacks is generally hard to assess.

In our security analysis, we also occasionally make considerations about passive network attackers, who just sniff the network traffic and do not take actions to avoid detection. These attackers are particularly interesting because they only require very limited skill.

2.4 Analysed Websites

We focus on the HTTPS deployment of the Italian universities which are included in the QS World University Ranking 2020.[1] We identified 34 universities overall, then extracted their official names from QS and searched for them on Google: the first website in the results page is the one we used as the entry point of our security analysis. The rationale of this choice is that we want to analyse the website which is most likely accessed when users look for a specific university via a search engine. We provide the full list of the analysed websites online [6].

3 HTTPS Adoption and Activation

3.1 Security Practices

When users access a website like `www.unive.it` without specifying any protocol, browsers normally contact the site using the HTTP protocol. A common practice is then to upgrade the communication to HTTPS through a redirection, e.g., by using the `Location` header, to send the browser to `https://www.unive.it`. Though this practice definitely improves security over the plain adoption of HTTP, it is still vulnerable to attacks like SSL stripping [17]. Specifically, since the first communication still happens over HTTP, the attacker can block the redirection to HTTPS and forcefully prevent the security upgrade to encrypted communication. To avoid this pitfall, websites can deploy HTTP Strict Transport Security (HSTS). HSTS instructs the browser to never contact the website over HTTP, so that every communication attempt on HTTP is automatically upgraded to HTTPS [13].

There are several deployment options for HSTS. Specifically, HSTS can be activated using the `Strict-Transport-Security` header, using the `max-age` attribute to specify the lifetime of protection. The `includeSubDomains` option can be used to extend the HTTPS upgrade to all the sub-domains of the protected website, which is particularly useful to protect other web applications and ensure the confidentiality and integrity of cookies shared among them [8]. To further reduce the attack surface of the "trust upon first use" model of HSTS, browsers include a *preload* list of HSTS-protected websites: communication with such websites is automatically upgraded to HTTPS, even before getting the HSTS header from them.

Finally, once a website is accessed over HTTPS, it is important that it does not include sub-resources over HTTP, otherwise the confidentiality and integrity guarantees of HTTPS can be undermined. Modern browsers mitigate this severe threat by implementing the Mixed Content

[1] `https://www.topuniversities.com/university-rankings/world-university-rankings/2020`

policy.[2] Roughly, this policy mandates that active contents like scripts must be blocked when they are included in HTTPS pages over HTTP connections, while browser vendors are left at liberty of being more tolerant towards passive content like images. It is a good practice to avoid the use of mixed content in high-security sites to ensure appropriate protection also to users of browsers which do not implement the Mixed Content policy, e.g., legacy browsers. It is worth noticing that, starting from 2020, Google Chrome will automatically block all forms of mixed content.[3] It is thus even more important to avoid the use of mixed content to prevent breakage.

3.2 Security Measurement

To analyse the practices used to activate HTTPS, we have to check for the presence of redirects. Since redirects can be performed using JavaScript, we cannot just check for the use of `Location` headers and we rather use Puppeteer[4] to access the websites over HTTP using Chrome. We then check the protocol used in the final landing page to see whether HTTPS was activated in some way since the original HTTP request. Since Puppeteer has access to the response headers of the landing page, we can also use it to log the presence of HSTS headers. However, this is not entirely sufficient, because HSTS activation can (and should) be forced on the TLD+1 using the `includeSubDomains` option; we then also check for the presence of such headers there using the `curl` command line tool. Finally, we assign each website a score as follows:

- 0 points: no redirection to HTTPS took place. This means that the navigation is performed over HTTP by default, which makes network attacks trivial to carry out;
- 3 points: redirection from HTTP to HTTPS, but lack of HSTS adoption. The website is vulnerable to SSL stripping, yet it is normally served over HTTPS and careful users might notice when the site is unexpectedly served over HTTP;
- 5 points: redirection from HTTP to HTTPS + HSTS. The website is protected against SSL stripping, but other applications served on sub-domains might be vulnerable to this attack and domain cookies might be leaked or set over HTTP;
- 7 points: redirection from HTTP to HTTPS + HSTS with the `includeSubDomains` option on TLD+1. This ensures that all the applications on the domain are always accessed over HTTPS, as well as granting the confidentiality and integrity of domain cookies.

As for mixed content checking, we still rely on Puppeteer to track all the outgoing HTTP requests of the accessed web pages, including their type. We then assign the following scores:

- 0 points: presence of active mixed content in the homepage. The integrity of the page is seriously at harm on legacy clients not implementing the Mixed Content specification;
- 1 point: presence of passive mixed content in the homepage. Website defacement is potentially possible and there is a significant risk of privacy breaches due to the presence of outgoing HTTP requests, even on modern clients which are tolerant on such content;
- 2 points: there is no presence of mixed content in the homepage. This ensures that the full page content is served over encrypted connections, which is necessary to provide optimal confidentiality and integrity guarantees.

[2] https://www.w3.org/TR/mixed-content/

[3] https://blog.chromium.org/2019/10/no-more-mixed-messages-about-https.html

[4] https://github.com/puppeteer/puppeteer

Based on this scoring system, websites with a score lower than 5 are not secure even against passive network attackers, since they either do not redirect to HTTPS or include some HTTP content in the homepage. A minimal score for protection against active network attackers is 7: this ensures that SSL stripping is not possible and that the confidentiality and integrity of the homepage cannot be trivially harmed by the use of mixed content.

3.3 Experimental Results

The first observation we make is that the majority of the websites is eventually accessed over HTTPS: this is the case for 28 out of 34 websites (82%). As to the remaining 6 websites, we observed that `web.uniroma2.it`, `poliba.it`, `www.unina.it` and `www.unipa.it` do not perform any redirection to HTTPS, but can still be accessed over HTTPS when the protocol is explicitly typed in the browser address bar. This means that users can access these sites securely, but the very large majority of users is normally left unprotected; in particular, even passive network attackers have full visibility of the exchanged HTTP traffic. The other two websites are even more problematic from a security perspective, because `www.uniroma3.it` cannot be accessed over HTTPS and `www.unife.it` forces an explicit downgrade from HTTPS to HTTP using JavaScript. We manually verified that both websites have a private area, which is accessible over HTTPS. However, this is clearly not sufficient for security, because an active network attacker can break the integrity of the homepage and force the adoption of the HTTP protocol also on the private area. Given that it is not possible to navigate these two sites over HTTPS, we excluded them from all further analyses.

As to the 28 websites which force the activation of HTTPS in some way, they mostly do it by means of redirects: we observed that this is the case for 25 websites. The security implication is that the majority of the Italian university websites are vulnerable to SSL stripping. Only 3 websites make use of HSTS: `www.unibo.it`, `www.unifi.it` and `www.polimi.it`, with the latter two even activating the `includeSubDomains` option. Unfortunately, in both cases, this is not done on the TLD+1, but on the `www` sub-domain, which essentially voids protection.

As for the presence of mixed content, the picture is largely positive, yet we also observed two insecure practices due to the use of passive mixed content. First, `www.unina.it` includes an image over HTTP, which is sufficient to leak the website cookies even against a passive network attacker who just sniffs the HTTP traffic. A second problem affects `www.unipa.it`, where the form of the internal search engine is sent over HTTP: this might not only expose the website cookies, but also reveal all the search keywords typed by the website users, including the corresponding results.

The complete picture of this part of our analysis is summarized by the scores in the "Activation" column at [6]. It turns out that 6 out of 34 websites do not comply with minimal security practices, hence are completely at harm even against passive network attackers: in particular, 4 sites got a score lesser than 5 and 2 sites were excluded from our analysis because they could not be accessed over HTTPS in the first place. The other 28 websites offer a minimal degree of protection against passive network attackers by ensuring that communication is encrypted and by ruling out mixed content from their homepages, yet only 3 of them implement safeguards against active tampering attempts like SSL stripping.

4 HTTPS Certificates

4.1 Security Practices

It is well-known that certificates should only be signed by a trusted certification authority and should only be considered valid up to a given expiration date. What is likely less known is that certificates come in different forms. In particular, by increasing level of security guarantees:

1. *Domain Validated* (DV) certificates are issued after proving some form of control over a given domain name, but do not provide any form of binding between the domain name and the organization which claims ownership of the domain;

2. *Organization Validated* (OV) certificates are only issued after proving that a domain name is actually controlled by a given physical organization. This requires the presentation of appropriate documentation about the organization asking for the certificate;

3. *Extended Validated* (EV) certificates are similar to OV certificates, but are subject to even stricter security checks. Browsers often rely on custom security indicators for EV certificates and show the name of the owning organization directly in the address bar.

Major organizations like universities should only use OV or EV certificates, since DV certificates provide no protection against phishing attempts. For example, an attacker could get a valid DV certificate for `www.unvie.it` and host a website which pretends to be the legitimate website of Università Ca' Foscari Venezia (`www.unive.it`).[5]

Moreover, security-conscious administrators should avoid the use of *wildcard* certificates. Wildcard certificates apply to arbitrary sub-domains like `*.unive.it`, hence are typically reused on a multitude of different hosts. This simplifies the HTTPS deployment, but also implies that all such hosts have access to the same cryptographic keys, hence the compromise of any host would suffice to get read and write access to all the HTTPS traffic exchanged with any subdomain of `unive.it`. Wildcards cannot be used in EV certificates, but it is worth noticing that even certificates which do not make use of wildcards might be unduly issued for a large number of domains by specifying multiple Subject Alternative Names (SANs) in them.

4.2 Security Measurement

We check the use of certificates at the analysed websites by using the `openssl` tool from the command line. In particular, we rely on the following scoring system:

- 0 points: the certificate is not signed by a trusted certification authority or has expired. In the former case a network attacker can just replace the server's certificate with a fake one while going unnoticed, while in the latter case it is plausible that keys might have been compromised over time, e.g., by brute-forcing;

- 2 points: the website uses a valid DV certificate signed by a trusted certification authority. This provides both confidentiality and integrity, but not necessarily authenticity;

- 4 points: the website uses a valid OV certificate signed by a trusted certification authority. This gives attentive and technically educated users a way to check the identity of the server's organization when in doubt;

[5] We actually registered `www.unvie.it` after realizing that this is one of our most common typos...

- 5 points: the website uses a valid EV certificate signed by a trusted certification authority. This provides immediate visual feedback about the identity of the server's organization or at least easy access to this information.

Finally, we award 2 extra points to certificates which are only valid for a "small" number of domain names. Such certificates do not make use of wildcards and keep the number of SANs under a given threshold t. To choose t, we rely on the distribution of the number of SANs in the collected certificates as reported below.

Based on this scoring system, certificates should be awarded at least 6 points to be considered compliant with security best practices: this requirement is satisfied by OV and EV certificates with only limited reuse on multiple domains.

4.3 Experimental Results

We start by discussing a positive result: all the analysed websites have valid certificates and none of them relies on DV certificates. This means that minimal security practices to ensure communication security and complicate phishing attempts are put in place, and all websites were awarded at least 4 points. Only 5 websites, however, use EV certificates: `www.unimore.it`, `www.unipg.it`, `www.unitn.it`, `www.uniud.it` and `www.unive.it`. An interesting fact is that all the collected certificates were issued by the same certification authority (TERENA).

We also noticed that only two websites make use of wildcard certificates, i.e., `www.unibo.it` and `www.unicatt.it`. This means that most certificates cannot be arbitrarily reused on subdomains, which is important to minimize the risk of leaking private keys and reduce the attack surface. There are however a few interesting observations about the use of SANs, since their distribution is highly skewed. In particular, the mode of the distribution is 2, the mean is 21.7 and the variance is 47.4. We show the full frequency table of the number of SANs in Table 1. Observe that there are a few cases where the certificate is valid for so many domains that it is essentially equivalent to having a wildcard: for example, the certificate of `www.unifi.it` is valid on 247 different domains. Based on the collected numbers, we empirically set the threshold $t = 10$: indeed, 24 out of 32 certificates use a number of SANs which is lower than this threshold, hence it is common practice to avoid a more widespread certificate reuse.

The complete picture of this part of our analysis is summarized by the scores in the "Certificate" column at [6]. The numbers confirm the good state of the certificate ecosystem of Italian university websites. Though EV certificates are still underrepresented, DV certificates are not used on website homepages and 22 out of 32 certificates (69%) were awarded the extra points granted by the limited amount of reuse on multiple domains.

5 Cryptographic Implementations

5.1 Security Practices

Even when HTTPS is up and running, cryptographic flaws in the implementation of TLS may break its intended security guarantees. Researchers identified several attacks against TLS in the past, which may even lead to the disclosure of the cryptographic keys used to protect the HTTP traffic [3, 4, 18]. In recent work, our research group developed a tool which automates the detection of TLS vulnerabilities exploitable on modern clients and estimates their impact on web application security [5]. Part of the tool has been recently integrated in the Discovery service of Cryptosense and is publicly available online.[6]

[6] `https://discovery.cryptosense.com/`

Number of SANs	Number of Websites	Frequency
1	6	18.8%
2	9	28.1%
3	2	6.3%
4	3	9.4%
6	1	3.1%
7	3	9.4%
19	1	3.1%
29	1	3.1%
31	1	3.1%
49	1	3.1%
71	1	3.1%
85	1	3.1%
95	1	3.1%
247	1	3.1%

Table 1: Frequency table for the observed number of SANs

Given a domain name like `www.unive.it`, the tool performs a cryptographic assessment of all the hosts to which `www.unive.it` might resolve. Moreover, the tool also analyzes all the hosts to which any other sub-domain of `unive.it` might resolve, since other web applications might be hosted there. Finally, the security analysis is done on all the hosts from which the web page at `www.unive.it` is including content, e.g., scripts and stylesheets. Cryptographic vulnerabilities on such hosts might seriously undermine page integrity and introduce confidentiality breaches, despite the adoption of the HTTPS protocol.

The output of the tool builds on the following taxonomy of insecure channels:

1. *Leaky channels*: established with servers vulnerable to confidentiality attacks, which give the attacker the ability to decrypt the network traffic;

2. *Tainted channels*: susceptible to man-in-the-middle attacks, which give the attacker the ability to decrypt and arbitrarily modify the network traffic. Observe that tainted channels are also leaky by definition;

3. *Partially leaky channels*: suffering from side-channels, which give the attacker the ability to disclose selected "small" secrets (like cookies) over time. Observe that leaky and tainted channels also qualify as partially leaky.

5.2 Security Measurement

We assign to each website a score from 0 to 4 based on the following system:

- 1 point: no (partially) leaky channel on the main domain or domains from which resources are loaded from the homepage. This ensures that network attackers cannot leverage cryptographic flaws to disclose information on the homepage;

- 1 point: no tainted channel on the main domain or domains from which resources are loaded from the homepage. This ensures that network attackers cannot exploit cryptographic flaws to undermine the integrity of the homepage;

- 1 point: no (partially) leaky channel on sub-domains. This is important at the very least to ensure the confidentiality of domain cookies;
- 1 point: no tainted channel on sub-domains. This is important at the very least to ensure the confidentiality and integrity of domain cookies.

Notice that, since tainted channels are also (partially) leaky, integrity violations effectively weigh twice as much as confidentiality violations in our scoring system.

5.3 Experimental Results

At the end of our measurement, we analysed 2,654 domains. In particular, we scanned 2,601 sub-domains of Italian universities and 53 external domains from which resources are loaded on the homepage of the analysed websites, the large majority of these being well-known analytics or library providers. This is already an interesting observation, because it shows that the attack surface is mostly under the control of the universities and does not significantly depend upon the security settings of external entities. This is in stark contrast with what we observed in general websites from the Alexa list [5]. On the other hand, the average number of sub-domains analysed on each website is 83: this is in line with our previous study, where we emphasized the growing importance of sub-domains on web application security.

A positive result of our analysis is that, as long as we focus on the main domain and the homepage of the analysed websites, the quality of the cryptographic deployment is very high. We were not able to find any cryptographic flaw which could affect the confidentiality and the integrity guarantees of the homepages. Unfortunately, the picture is way less positive when we turn our attention to sub-domains, given the size of the attack surface. In particular, we found tainted channels on at least one sub-domain on 15 out of 32 analyzed main domains. The key culprits of this are vulnerabilities enabling attacks like DROWN [3] and ROBOT [4]. The summary of our findings is given by the scores in the "Cryptography" column at [6].

6 Closing Remarks

We draw the key conclusions of our study, discussing ethical considerations and limitations.

6.1 Italian University Websites: Secure or Not?

To understand the state of the HTTPS deployment on the Italian university websites, we take a final look at the collected data [6] and assign a final score to each website as follows:

1. *Insecure* (6 sites): these websites got a score lower than 5 in the "Activation" column or could not be accessed over HTTPS. The implication is that these sites do not satisfy even minimal security requirements, because they suffer from confidentiality flaws even against passive network attackers;
2. *At risk* (25 sites): these websites got a score of 5 in the "Activation" column. They do provide a minimal level of protection, in particular against passive network attackers, but they are vulnerable to active tampering which can bypass HTTPS (SSL stripping);
3. *Acceptable* (2 sites): these websites got a score of at least 7 in the "Activation" column, but got a score lower than 6 in the "Certificate" column or a score lower than 4 in the "Cryptography" column. Though these sites implement some safeguards against active

network attackers, they also take unnecessary risks in their certificate management, e.g., by abusing of certificate reuse, or suffer from crytographic flaws which can be exploited by sophisticated attacks;

4. *Close to secure* (1 site): we put in this category all the other websites. We do not call this category "secure", because none of the analysed websites implement state-of-the-art protection mechanisms against all the considered threats.

In the end, our analysis shows that the current state of the HTTPS deployment of Italian university websites is unsatisfactory, because active network attackers have easy life on the very large majority of the analysed sites. However, our analysis also has positive implications, because it shows a reasonable state of health of the certificate and cryptographic ecosystems. This means that many websites can improve their security with limited effort by exclusively working at the web application layer. In particular, we observe that activating HSTS on the 25 websites marked as at risk would lead to the following improvement: 16 websites would be deemed acceptable and 9 websites would be considered close to secure (or even better). We hope this observation will encourage site operators to take actions to improve the current state of protection.

6.2 Ethics and Limitations

Due to both legal and ethical reasons, our analysis of TLS vulnerabilities in the wild was limited to an unintrusive scan based on the use of publicly available tools. The exploitability of the discovered vulnerabilities was exclusively judged through a systematic analysis of the output of those tools, as discussed in previous work [5]. The analyses of HTTPS activation and HTTPS certificates do not pose legal or ethical implications.

A limitation of our analysis is that it considers the homepage of the website as the key entry point. Though this makes sense from the perspective of a navigation session, it is possible that security-sensitive services are deployed on external domains which we did not analyse. Notice, however, that we provided a reasonable coverage of the security guarantees of sub-domains. A second limitation of our study is that we did not authenticate to the analysed websites, because we do not own valid access credentials for them. Finally, it is worth mentioning that our analysis is entirely focused on network attackers: we leave the study of security mechanisms designed to prevent web attacks to future work.

We are making plans to responsibly disclose all our results with the technical staff of the universities involved in our study. For now, we contacted our own IT office at Ca' Foscari and convinced them to activate HSTS on `www.unive.it`. We hope we will be just as successful with further disclosures.

7 Related Work

Many papers presented attacks exploiting cryptographic vulnerabilities in TLS, like DROWN [3], ROBOT [4] and TLS-POODLE [18]. The presence of such flaws in the wild is well-known and constantly measured by organizations like Qualys.[7] In this paper, we focused on the analysis of cryptographic vulnerabilities in the Italian university websites, which is an important and interesting aspect for our country and the ITASEC community. Most importantly, our analysis

[7] https://www.ssllabs.com/ssl-pulse/

leverages a research tool which only captures *exploitable* vulnerabilities still working on modern clients and are thus particularly relevant from a security perspective [5].

Besides cryptographic vulnerabilities, there might be several other reasons why HTTPS deployments turn out to be susceptible to attacks [14]. SSL stripping is a prominent example of an application layer attack against HTTPS [17]. To prevent it, browsers had to implement a new security mechanism in the form of HSTS. Kranch and Bonneau measured the HSTS adoption in the wild in 2015 and found it very limited; also, they observed that many configurations turn out to be insecure [13]. Luckily, anecdotal evidence shows that HSTS has been gaining traction over the years: for example, a recent small-scale session security study on 20 popular sites found that more than a half of the analyzed sites made use of HSTS [8]. Other studies on HTTPS security focused on the certificate ecosystem [11, 1, 15] and certificate errors in particular [2].

The present paper positions itself in the popular research line of security measurements of the Web. For example, previous work analyzed the security of European sites [20] and Chinese sites [10]. Other papers, instead, focused on the adoption of selected web security mechanisms like Content Security Policy [7], Cross Origin Resource Sharing [9] and postMessage [19]. We would like to expand our security analysis to also cover these important aspects.

References

[1] Josh Aas, Richard Barnes, Benton Case, Zakir Durumeric, Peter Eckersley, Alan Flores-López, J. Alex Halderman, Jacob Hoffman-Andrews, James Kasten, Eric Rescorla, Seth D. Schoen, and Brad Warren. Let's encrypt: An automated certificate authority to encrypt the entire web. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019, London, UK, November 11-15, 2019*, pages 2473–2487, 2019.

[2] Mustafa Emre Acer, Emily Stark, Adrienne Porter Felt, Sascha Fahl, Radhika Bhargava, Bhanu Dev, Matt Braithwaite, Ryan Sleevi, and Parisa Tabriz. Where the wild warnings are: Root causes of chrome HTTPS certificate errors. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, pages 1407–1420, 2017.

[3] Nimrod Aviram, Sebastian Schinzel, Juraj Somorovsky, Nadia Heninger, Maik Dankel, Jens Steube, Luke Valenta, David Adrian, J. Alex Halderman, Viktor Dukhovni, Emilia Käsper, Shaanan Cohney, Susanne Engels, Christof Paar, and Yuval Shavitt. DROWN: Breaking TLS Using SSLv2. In *Proceedings of the 25th USENIX Security Symposium (USENIX Security 16)*, pages 689–706. USENIX Association, 2016.

[4] Hanno Böck, Juraj Somorovsky, and Craig Young. Return Of Bleichenbacher's Oracle Threat (ROBOT). Cryptology ePrint Archive, Report 2017/1189, 2017. Online, cit. [2018-10-29].

[5] Stefano Calzavara, Riccardo Focardi, Matús Nemec, Alvise Rabitti, and Marco Squarcina. Postcards from the post-http world: Amplification of HTTPS vulnerabilities in the web ecosystem. In *2019 IEEE Symposium on Security and Privacy, SP 2019, San Francisco, CA, USA, May 19-23, 2019*, pages 281–298, 2019.

[6] Stefano Calzavara, Riccardo Focardi, Alvise Rabitti, and Lorenzo Soligo. Results of our security assessment. https://secgroup.dais.unive.it/https_universities/, 2020.

[7] Stefano Calzavara, Alvise Rabitti, and Michele Bugliesi. Semantics-based analysis of content security policy deployment. *TWEB*, 12(2):10:1–10:36, 2018.

[8] Stefano Calzavara, Alvise Rabitti, Alessio Ragazzo, and Michele Bugliesi. Testing for integrity flaws in web sessions. In *Computer Security - ESORICS 2019 - 24th European Symposium on Research in Computer Security, Luxembourg, September 23-27, 2019, Proceedings, Part II*, pages 606–624, 2019.

[9] Jianjun Chen, Jian Jiang, Hai-Xin Duan, Tao Wan, Shuo Chen, Vern Paxson, and Min Yang. We still don't have secure cross-domain requests: an empirical study of CORS. In *27th USENIX Security Symposium, USENIX Security 2018, Baltimore, MD, USA, August 15-17, 2018*, pages 1079–1093, 2018.

[10] Ping Chen, Nick Nikiforakis, Lieven Desmet, and Christophe Huygens. Security analysis of the chinese web: How well is it protected? In *Proceedings of the 2014 Workshop on Cyber Security Analytics, Intelligence and Automation, SafeConfig '14, Scottsdale, Arizona, USA, November 3, 2014*, pages 3–9, 2014.

[11] Zakir Durumeric, James Kasten, Michael Bailey, and J. Alex Halderman. Analysis of the HTTPS certificate ecosystem. In *Proceedings of the 2013 Internet Measurement Conference, IMC 2013, Barcelona, Spain, October 23-25, 2013*, pages 291–304, 2013.

[12] Adrienne Porter Felt, Richard Barnes, April King, Chris Palmer, Chris Bentzel, and Parisa Tabriz. Measuring HTTPS adoption on the web. In *26th USENIX Security Symposium, USENIX Security 2017, Vancouver, BC, Canada, August 16-18, 2017*, pages 1323–1338, 2017.

[13] Michael Kranch and Joseph Bonneau. Upgrading HTTPS in mid-air: An empirical study of strict transport security and key pinning. In *22nd Annual Network and Distributed System Security Symposium, NDSS 2015, San Diego, California, USA, February 8-11, 2015*, 2015.

[14] Katharina Krombholz, Wilfried Mayer, Martin Schmiedecker, and Edgar R. Weippl. "i have no idea what i'm doing" - on the usability of deploying HTTPS. In *26th USENIX Security Symposium, USENIX Security 2017, Vancouver, BC, Canada, August 16-18, 2017*, pages 1339–1356, 2017.

[15] Deepak Kumar, Zhengping Wang, Matthew Hyder, Joseph Dickinson, Gabrielle Beck, David Adrian, Joshua Mason, Zakir Durumeric, J. Alex Halderman, and Michael Bailey. Tracking certificate misissuance in the wild. In *2018 IEEE Symposium on Security and Privacy, SP 2018, Proceedings, 21-23 May 2018, San Francisco, California, USA*, pages 785–798, 2018.

[16] Salvatore Manfredi, Silvio Ranise, and Giada Sciarretta. Lost in tls? no more! assisted deployment of secure TLS configurations. In *Data and Applications Security and Privacy XXXIII - 33rd Annual IFIP WG 11.3 Conference, DBSec 2019, Charleston, SC, USA, July 15-17, 2019, Proceedings*, pages 201–220, 2019.

[17] Moxie Marlinspike. New Tricks for Defeating SSL in Practice, 2009. Online, cit. [2019-02-12].

[18] Bodo Möller, Thai Duong, and Krzysztof Kotowicz. This POODLE Bites: Exploiting The SSL 3.0 Fallback, 2014. Online, cit. [2018-10-29].

[19] Sooel Son and Vitaly Shmatikov. The postman always rings twice: Attacking and defending postmessage in HTML5 websites. In *20th Annual Network and Distributed System Security Symposium, NDSS 2013, San Diego, California, USA, February 24-27, 2013*, 2013.

[20] Tom van Goethem, Ping Chen, Nick Nikiforakis, Lieven Desmet, and Wouter Joosen. Large-scale security analysis of the web: Challenges and findings. In *Trust and Trustworthy Computing - 7th International Conference, TRUST 2014, Heraklion, Crete, Greece, June 30 - July 2, 2014. Proceedings*, pages 110–126, 2014.