A Dangerous Mix: Large-scale analysis of mixed-content websites

Ping Chen, Nick Nikiforakis, Christophe Huygens, and Lieven Desmet

iMinds-DistriNet, KU Leuven
3001 Leuven, Belgium
{firstname.lastname}@cs.kuleuven.be

Abstract In this paper, we investigate the current state of practice about mixed-content websites, websites that are accessed using the HTTPS protocol, yet include some additional resources using HTTP. Through a large-scale experiment, we show that about half of the Internet's most popular websites are currently using this practice and are thus vulnerable to a wide range of attacks, including the stealing of cookies and the injection of malicious JavaScript in the context of the vulnerable websites. Additionally, we investigate the default behavior of browsers on mobile devices and show that most of them, by default, allow the rendering of mixed content, which demonstrates that hundreds of thousands of mobile users are currently vulnerable to MITM attacks.

1 Introduction

Internet users today rely on HTTPS (HTTP over SSL/TLS) for secure communication of sensitive data. While websites are migrating to HTTPS, attackers are also shifting efforts to break the TLS communication. Complementary to protocol and infrastructure vulnerabilities in TLS and HTTPS, as illustrated by recent attacks such as CRIME [19] and Lucky 13 [10], attackers can also exploit mixed-content vulnerabilities to compromise TLS-protected websites.

In mixed-content websites, the webpage is delivered to the browser over TLS, but some of the additional content, such as images and scripts, are delivered over a non-secured HTTP connection. These non-secured communications can be exploited by network attackers to gain access to wide set of capabilities ranging from access to cookies and the forging of arbitrary requests, to the execution of arbitrary JavaScript code in the security context of the TLS-protected website.

Desktop browsers are recently catching up to mitigate this vulnerability, but the large majority of browsers on mobile devices, such as smartphones and tablets, leave the end-user unprotected against this type of attack. This is worsened by the fact that it is typically pretty straightforward to launch an active network attack against a mobile user (e.g. via setting up a fake wireless hotspot).

In this paper, we report on an in-depth assessment of the state-of-practice with respect to mixed-content vulnerabilities. In particular, the main contributions of this paper are the following: (1) We study the different types of

mixed-content inclusions, and assess their security impact. (2) We present a detailed analysis of mixed-content inclusions over the Alexa top 100,000 Internet domains, showing that 43% of the Internet's most popular websites suffer from mixed-content vulnerabilities. (3) We document the behavior of mobile browsers in the face of mixed-content inclusions. (4) We enumerate the best practices as well as novel mitigation techniques against mixed-content inclusions for browsers, website owners and content providers.

2 Problem statement

It is well-known that HTTP is vulnerable to eavesdropping and man-in-the-middle (MITM) attacks, and HTTPS is designed to precisely prevent these attacks by adding the security capabilities of SSL/TLS to HTTP. SSL/TSL enables authentication of the web server, and provides bidirectional encryption of the communication channel between the client and server. Apart from the attacks against the SSL/TLS protocol, we show that attackers can also exploit mixed-content vulnerabilities to compromise TLS-protected websites.

The attacker model used in this paper, is the *active network attacker*. The active network attacker positions himself on a network between the web browser and the web server, and is able to intercept and tamper with the network traffic passing by. The attacker can read, modify, delete, and inject HTTP requests and responses, but he is not able to decipher encrypted information, nor impersonate an HTTPS endpoint without a valid TLS certificate.

In mixed content (also known as non-secure/insecure content) websites, the web page is delivered to the browser over TLS, but some of the additional content, such as images and scripts, are directly delivered over a non-secured HTTP connection from the content provider towards the web browser. Although the active network attacker can not attack the web page delivery over HTTPS, he can still compromise the TLS-enabled website by compromising any of the additional resources that are loaded over HTTP.

3 Impact of mixed content attacks

Five specific types of mixed content are studied in this paper: Image, iframe, CSS, JavaScript and Flash. The impact of different types of mixed content attack can be categorized as follows:

– **Cookie stealing:** When a browser requests mixed content, it may include cookies associated with the content provider, which allows the attacker to obtain the cookies. Moreover, if the content provider and the TLS-protected website using mixed content happen to be on the same domain, sensitive cookies used over HTTPS can get exposed to the attacker via a HTTP request, unless the cookie is protected by the "secure" flag.

– **Request forgery:** As mixed content is requested over HTTP, the attacker can manipulate the HTTP requests and responses and use them to trigger or forge arbitrary HTTP requests, which may lead to certain variants of SSL-Stripping [16] and Cross-Site Request Forgery (CSRF) [12] attacks.
– **DOM data leakage:** Mixed content may leak confidential data that is displayed as part of the HTTPS webpage. For example, mixed-CSS content can be used to obtain sensitive data in the DOM via scriptless attacks [14]: CSS selectors can match against particular content in the DOM, and leak the result of the test by fetching a web resource (e.g. image) monitored by the attacker.
– **JavaScript execution:** For mixed-JavaScript and mixed-Flash content, the attacker can inject arbitrary JavaScript code that will be executed in the context of the HTTPS website using the mixed content. This allows the attacker to run arbitrary JavaScript code as if it was originating from the TLS-protected site, and access a variety of security-sensitive JavaScript APIs. Moreover, the attacker can inject malicious payloads, such as the BeEF framework [2], to take over the user's browser and launch various attacks.

The various types of mixed content and their impact are summarized in Table 1.

Type	Cookie stealing	Request forgery	DOM data leakage	JavaScript execution
Image	x	x		
iframe	x	x		
CSS	x	x	x	
JavaScript	x	x	x	x
Flash	x	x	x	x

Table 1: Impact of mixed content attacks

4 Data collection

In this section, we describe the setup and results of our large-scale data collection experiment.

4.1 Crawling experiment

Starting with Alexa's list of 100,000 most popular domains, we first filter out the domains that are not available over TLS. Next, we use the Bing Search API [3] to automatically query for a set of 200 HTTPS page URLs for each website.

To discover mixed-content inclusions on TLS-enabled websites, we apply the approach as described in [18]: the headless HtmlUnit browser is used to visit the page URLs, and the HTTPS pages are analyzed to locate mixed-content inclusions. HtmlUnit is able to execute the JavaScript code similar to a real browser, and as such, it can detect dynamically-included mixed content.

4.2 Data collection results

With the aforementioned approach, we extracted 18,526 HTTPS websites from Alexa top 100,000 Internet domains, and in total 481,656 HTTPS pages are crawled, with an average of 26 HTTPS pages per website.

From the crawled HTTPS websites, 7,980 (43%) were found to have at least one type of mixed content. This means that almost half of the HTTPS protected websites, are vulnerable to one or more of the attacks mentioned in the previous sections. In total, 620,151 mixed-content inclusions were found through our experiment, which maps to 191,456 mixed-content files and 74,946 HTTPS webpages. Table 2 gives an overview of the distribution of mixed-content inclusions. Image and JavaScript are the most included mixed content types, with 30% and 26% of the HTTPS websites using them respectively, while mixed-Flash content is much less used. As for the distribution over remote and local inclusions for each mixed content type, mixed iframe, JavaScript, and Flash content is mostly served by remote providers, while the majority of mixed Image and CSS inclusions are locally included.

	# Inclusions	% remote inclusions	# Files	# Webpages	% Websites
Image	406,932	38%	138,959	45,417	30%
iframe	25,362	90%	15,227	15,419	14%
CSS	35,957	44%	6,680	15,911	12%
JavaScript	150,179	72%	29,952	45,059	26%
Flash	1,721	62%	638	1,474	2%
Total	620,151	47%	191,456	74,946	43%

Table 2: Overview of distribution of mixed-content inclusions

To better understand the risks associated with websites using different types of mixed content, we calculate the percentage of websites that are exposed to different levels of attacks as shown in Figure 1. The calculation is based on the impact analysis for each mixed content type (Table 1), which groups different types of mixed-content inclusions according to the associated attacks. Figure 1 shows that 27% websites are exposed to attacks up to "JavaScript execution", by including mixed JavaScript or Flash content.

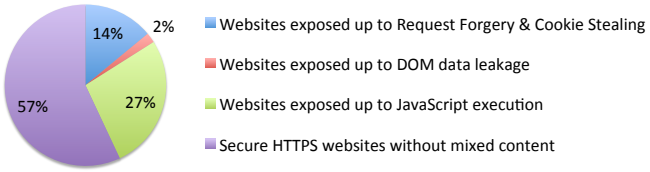

Figure 1. Percentage of TLS-enabled websites vulnerable to different attacks

5 Discussion

In this section, we discuss some characteristics of mixed content and the websites including them, as discovered in our experiment. First, we identify the distribution of websites having mixed content over different categories, and then present some examples of important websites having mixed-JavaScript content. Second, we investigate the availability of mixed content files over HTTPS.

5.1 Websites having mixed content

To better understand the websites having mixed content, we categorize the websites based on McAfee's TrustedSource Web Database [17]. Most of the visited HTTPS websites are categorized into 88 categories, with 1,181 websites remaining uncategorized. The majority of visited websites (66%) can be categorized into 10 popular categories. As shown in Figure 2, The 'Government/Military' websites are doing better than websites in all other categories, with "only" 31% of them websites having mixed content. 38% of 'Finance/Banking' websites having mixed content, which is worrisome, since these websites contain valuable information and are typically the targets of attackers.

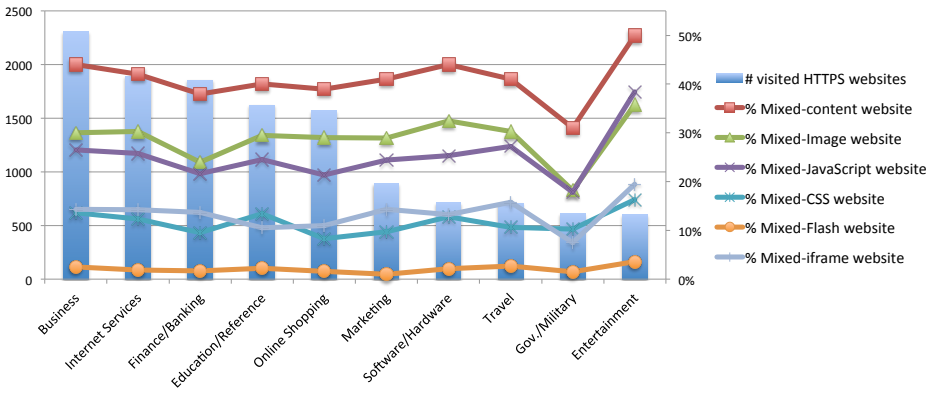

Figure 2. Distribution of websites having mixed content over top 10 categories

For the 74,946 HTTPS pages having mixed content, we check whether these pages have an equivalent HTTP version of the same content. While most of them do have an HTTP version, 9,792 (11%) pages are only served over HTTPS, and these "HTTPS-Only" pages map to 1,678 (9%) HTTPS websites. We consider it likely that these "HTTPS-Only" pages contain more sensitive data and should be more secure, compared to those pages having the same content served over HTTP. Thus, mixed-content inclusions on "HTTPS-Only" pages can have more severe consequences when successfully exploited.

Table 3 lists ten examples of "HTTPS-Only" pages (selected from Alexa's top 1,000 websites) having mixed-JavaScript content. These pages provide important functionalities like "Account Signup" , "Account Login" , and "Password Recovery", all of which process sensitive user information and thus can lead to user-data leakage if the mixed-JavaScript content is intercepted by an attacker.

HTTPS-Only pages	Functionality
`www.aweber.com/signup.htm`	Account Signup
`www36.verizon.com/callassistant/signin.aspx`	Account Login
`secure.pornhublive.com/forgot-password/`	Password Recovery
`euw.leagueoflegends.com/account/recovery/password`	Password Recovery
`ww15.itau.com.br/privatebank/contatoprivate/en/index.aspx`	Contact Form
`dv.secure.force.com/applyonline/Page1?brand=ccn`	Application Form
`www.tribalfusion.com/adapp/forms/contactForm.jsp`	Contact Form
`support.makemytrip.com/ForgotPassword.aspx`	Password Recovery
`jdagccc.custhelp.com/app/utils/create_account/red/1`	Account Signup
`ssl6.ovh.net/~pasfacil/boutiquemedievale/login.php`	Account Login

Table 3: Ten example "HTTPS-Only" pages having mixed-JavaScript content

Of the 1,678 HTTPS websites that have "HTTPS-Only" pages, we found 97 websites that are using HTTP Strict Transport Security (HSTS) policy [15], which indicates that these websites are making use of the latest protection technology for ensuring the use of SSL, but they still fail to achieve their goal by including mixed content from insecure channels.

5.2 Providers of mixed-content files

For the total of 191,456 mixed-content files, we check whether the providers serve these files over a secure HTTPS channel next to their insecure HTTP versions. While the majority of mixed JavaScript, iframe and CSS content files are available over HTTPS, the percentage of mixed Image content files available over HTTPS is significantly less, as shown in Table 4. Though website owners should be responsible for the mixed content issue, the data in Table 4 indicates that blaming them is too simplistic, since it ignores the fact that approximately half of the mixed content files are only available over HTTP.

Type	# Files	% HTTPS-Available	Type	# Files	% HTTPS-Available
Image	138,959	40%	JavaScript	29,952	58%
iframe	15,227	77%	Flash	638	46%
CSS	6,680	60%	Total	191,456	47%

Table 4: Percentage of "HTTPS-Available" files, per mixed content type

6 Mixed content mitigation techniques

In this section, we investigate and enumerate protection techniques that can be used for browsers, TLS-protected websites, and content providers, to mitigate the issue of insecure inclusion of content.

6.1 Browser Vendors

Blocking mixed content at the browser-level, is the most straightforward way to mitigate the mixed content issue. While most desktop browsers have developed a mixed-content blocker to protect users against insecure content, mobile browsers strongly lag behind on this, despite the fact that mobile browsing is becoming increasingly important to users. According to recent statistics from StatCounter, the market share of mobile browsing has almost tripled in the last two years, having reached 16.08% in June 2013.

As part of our study, we investigated how all the major mobile browsers for Android, iOS, Windows Phone and Windows RT platform handle mixed content – shown in Table 5. We unfortunately discovered that most of them do not have a mixed-content blocker, with the exception of Chrome for Android and IE 10 Mobile which protect the user against mixed content. Firefox for Android plans to have a mixed-content blocker in a future release [1].

Platform	Mobile browser	blocked ?	secure padlock shown ?
Google Android 4.2	Chrome 28	Yes	with a yellow triangle
	Firefox 23	No	No
	Android browser	No	open padlock
	Opera Mobile 12	No	No
Apple iOS 6.1	Safari 6	No	No
	Chrome 28	No	with a yellow triangle
	Opera Mini 7	No	No
Windows Phone/RT 8	IE 10 Mobile	Yes	No

Table 5: Mobile browsers' behavior towards mixed content

With respect to desktop browsers, Internet Explorer (IE) is the first browser that detected and blocked mixed content with IE 7, released in 2006. When mixed content is detected, the browser warns the user and allows her to choose whether insecure content should be loaded [7]. Many users, however, would probably click "Yes", rendering the mixed-content blocker useless [21]. An elegant way to handle mixed content would be to silently block mixed content without prompting the users. This approach has been chosen in Chrome (version 21+) [8], Internet Explorer (version 9+) [4], and the recently released Firefox 23 [9]. Safari and Opera browsers do not currently have a mixed-content blocker, which means that about 10% of desktop users are still exposed to the dangers of mixed content.[1]

[1] Safari and Opera each owns 8.39% and 1.03% market share respectively, according to the statistics of usage share of desktop browsers for June 2013 from StatCounter[6].

Chrome, IE, and Firefox all have a mixed-content blocker, but they only block mixed iframe, CSS, JavaScript and Flash content, and mixed Image content is left out. An interesting fact is that mixed-Image content is blocked in IE 7 and IE 8, but it is not blocked in IE 9 and IE 10. Since mixed Image and iframe content technically have the same impact which may lead to attacks "Request forgery" and "Cookie stealing", we recommend all browsers vendors to block all types of mixed content, thus completely eliminating the mixed content issue from the browser side. While this move would likely break some insecurely-coded websites, the security benefits of mixed-content-blocking definitely outweigh the temporary frustration of users when they encounter some websites that do not properly work.

6.2 Website owners

TLS-protected websites can explicitly opt-in to only include content from secure channels. As shown in Table 4, 47% of the mixed-content files are not correctly included, since the secure version of the resources exist and could thus be used. For the remaining set of mixed-content files that do not have a secure version, the resources can be cached locally, or proxied using their own SSL server.

To provide better security, a website using HTTPS can use a combination of the HTTP Strict Transport Security (HSTS) and Content Security Policy (CSP) [20] protocols, as illustrated in Listing 1.1.

Listing 1.1. Protecting TLS-protected sites via HSTS and CSP

```
1 Strict-Transport-Security: max-age=86400; includeSubDomains
2 Content-Security-Policy: default-src https:; \
3 script-src https: 'unsafe-inline'; \
4 style-src https: 'unsafe-inline'
```

First, HSTS can be used to guarantee that webpages are only served over HTTPS by forcing a compliant browser to only issue HTTPS requests for that website (line 1). By enforcing the HSTS policy, it can prevent SSL-stripping attacks [16]. Second, CSP can be used to detect mixed content violations (in report-only mode), and to actively block mixed content by specifying that only secure resources are allowed to be included (line 2). Notice that in this example the *unsafe-inline* directives are added to preserve temporary compatibility (lines 3-4), but website owners are encouraged to fully embrace the CSP technology so that they achieve full protection and no longer need these unsafe directives.

6.3 Resource providers

Resource providers can also mitigate the mixed content issue by offering content over HTTPS (even only over HTTPS). Moreover, resource provider can also use HSTS to migrate non-HTTPS resources automatically and secure to HTTPS version. Notice, however, that not all browsers have support for HSTS policies (e.g., IE 10 and Safari 6), and that HSTS inherently has a bootstrapping problem during a browser's very first visit to an HSTS website. During this first request, an active network attacker can strip the HSTS header and circumvent this protection technique.

7 Related work

To the best of our knowledge, this paper is the first that attempts to systematically discover the current state of practice of mixed content and uncover the various types of mixed-content inclusions and how they could be used to attack users and services.

While HTTPS is widely used for securing web communications, many attacks on HTTPS have been reported over the years [13]. Apart from the exploit of cryptographic weaknesses and design flaws in the SSL and TLS protocols, e.g., CRIME [19] and Lucky 13 [10], the incorrect adoption and configuration of HTTPS by websites, may also allow attackers to bypass HTTPS. For example, websites not using an HSTS policy are vulnerable to SSL-Stripping attacks [16]. According to the latest surveys of August 2013, about 76% of HTTPS websites have security issues with their SSL implementations [5].

Web browsers also play an important role in web security, since they can automatically handle many sensitive, HTTPS security decisions, and provide security indicators through their user interfaces. While, in the last few years, desktop browsers, in response to various attacks like XSS and CSRF, have been substantially hardened, mobile browsers have unfortunately not caught up. Mobile browsers, when compared to desktop browsers, have less support for displaying HTTPS connection details, and for the warning about mixed content [11].

8 Conclusion

When migrating to HTTPS, many websites fail to fully update their applications, resulting in mixed-content inclusion, which can render the HTTPS protection useless. In this paper, we show that there is a considerable number of TLS-protected websites that currently have mixed content. We also observed that, while the desktop browsers are catching up to mitigate this issue, most mobile browsers do not have protections against mixed content yet, despite the increasing popularity of mobile devices. Since users are entering into the "Post-PC era", i.e., prefering mobile devices for regular Internet browsing and even for sensitive online transactions, it is important for mobile browsers to develop a mixed-content blocker, as several desktop browsers have already done. To handle this transitory phase, we investigated and reported the best practices for websites owners and content providers which can be used to counter the issue and protect their users against MITM attacks.

Acknowledgements This research is partially funded by the Research Fund KU Leuven, iMinds, IWT, and by the EU FP7 projects WebSand, NESSoS and STREWS. With the financial support from the Prevention of and Fight against Crime Programme of the European Union (B-CCENTRE).

References

1. Add support for Mixed Content Blocking - Android. https://bugzilla.mozilla.org/show_bug.cgi?id=860581.
2. BeEF - The Browser Exploitation Framework Project. http://beefproject.com/.
3. Bing Search API. http://datamarket.azure.com/dataset/bing/search.
4. "only secure content is displayed" notification in internet explorer 9 or later. http://support.microsoft.com/kb/2625928.
5. SSL Pulse. https://www.trustworthyinternet.org/ssl-pulse/.
6. StatCounter. http://statcounter.com/.
7. Internet Explorer 8 Mixed Content Handling. http://msdn.microsoft.com/en-us/library/ee264315(v=vs.85).aspx, 2009.
8. Ending mixed scripting vulnerabilities. http://blog.chromium.org/2012/08/ending-mixed-scripting-vulnerabilities.html, 2012.
9. Mixed content blocking enabled in firefox 23! https://blog.mozilla.org/tanvi/2013/04/10/mixed-content-blocking-enabled-in-firefox-23/, 2013.
10. N.J. Al Fardan and K.G. Paterson. Lucky Thirteen: Breaking the TLS and DTLS Record Protocols. In *IEEE Symposium on Security and Privacy*, SP '13, pages 526–540, 2013.
11. Chaitrali Amrutkar, Patrick Traynor, and Paul C. van Oorschot. Measuring ssl indicators on mobile browsers: extended life, or end of the road? In *Proceedings of the 15th International Security Conference*, ISC '12, pages 86–103. Springer, 2012.
12. Adam Barth, Collin Jackson, and John C. Mitchell. Robust defenses for cross-site request forgery. In *Proceedings of the 15th ACM conference on Computer and communications security*, CCS '08, pages 75–88, New York, NY, USA, 2008. ACM.
13. Jeremy Clark and Paul C. van Oorschot. SoK: SSL and HTTPS: Revisiting Past Challenges and Evaluating Certificate Trust Model Enhancements. In *IEEE Symposium on Security and Privacy*, SP '13, pages 511–525, 2013.
14. Mario Heiderich, Marcus Niemietz, Felix Schuster, Thorsten Holz, and Jörg Schwenk. Scriptless attacks: stealing the pie without touching the sill. In *Proceedings of the 2012 ACM conference on Computer and communications security*, CCS '12, pages 760–771, New York, NY, USA, 2012. ACM.
15. J. Hodges, C. Jackson, and A. Barth. HTTP strict transport security (HSTS). *IETF RFC*, 2012.
16. Moxie Marlinspike. New Tricks for Defeating SSL in Practice. *Blackhat*, 2009.
17. McAfee. TrustedSource Web Database. https://www.trustedsource.org/en/feedback/url.
18. Nick Nikiforakis, Luca Invernizzi, Alexandros Kapravelos, Steven Van Acker, Wouter Joosen, Christopher Kruegel, Frank Piessens, and Giovanni Vigna. You are what you include: large-scale evaluation of remote javascript inclusions. In *Proceedings of the 2012 ACM conference on Computer and communications security*, CCS '12, pages 736–747, New York, NY, USA, 2012. ACM.
19. Juliano Rizzo and Thai Duong. Crime: Compression ratio info-leak made easy. In *ekoparty Security Conference*, 2012.
20. Sid Stamm, Brandon Sterne, and Gervase Markham. Reining in the web with content security policy. In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 921–930, New York, NY, USA, 2010. ACM.
21. Joshua Sunshine, Serge Egelman, Hazim Almuhimedi, Neha Atri, and Lorrie Faith Cranor. Crying Wolf: An Empirical Study of SSL Warning Effectiveness. In *Proceedings of the 18th Usenix Security Symposium*, pages 399–416, 2009.