Web-platform security guide: Security assessment of the Web ecosystem

Philippe De Ryck, Lieven Desmet, Martin Johns and Frank Piessens

Abstract—**In this document, we report on the Web-platform security guide, which has been developed within the EC-FP7 project STREWS. Based on their research, the STREWS consortium argues that in order to strengthening the Internet (e.g. against pervasive monitoring), it is crucial to also strengthen the web application ecosystem, the de-facto Internet application platform.**

The Web security guide is the result of a broad security assessment of the current situation on the Web[1]. It looks at the Web ecosystem and provides a **timely and comprehensive web security overview**. It was written by the **STREWS Consortium**, that brings together a **unique set of expertise in Europe** to grasp the complexity of the Web platform and its security characteristics. It is unique because it brings together strong peers in academic web security research in Europe, a large European software vendor, and principal actors in standardisation activities in W3C and IETF, the predominant specification developing organisations for the Web.

The Web platform security guide consists of four parts, and looks as follows:

1) The first part gives a comprehensive overview of the current Web and the expected developments in the near future.
2) Based on the understanding of the Web ecosystem in the first part, the second part captures the breadth and complexity of the **Web security vulnerability landscape**.
   It describes the **Web assets** that are worth attacking and lists the capabilities attackers may have at their disposition and discusses the commonly-used **attacker models**.
3) In the third part, the **twenty most representative attack techniques** are discussed and analyzed, grouped in **seven high-level threat categories**.
   The guide presents and discusses the latest **state-of-the-art**, both from a **research perspective** as well as from a **standardization perspective**.
   Moreover, the guide provides a **catalogue of best practices** designed to mitigate the threats discussed, and to gradually improve the trustworthiness of web-enabled services.
4) Part four gives the **full Web security threat landscape** as an overview, indicates **upcoming challenges** resulting from the change of the web ecosystem and hints at some **interesting opportunities** for future research.

In the following paragraphs, we briefly highlight the most important contributions and key takeaways for each part.

PART I: Foundations of the Web platform

In the first part of the guide, we briefly recap and discuss the foundations of the Web ecosystem. The goal of this first part is to provide the reader with a basic understanding of the Web ecosystem, needed to understand the security assessment.

Over the last 25 years, the Web ecosystem went through a series of technological waves (as depicted in Figure 1), enriching the platform to the current level where it provides an attractive alternative to stand-alone applications (or even replacing the operating system itself). Evolutions in the Web platform include richer presentation capabilities (e.g., graphics, style sheets and multimedia tags), client-side state (cookies and storage), client-side interactivity (JavaScript, the DOM and a rich set of JavaScript APIs), as well as rich Internet Applications (such as Flash, ActiveX and Silverlight).

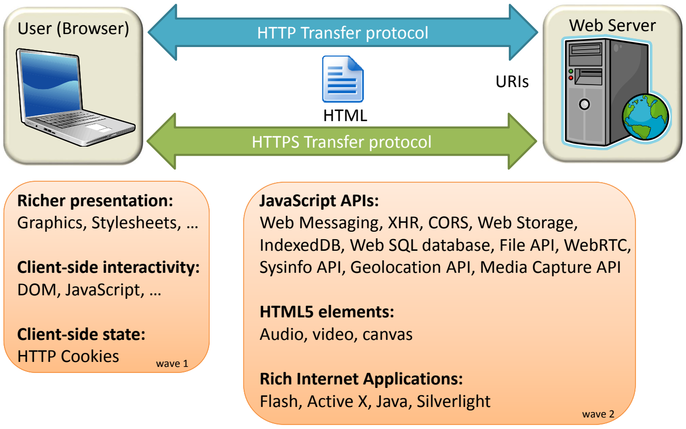


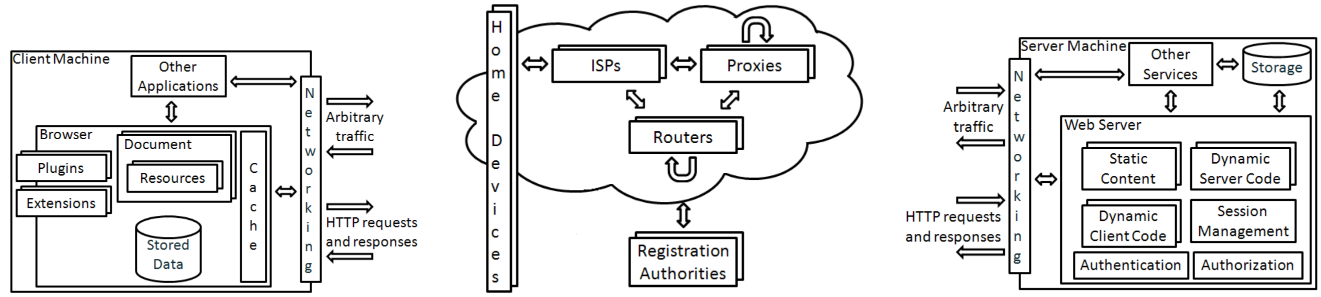
Fig. 1: The Web ecosystem went through a series of technological waves.

The resulting ecosystem is represented in Figure 2, and is discussed in more detail in part I of this security guide. Important in this context is that, although the Web ecosystem has grown substantially over the last two decades, the basic security model of the Web still strongly relies on the Same-Origin Policy (SOP) from the mid 1990s. Major changes to this model face prohibitive deployment obstacles, as the currently-deployed legacy of web applications relies on the legacy model's properties.

This tension between ever-increasing complexity and limited built-in security fuels a continuous arms race between attackers and defenders, as will be illustrated by the wide variety of attacks discussed later in this document.

After reading part I, we expect all readers to share a common level of understanding of the Web platform. To

[1] The Web-platform security guide can freely be downloaded at http://www.strews.eu/results/5-web-platform-security-guide



(a) Client Side

(b) Network

(c) Server Side

Fig. 2: Perspectives on the contemporary Web platform

serve the different audiences of the guide, the basic Web building blocks in part I are interleaved with some more advanced insights and additional pointers to aspects of the Web ecosystem, targeted to more advanced readers from industry and academia. For novice readers, the appendix discusses in more detail the underlying technologies of the Web, such as HTML, CSS, JavaScript and HTTP.

PART II: Threats to the Web platform

In part II, we identify the assets of the Web ecosystem, based on the model and concepts developed in part I, and enumerate the set of capabilities an attacker might have.

The assets are approached from an application-agnostic, technical point-of-view, but they can easily be correlated to actual business values once enriched with concrete application context. For instance, during the risk analysis of an E-Health web application, the *Server-side Content Storage* asset can be instantiated with electronic health records of the patients, and use their business value in the risk calculations.

The Web security platform guide identifies assets in the web infrastructure (i.e., the client and server machines), assets in the application (such as client-side application code and server-side application storage) and user-related assets (such as authentication credentials and personal information) (Figure 3).

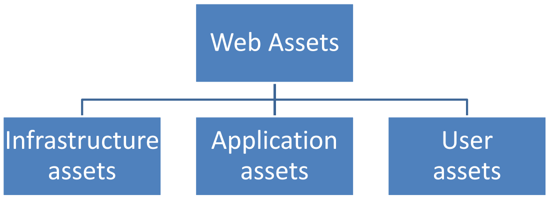


Fig. 3: Assets in the web infrastructure, assets in the application, and user-related assets

For each asset, we discuss its importance and the attacker's incentives for compromising this asset, and analyze how an asset can be compromised. In order to structure the variety of ways available to compromise an asset, we use trees to explicitly define high-level threats against an asset. These threats are intermediate steps that an attacker has to take in order to compromise an asset, and are typically common across different assets.

For instance, if an attacker wants to tamper with an *Application Transaction* (e.g. forge a new wire transfer in an online bank application), the attacker can do this by first compromising the *Client-side Application Code* asset, as depicted in Figure 4. Similarly, there are various ways to compromise the client-side code, and an attacker might choose to intercept and manipulate the network traffic in order to control the client-side application code.

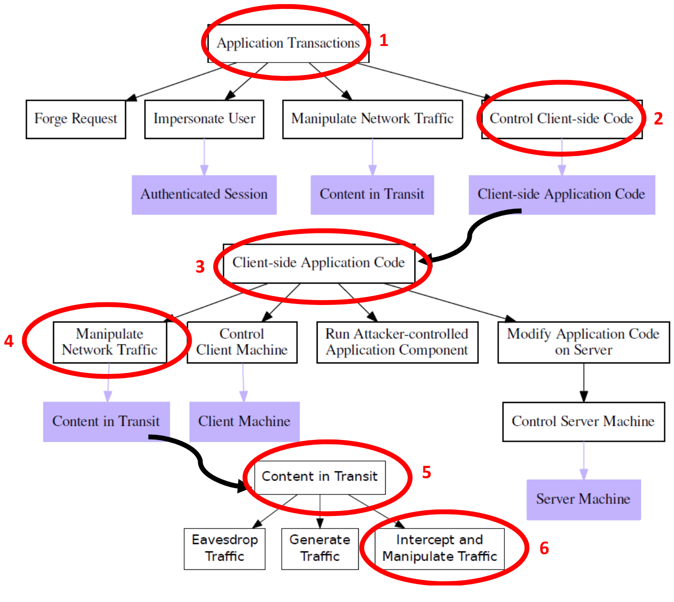


Fig. 4: Walking the asset tree.

To be able to define an attacker's power in an unambiguous way, the platform guide enumerates the set of primitive and disjoint *attacker capabilities*. Essentially, attacker capabilities are the lowest level of capabilities available to an attacker. Examples of such fine-grained attacker capabilities are *Host content under a registered domain* or *Send a well-formed request to the application*.

The attacker capabilities can be composed to give a specific threat model, if desired. Within the security guide, we discuss threat models used in academic web security literature since 2008 (such as the *web attacker*, the *related-domain attacker* and the *passive network attacker*). We explicitly decompose the threat models into attacker capabilities, and compare them against each other (as illustrated in Table I).

	Forum Poster	Web Attacker	Gadget Attacker	Related-domain Attacker	Passive Network Attacker	Active Network Attacker
Register Available Domain		★	★	★	★	★
Host Content under Registered Domain		★	★	★	★	★
Host Content under Existing Domain				★		
Register Valid Certificate for Domain Name		★	★	★	★	★
Respond to Legitimate Client Request		★	★	★	★	★
Send Well-formed Request to Application	★	★	★	★	★	★
Send Arbitrary Network Request to Server		★	★	★	★	★
Eavesdrop on Network Traffic					★	★
Generate Network Traffic						★
Intercept and Manipulate Network Traffic						★

TABLE I: An overview of academic threat models, decomposed into fine-grained attacker capabilities.

The main advantages of using attacker capabilities over threat models are that (1) they precisely define what technical capabilities an attacker has, and that (2) they can be more dynamically composed into new threat models that list the minimal set of capabilities needed to perform an attack.

PART III: Attacks on the Web platform

In the third part, the **web security vulnerability landscape** is constructed, by investigating how an attacker can execute the threats to compromise an asset. To do so, the threats identified in part II are grouped together in **seven high-level threat categories**:

1) Impersonating users
2) Forging requests
3) Attacking through the network
4) Controlling the client-side context
5) Attacking the client-side infrastructure
6) Directly attacking the web application
7) Violating the user's privacy

For each of the seven high-level threats, the **most representative attack techniques** have been selected, and are reported in more detail. Selection of the representative subset of attack techniques is mainly driven by their prevalence, associated risk and potential impact, as indicated by the *OWASP top 10*, the *CWE/SANS Top 25 most dangerous programming errors* and relevant academic work, as presented in important security-related journals and conference proceedings. The guide aims to achieve completeness for the set of threats in part II, and to achieve a good coverage on the variety of attack techniques in part III.

Each of the **20 attack techniques** in the Web security platform guide is documented according to the following structure:

Problem description The problem description explains in detail the problem setting, the goal of the attack, the necessary attacker capabilities and how the assets get attacked. If multiple variations of the attack technique exist, the differences are briefly discussed. For more complex attack scenarios, an attack tree is provided to guide the reader through the different steps of the attack.

Mitigation techniques The most common mitigation techniques for the attack are presented and discussed. In this section, the reader gets an understanding of how the mitigation technique works, and insights in their effectiveness (or ineffectiveness).

State-of-practice In the state-of-practice section, insights and statistics on the prevalence of the attack, or the level of deployment of mitigation techniques are presented. This provides researchers and industry players a good understanding of the current security state of the web ecosystem: Are available best practices being deployed? How widespread are known vulnerabilities? How often are these vulnerabilities being attacked?
Unfortunately, this highly-relevant material is not always available, or difficult to acquire. This platform guide aimed to include the publicly available statistics, which was possible for about half of the attack techniques presented.

Research and standardization activities The security guide summarizes the most important recent and ongoing research and standardization activities concerning this type of attack. This collection of key reference material captures most relevant evolutions and trends in academia (mainly geared towards an academic audience), and ongoing activities and new initiatives in standardization.

Best practices The guide provides a set of best practices to tackle this attack, now and in the near future. This catalogue of best practices should guide industry actors to gradually improve the security of their web-enabled services.

PART IV: Conclusion

In the conclusion of the Web platform security guide, we link together the assets from part II and the corresponding attacks from part III. This overview of the Web security threat landscape is depicted in Table II, and can be interpreted in two ways:

- On the one hand, Table II illustrates the impact of a

	Server Machine	Client Machine	Server-side Content Storage	Client-side Content Storage	Content in Transit	Client-side Application Code	Authenticated Session	Application Transactions	Authentication Credentials	Personal Information
Session Hijacking							⋆			
Session Fixation							⋆			
Brute Force									⋆	
Stealing Authentication Credentials									⋆	
Cross-site Request Forgery								⋆		
Login Cross-site Request Forgery								⋆		
Clickjacking								⋆		
Eavesdrop on Network Traffic				⋆	⋆	⋆	⋆	⋆	⋆	⋆
SSL Stripping				⋆	⋆	⋆	⋆	⋆	⋆	⋆
Man-in-the-Middle Attack				⋆	⋆	⋆	⋆	⋆	⋆	⋆
Internal Attacks on TLS				⋆	⋆	⋆	⋆	⋆	⋆	⋆
Cross-site Scripting				⋆		⋆	⋆	⋆	⋆	⋆
Compromising JavaScript Inclusions				⋆		⋆	⋆	⋆	⋆	⋆
Malicious Browser Extensions		⋆		⋆		⋆	⋆	⋆	⋆	⋆
Drive-By Download		⋆		⋆		⋆	⋆	⋆	⋆	⋆
Attacking Local Infrastructure		⋆		⋆		⋆	⋆	⋆	⋆	⋆
Injection Attacks	⋆		⋆	⋆		⋆	⋆	⋆	⋆	⋆
Break Access Control	⋆		⋆	⋆		⋆	⋆	⋆	⋆	⋆
User Tracking/Fingerprinting							⋆	⋆	⋆	⋆
History Sniffing							⋆	⋆	⋆	⋆

TABLE II: Overview of the Web security threat landscape: mapping assets from part II to attacks from part III.

particular Web attack on the various assets of the Web platform. For instance, a successful *injection attack* can potentially impact 8 of 10 assets of the Web platform.

- On the other hand, Table II enumerates the list of attacks that need to be mitigated in a particular Web application in order to protect an asset. For instance, in order to fully protect *application transactions*, at least 17 attacks need to be mitigated.

The overview of the Web security threat landscape clearly illustrates the complexity of the Web ecosystem. To improve the end-to-end security, it is necessary to raise the bar on several (if not all) topics in parallel.

Finally, this document expresses the insights of the STREWS consortium on the way forward for web application security. We point out a set of interesting challenges for securing the Web platform, opportunities for future research and trends in improving Web security. Some important examples include:

Limited adoption of the best practices. There exists a remarkable mismatch between state-of-the-art mitigation techniques and best practices being available for almost all vulnerabilities, and we measured only a limited adoption of the best practices in the state-of-practice.

The question remains as to how web site owners can be incentivized to actually deploy best practices on their sites? Similarly, to track the adoption rate over time, it is important to have good metrics and measurements in place to be able to assess the state-of-practice of the Web ecosystem.

Trend towards server-driven browser enforcement. Significant areas of novel web security technology (both in research and standardization) follow the same pattern: The server issues a security policy, the policy is pushed towards the client as part of the web application, and the client is responsible for enforcing the policy correctly. Well-known examples in recent specifications are CSP, X-Frame-Options, HSTS, and Certificate Pinning.

In this context, the Content Security Policy (CSP) seems to be a very promising additional layer of defense, protecting against cross-site scripting and UI redressing.

Legacy building block as weak links. We clearly see the urge to fix some of the legacy building blocks of the Web model. For instance, passwords are still the primary authentication technique on the Web, and are almost always used in combination with bearer tokens (e.g., session management cookies and OAuth tokens).

Major changes to legacy building blocks of the Web model face prohibitive deployment obstacles, as the currently-deployed legacy of web applications relies on the legacy model's properties, and the adoption of best practices is rather slow.

Increasing need to compartmentalize web applications. As web applications are becoming larger, and contain more third-party components (e.g., third-party JavaScript inclusions), the secure containment or sandboxing of untrusted parts of the web application becomes crucial. Current state-of-the-art containment techniques still need to mature, both in terms of policy specification as well as enforcement techniques.

Shift from purely technical to user-centered. Web security is partially shifting from a purely technical topic to a user-centered topic. This is illustrated with the numerous phishing and social engineering attacks, and the web permission model relying more and more on decisions of the end-user. For sure, attackers will more and more target the user as the weakest link of the web infrastructure.

Moreover, UI security becomes a key factor in delivering a secure web ecosystem, especially with the rise of new web-capable devices, such as smartphones, tablets, etc. The web permissions model is extraordinarily complex, and hard to understand for the end-user. Key questions are how to involve (or not involve) users in security-related web decisions and how to communicate back security results to the user.

These topics have been identified during the assessment, and a representative selection will be tackled in more detail in the upcoming case study and the roadmapping activities of the STREWS project.

ACKNOWLEDGEMENTS

The research of this Web-platform security guide is research project STREWS (Strategic Research Roadmap for European Web Security), which is funded by the European Union's 7th Framework Programme.

The Web-platform security guide is freely available at http://www.strews.eu/results/5-web-platform-security-guide.

The security guide incorporates contributions from Bert Bos, Elwyn Davies, Philippe De Ryck, Lieven Desmet, Stephen Farrell, Martin Johns, Wouter Joosen, Sebastian Lekies, Jan Tobias Mühlberg, Frank Piessens, Thomas Roessler, and Steven Van Acker.