Report from Dagstuhl Seminar 18321

Web Application Security

Edited by

Martin Johns[1], Nick Nikiforakis[2], Melanie Volkamer[3], and John Wilander[4]

1 **TU Braunschweig, DE,** mj@martinjohns.com
2 **Stony Brook University, US,** nick@cs.stonybrook.edu
3 **KIT – Karlsruher Institut für Technologie, DE,** melanie.volkamer@kit.edu
4 **Apple Computer Inc. – Cupertino, US,** john@wilander.net

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 18321 "Web Application Security". In this third seminar on the topic, a healthy mix of academics, practitioners and representatives of all major browser vendors reflected on the last decade of web security research and discussed the upcoming security challenges for the Web platform. In addition, for the first time, the list of attendees included several members of the *human factors in security* community, to enable broadening the web security topic towards this important facet of application security.

Seminar August 5–8, 2018 – http://www.dagstuhl.de/18321

2012 ACM Subject Classification Security and privacy → Browser security, Security and privacy → Information flow controlSecurity and privacy → Web protocol security, Security and privacy → Software security engineeringSecurity and privacy → Web application security, Security and privacy → Privacy protections, Security and privacy → Usability in security and privacy

Keywords and phrases Web Application Security, Browser Security, Software Security, Human Aspects in Security

Digital Object Identifier 10.4230/DagRep.8.8.1

1 Executive Summary

Martin Johns
Nick Nikiforakis
Melanie Volkamer
John Wilander

Introduction

Motivation

Since its birth in 1990, the Web has evolved from a simple, stateless delivery mechanism for static hypertext documents to a fully-fledged run-time environment for distributed, multi-party applications. Even today, there is still a continuous demand for new features and capabilities which drives the Web's evolution onwards. This unplanned and often chaotic development has led to several deeply ingrained security and privacy problems that plague the platform:

- The Web's original hypertext, multi-origin nature which is manifested in the design of HTML and HTTP is in fundamental conflict with JavaScript's Same-Origin Policy, the Web's most important security mechanism.
- Important security properties, such as end-to-end communication security or endpoint identity are outside of the control of the actual applications. Instead, they depend on the security of external entities, such as domain name servers or certificate authorities.
- Data/code separation in web applications is practically infeasible, as the HTTP link between server-side application logic and client-side application interface requires an intermixing of protocol, data and code fragments within a single continuous character stream.
- HTTP is a stateless protocol without a native session or authentication tracking concept.
- Users are not aware of general or application specific threats. Protecting against these threats (incl. to know which security indicators to trust) is nowadays difficult and time consuming.

Using this fragile basis, critical applications are created, that long have left the strict client-server paradigm, on which the Web was initially built. Instead, scenarios are realized that involve several mutually distrusting entities in a single security and application context. In many cases the browser is the link that connects the remote parties, either via direct JavaScript inclusion, web mashups, or through the usage of web protocols, such as OpenID and OAuth.

The accumulated ballast of the last two decades of web evolution, the ever growing functional demands of sophisticated web applications and the ambitious vision of the web platform's drivers creates an exciting tension field which is in constant conflict with the required security assurances of high value business applications.

Since approximately ten years, academic security and privacy research has recognized the importance of the web platform and the unique characteristics and challenges of the web security and privacy topic. And while specific techniques, that originated from academic research, such as the Content Security Policy, have been adapted in practice, the fundamental security problems of the web remain and the overall vulnerability landscape is getting worse, as it can be seen in the constant flow of reported web security issues in bug trackers and vulnerability databases.

Academic web security research has started 2007 and usable security research started almost at the same time. In the context of this Dagstuhl Seminar, we will revisit the lessons learned from the last decade and revisit the success stories and mistakes that have been made. Questions, that have to be raised in include "What has worked?", "What has been taken up by industry?", "What failed and why?", and – most importantly – "What did we learn?"

Seminar Objectives

Today, several unconnected groups drive the topic, including Security, Privacy as well as Usable Security & Privacy Academics, standardization, and browser vendors. The seminar will facilitate essential exchange between them. This will allow academia to directly influence browser vendors and standardization representatives, and allow industry representatives to influence the research community.

Overview

Participants

The seminar was well attended with 39 participants. A good balance of European and American researchers was present. Furthermore, the group represented a nice mix of participants of academia and industry. Compared to the previous editions, not only researchers from the web security area participated but also from the field of human factors in security.

Structure

This was the third Dagstuhl seminar on Web application security. The seminar's organisation combined overview presentation of various subfields, highlight talks, and discussions in working groups. In particular the overview presentations were important to connect the two research fields web security from a more technical point of view and human factors in security. This way, also a good, comprehensive view on current activities and open problems in the realm of Web application security in particular from a user's point of view could be achieved and areas for potential future collaborations could be identified.

Summary

Talks

The following people presented either an overview of their research field, very recent research results or overarching observations on the field of web application security. Please also refer to Section 3 for selected talk abstracts.

- Stefano Calzavara, University of Venezia, IT: REASON – A programmable architecture for secure browsing
- Luca Compagna, SAP Labs France – Mougins, FR: Analysis & Detection of Authentication Cross-Site Request Forgeries
- Lieven Desmet, KU Leuven, BE: Detecting and Preventing Malicious Domain Registrations in the .eu TLD
- Steven Englehardt, Mozilla – Mountain View, US: No Boundaries: Data exfiltration by directly embedded tracking scripts
- Thomas Gross, Newcastle University, GB: Investigating Cognitive and Affective Predictors Impacting Password Choice
- Mario Heiderich, Cure53 – Berlin, DE, DOMPurify: Client-Side Protection Against XSS and Markup Injection
- Boris Hemkemeier, Commerzbank AG – Frankfurt, DE: Web application security in vulnerable environments
- Martin Johns, TU Braunschweig, DE: WebAppSec @ Dagstuhl – The Third Iteration
- Christoph Kerschbaumer, Mozilla – San Francisco, US: Could we use Information Flow Tracking to generate more sophisticated blacklists?
- Pierre Laperdrix, Stony Brook University, US: Browser fingerprinting: current state and possible future
- Sebastian Lekies, Google Switzerland – Zürich, CH: Trusted Types: Prevent XSS with this one simple trick!
- Benjamin Livshits, Imperial College London, GB: Browser Extensions for the Web of Value

- Marius Musch, TU Braunschweig, DE: On measurement studies and reproducibility
- Lukasz Olejnik, Independent researcher, W3C TAG, FR: Private browsing modes guaranteed. On the example of Payment Request API
- Juan David Parra, Universität Passau, DE: Computational Resource Abuse through the Browser
- Giancarlo Pellegrino, Stanford University, US: Removing Browsers from the Equation: A New Direction for Web Application Security
- Tamara Rezk, INRIA Sophia Antipolis, FR: Content Security Policy Challenges
- Konrad Rieck, TU Braunschweig, DE: Beyond the Hype: Web Security and Machine Learning?
- Andrei Sabelfeld, Chalmers University of Technology – Göteborg, SE: A Challenge for Web of Things: Securing IoT Apps
- Sebastian Schinzel, FH Münster, DE: Handling HTML Emails after the Efail Attacks
- Zubair Shafiq, University of Iowa – Iowa City, US: The Arms Race between Ad Tech vs. Adblockers: Key Challenges and Opportunities
- Lynsay Shepherd, Abertay University – Dundee, GB: How to Design Browser Security and Privacy Alerts
- Dolière Francis Somé, INRIA Sophia Antipolis, FR: The Same Origin Policy and Browser Extensions
- Ben Stock, CISPA – Saarbrücken, DE: Persistent Client-Side Cross-Site Scripting in the Wild
- Melanie Volkamer, KIT – Karlsruher Institut für Technologie, DE: Web Security Meets Human Factors in Security
- Mike West, Google – München, DE: HTTP State Tokens

Conclusions

This seminar was the third Dagstuhl Seminar von Web Application Security, following Seminar 09141 (2009) and Seminar 12401 (2012). Thus, it was a great opportunity to reflect on a decade of web security research. In 2009 the field was largely undefined and that year's seminar offered a wild mix of various topics, some with lasting impact and many that went nowhere. Where the 2009 seminar was overly broad, the 2012 iteration had a comparatively narrow focus as the seminar was dominated by the notion that solving web security mainly revolves around solving the security properties of JavaScript.

This year's seminar reflected the ongoing maturing of the topic very well. Fundamental problems, such as Cross-site Scripting or the Web Browser security model, are well explored and their understanding served as a great foundation for the seminar's discussions. This allowed the extension of the topic toward important facets, such as privacy problems or human factors. While the addressed topics were too broad and the time for overarching discussions was limited due to the three-day format of the seminar, the sparked discussions were fruitful for several follow-up activities (see above). An underlying theme of the seminar can be summarized as "the last decade of web security has broad good progress and development but the overall problem is still neither fully understood nor solved". Especially, the newly introduced dimension of integrating human factors in security, which was reflected through including several high-profile members of this community in the seminar, is still immature.

One of the seminar's prime objectives has been reached very nicely: The fostering of collaboration between the different web security communities. For one, several compelling

interactions between practitioners from industry (such as SAP, Commerzbank and Cure53) and researcher from academia took place. Furthermore, thanks to the fact that all major web browser vendors (plus the new privacy-centric browser Brave) were represented at the seminar, both cross-browser vendor interaction as well as browser/academia collaborations were initiated, with the browser-based sanitizer initiative (see breakout session 4.3) being a prominent example.

2 Table of Contents

3 Overview of Talks

3.1 No Boundaries: Measuring data exfiltration by third-party scripts

Steven Englehardt (Mozilla – Mountain View, US)

Web tracking is pervasive. A core requirement of web tracking – the identification of individuals across website – is increasingly difficult as browser vendors adopt strict cookie policies and users take steps to protect their privacy. As a result, web trackers have deployed invasive tracking techniques that lack user (and sometimes browser) controls.

In this talk I'll explore findings from our recent web tracking measurements, which show the lengths to which trackers have gone to collect user information. Examples include: the abuse of browser autofill to collect email addresses, the exfiltration of information from social login APIs, and the collection of user information from the DOM. We find that some websites which embed these trackers are – much like users – completely unaware of these practices. I'll close with a discussion of our options for preventing this type of tracking.

3.2 Could we use an Information Flow Tracking to generate more sophisticated blacklists?

Christoph Kerschbaumer (Mozilla – San Francisco, US)

JavaScript (JS) has become the dominant programming language of the Internet and powers virtually every web page. User agents face a difficult situation: on the one hand JavaScript allows websites to provide a rich user experience; on the other hand JavaScript allows adversaries to perform malicious actions. To distinguish between good and malicious JavaScript at runtime has proven complicated and quite often browser vendors see no other options than relying on pre-rendered blacklists to block malicious JavaScript from executing.

While the approach of building an Information Flow Tracking system into a web browser has proven questionable: (a) because of the performance drawback, and (b) because of various loopholes which do not allow precise information tracking in a browser mostly due to JavaScripts dynamic nature. Nevertheless, an enhanced browser performing information flow tracking might still be able to detect malicious actions of JavaScript and hence provide input for creating more sophisticated blacklists.

Hence we ask: Could we use an Information Flow Tracking to generate more sophisticated blacklists?

3.3 Browser fingerprinting: current state and possible future

Pierre Laperdrix (Stony Brook University, US)

Joint work of Vastel, Antoine; Rudametkin, Walter; Rouvoy, Romain; Gómez-Boix Alejandro; Baudry, Benoit
Main reference Alejandro Gómez-Boix, Pierre Laperdrix, Benoit Baudry: "Hiding in the Crowd: an Analysis of the Effectiveness of Browser Fingerprinting at Large Scale", in Proc. of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018, pp. 309–318, ACM, 2018.
URL https://doi.org/10.1145/3178876.3186097

After a brief introduction on what browser fingerprinting is, we will take a look at the latest studies published in the domain* and the current ecosystem regarding fingerprinting protection. Then, we will see what lies ahead by talking about how this technique could be used positively to increase online security.

Open questions: Is there a future for constructive fingerprinting? If so, how?

*3 papers:

- FP-Scanner: The Privacy Implications of Browser Fingerprint Inconsistencies Antoine Vastel, Pierre Laperdrix, Walter Rudametkin, Romain Rouvoy (USENIX Sec. 2018)
- FP-STALKER: Tracking Browser Fingerprint Evolutions Antoine Vastel, Pierre Laperdrix, Walter Rudametkin, Romain Rouvoy (S&P 2018)
- Hiding in the Crowd: an Analysis of the Effectiveness of Browser Fingerprinting at Large Scale Alejandro Gómez-Boix, Pierre Laperdrix, Benoit Baudry (WWW 2018)

3.4 Security of Modern Mobile Browsers

Nick Nikiforakis (Stony Brook University, US)

Joint work of Meng Luo, Oleksii Starov, Nima Honarmand, Nick Nikiforakis
Main reference Meng Luo, Oleksii Starov, Nima Honarmand, Nick Nikiforakis: "Hindsight: Understanding the Evolution of UI Vulnerabilities in Mobile Browsers", in Proc. of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017, pp. 149–162, ACM, 2017.
URL https://doi.org/10.1145/3133956.3133987

Much of recent research on mobile security has focused on malicious applications. Although mobile devices have powerful browsers that are commonly used by users and are vulnerable to at least as many attacks as their desktop counterparts, mobile web security has not received the attention that it deserves from the community. In particular, there is no longitudinal study that investigates the evolution of mobile browser vulnerabilities over the diverse set of browsers that are available out there. In this paper, we undertake the first such study, focusing on UI vulnerabilities among mobile browsers. We investigate and quantify vulnerabilities to 27 UI-related attacks–compiled from previous work and augmented with new variations of our own–across 128 browser families and 2,324 individual browser versions spanning a period of more than 5 years. In the process, we collect an extensive dataset of browser versions, old and new, from multiple sources. We also design and implement a browser-agnostic testing framework, called Hindsight, to automatically expose browsers to attacks and evaluate their vulnerabilities. We use Hindsight to conduct the tens of thousands of individual attacks that were needed for this study. We discover that 98.6% of the tested browsers are vulnerable to at least one of our attacks and that the average mobile web browser is becoming less secure with each passing year. Overall, our findings support the conclusion that mobile web security has been ignored by the community and must receive more attention.

3.5 Beyond the Hype: Web Security and Machine Learning?

Konrad Rieck (TU Braunschweig, DE)

Machine learning has made considerable progress in the last years. Unfortunately, this progress is overshadowed by a hype in the industry, and it has become difficult to separate good ideas from marketing phrases. While this talk cannot solve this problem, it aims at highlighting three recent learning concepts that might be fruitful in the context of Web security and deserve to be discussed, irrespective of the current hype.

3.6 A Challenge for Web of Things: Securing IoT Apps

Andrei Sabelfeld (Chalmers University of Technology – Göteborg, SE)

Joint work of Iulia Bastys, Musard Balliu, Andrei Sabelfeld
Main reference Iulia Bastys, Musard Balliu, Andrei Sabelfeld: "If This Then What?: Controlling Flows in IoT Apps", in Proc. of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS 2018, Toronto, ON, Canada, October 15-19, 2018, pp. 1102–1119, ACM, 2018.
URL https://doi.org/10.1145/3243734.3243841

IoT apps empower users by connecting a variety of otherwise unconnected services. Unfortunately, the power of IoT apps can be abused by malicious makers, unnoticeably to users. We demonstrate that popular web-based IoT app platforms are susceptible to several classes of attacks that violate user privacy, integrity, and availability. We estimate the impact of these attacks by an empirical study. We suggest short/medium-term countermeasures based on fine-grained access control and long-term countermeasures based on information flow tracking. Finally, we discuss general trends and challenges for securing the Web of Things.

3.7 Efail: Breaking S/MIME and OpenPGP Email Encryption using Exfiltration Channels

Sebastian Schinzel (FH Münster, DE)

Joint work of Damian Poddebniak, Christian Dresen, Jens Müller, Fabian Ising, Sebastian Schinzel, Simon Friedberger, Juraj Somorovsky, Jörg Schwenk
Main reference Damian Poddebniak, Christian Dresen, Jens Müller, Fabian Ising, Sebastian Schinzel, Simon Friedberger, Juraj Somorovsky, Jörg Schwenk: "Efail: Breaking S/MIME and OpenPGP Email Encryption using Exfiltration Channels", in Proc. of the 27th USENIX Security Symposium, USENIX Security 2018, Baltimore, MD, USA, August 15-17, 2018., pp. 549–566, USENIX Association, 2018.
URL https://www.usenix.org/conference/usenixsecurity18/presentation/poddebniak

The Efail attack abuses malleable encryption in the respective modes of encryption in the OpenPGP and S/MIME standards. The attacker changes an existing ciphertext in a way that its plaintext is exfiltrated to the attacker when opened. For encrypted emails, the attacker edges the actual content of the email in HTML tags that perform external HTTP requests (backchannels). The victim's email client decrypts the email and sends the plaintext to the attacker. Outdated cryptography clearly is the culprit here and deploying authenticated

encryption (AE) ciphers to the standards could prevent this attack in the future. Besides this cryptographic weakness, HTML emails also play an important role.

While it is possible to port Efail-like attacks to any data standard supporting backchannels, HTML makes the attack particularly easy. HTML emails and especially remote content loading (e.g. images, style sheets) can be used for user tracking and were known to be a privacy issue for many years. While it is quite common for privacy advocates to disable HTML in emails completely, most non-technical users insist on HTML emails because they value rich typesetting in their day-to-day work. This raises some questions:

Is HTML the way to go for future typesetting of emails? Are there safer alternatives?

What is a safe subset of the HTML standards that allows rich typesetting, but without allowing user-tracking or Efail-like attacks?

How to enforce this safe subset in existing emails clients?

3.8 How to Design Browser Security and Privacy Alerts

Lynsay Shepherd (Abertay University – Dundee, GB) and Karen Renaud (University of Abertay – Dundee, GB)

Main reference Lynsay A. Shepherd, Karen Renaud: "How to design browser security and privacy alerts", CoRR, Vol. abs/1806.05426, 2018.
URL http://arxiv.org/abs/1806.05426

Browser security and privacy alerts must be designed to ensure they are of value to the end-user, and communicate risks efficiently. We performed a systematic literature review, producing a list of guidelines from the research. Papers were analysed quantitatively and qualitatively to formulate a comprehensive set of guidelines. Our findings seek to provide developers and designers with guidance as to how to construct security and privacy alerts. We conclude by providing an alert template, highlighting its adherence to the derived guidelines.

3.9 REASON – A programmable architecture for secure browsing

Stefano Calzavara

Joint work of Stefano Calzavara, Riccardo Focardi, Niklas Grimm, Matteo Maffei
Main reference Stefano Calzavara, Riccardo Focardi, Niklas Grimm, Matteo Maffei: "Micro-policies for Web Session Security", in Proc. of the IEEE 29th Computer Security Foundations Symposium, CSF 2016, Lisbon, Portugal, June 27 - July 1, 2016, pp. 179–193, IEEE Computer Society, 2016.
URL https://doi.org/10.1109/CSF.2016.20

The REASON project is a research proposal which I wrote with the goal of improving the security architecture of web browsers. More specifically, REASON aims at replacing the traditional Same Origin Policy (SOP) of web browsers with a programmable security monitor amenable for formal verification.

Preliminary evidence of the effectiveness of the proposal was given in a paper at CSF'16, where a small fragment of the architecture was designed and implemented. This talk will discuss the main motivations behind REASON, its benefits and a few ideas on how to implement it on top of existing browsers.

3.10 Persistent Client-Side Cross-Site Scripting in the Wild

Ben Stock (CISPA – Saarbrücken, DE)

Joint work of Ben Stock, Marius Steffens

The Web has become highly interactive and an important driver for modern life, enabling information retrieval, social exchange, and online shopping. From the security perspective, Cross-Site Scripting (XSS) is one of the most nefarious attacks against Web clients. XSS was long since believed to fall into three categories: reflected, persistent, or DOM-based XSS. In this paper, we present the first systematic study of the threat of Persistent Client-Side XSS, which lies in the intersection of persistent and DOM-based XSS. While the existence of this class has been acknowledged, especially by the non-academic community like OWASP, prior works have either only found such flaws as side effects of other analyses or focussed on a limited set of applications to analyze. Therefore, the community lacks in-depth knowledge about the actual prevalence of Persistent Client-Side XSS.

To close this research gap, we leverage taint tracking to identify suspicious flows from client-side persistent storage (Web Storage, cookies) to dangerous sinks (HTML, JavaScript, etc.). We discuss two attacker models capable of injecting malicious payloads into these storages: one that can manipulate HTTP communication (e.g., in a public WiFi), another that abuses existing reflected Client-Side XSS vulnerabilities to persist their payload. With our tainting methodology and these models in mind, we study the prevalence of Persistent Client-Side XSS in the Alexa Top 5,000 domains. We find that more than 8% of them have unfiltered data flows from persistence to a dangerous sink, which showcases the developers' inherent trust in the integrity of storage content. Investigating those vulnerable flows allows us to categorize them into four disjoint categories and propose appropriate mitigations.

3.11 Human Factors in Web Application Security (and Privacy)

Melanie Volkamer (KIT – Karlsruher Institut für Technologie, DE)

Joint work of currenty and previous members of the SECUSO research group as well as Karen Renaud

The talk starts of by explaining main goals in the research area of human factors in security and privacy as well as the main ideas behind the human centered security / privacy by design methodology including the importance of identifying users' mental models and acknowledging that security / privacy is usually not the users primary task. Then selected research results in the area of web application security are presented: This includes just in time and place security interventions to support users in avoiding to provide sensitive information on http pages [1] and to support them in checking links in emails before actually clicking the link [2]. It also includes proposals how to design UIs for security and privacy settings [3]. The talk concludes by raising open research questions in this area.

References

1 Melanie Volkamer, Karen Renaud, Gamze Canova, Benjamin Reinheimer, Kristoffer Braun. *Design and Field Evaluation of PassSec: Raising and Sustaining Web Surfer Risk Awareness*. TRUST 2015: 104-122, Springer, 2015

2 Melanie Volkamer, Karen Renaud, Benjamin Reinheimer, Alexandra Kunz. *User experiences of TORPEDO: TOoltip-poweRed Phishing Email DetectiOn*. Computers and Security 71: 100-113, 2017

3 Oksana Kulyk, Peter Mayer, Oliver Käfer, Melanie Volkamer. *A Concept and Evaluation of Usable and Fine-Grained Privacy-Friendly Cookie Settings Interface*. IEEE TrustCom 2018

4 Break Out Sessions

We had two time slots to discuss in small groups and two to present and discuss the results of these break out sessions with the entire group. The following topics were discussed.

- Browser Fingerprinting: Friend or Foe?
- Policies and Capabilities
- Cookies are Bad (for Authentication)
- My Browser Needs a Sanitizer
- Browser Warning Fatigue
- Browser Extensions

In the following sections, we will briefly document the individual sessions' discussions and results.

Browser Fingerprinting: Friend or Foe?

This breakout session covered various topics under the umbrella of browser fingerprinting.

Destructive vs. Constructive Use. Traditionally, browser fingerprinting has been treated by researchers and privacy-aware users as an intrusive practice that should, ideally, be detected and stopped. Yet the act of detecting and recognizing the device of a user can be used for constructive purposes, i.e., detecting the takeover of an account by the fact that the current user's fingerprint does not match the fingerprint collect during previous visits.

The participants of this breakout session discussed the advantages and disadvantages of using browser fingerprinting as an intrusion detection technique. Some participants mentioned that there already exist companies that provide bot-detection services based on device fingerprinting by attempting to recognize a device fingerprint as belonging to a popular bot/attack tool. Others argued that this would be a losing strategy in the long run, since attackers could randomize the fingerprint of their bot so that it stops matching the previously recorded fingerprints. There was some level of disagreement in terms of how feasible this is in the long run, as defenders only need to find one feature to recognize the true nature of a bot.

During this discussion of constructive (improving security) vs. destructive (worsening privacy) fingerprinting, some participants mentioned that if fingerprinting is used constructively, perhaps it can be limited to first-party websites, i.e., the fingerprinting script should be collected and used by the website that a user visits, and not by third parties present on that website.

Diversity and linkage of fingerprints. Most past studies involving browser fingerprinting with the help of volunteers, have reported a high-level of fingerprinting uniqueness, i.e., different users exhibiting different fingerprints which can be used to tell them apart. One of the participants mentioned that in a recent paper published in WWW 2018, researchers used a popular website to deploy their fingerprinting code and were thus able to collect fingerprints from the "general public" rather than those of privacy-aware volunteers. This study discovered that only 30% of users was unique, which is significantly less than prior studies which reported 90% or more uniqueness. The participants concluded that we need more research to identify the main reasons behind this lack of uniqueness reported by this recent stud.

User control. Given the group's discussion of the constructive use of browser fingerprinting, some participants highlighted that current fingerprinting is done in a surreptitious manner which makes people further distrust it. That is, JavaScript programs collect user information and create fingerprints of the user's browsing environment without the knowledge or consent of users.

Some participants, proposed bringing fingerprinting "to the surface" by asking users whether they want to be fingerprinted (similar to current browser popups related to geolocation and web notifications). By making this choice explicit, these participants argued that users could learn to trust a certain number of websites with their fingerprints (by accepting the relevant dialogues) allowing browser vendors and researchers to defend against browser fingerprinting that is done surreptitiously and without user consent.

4.1 Policies and Capabilities

This breakout session covered security and privacy policies set for pages and contexts, and the capabilities of JavaScript in a specific context.

Policies. Content Security Policy (CSP) has an interesting scope in that it limits code injection but not markup injection. Should we expand on CSP or come up with a complimentary policy mechanism? Further, a threat model is missing from CSP. It is mostly about avoiding injection and doesn't address data exfiltration. Was this intentional? Exfiltration can happen in many ways that are not URL-based resource loads such as window.open() + postMessage() and window.name. CSP is also used as mixed content protection and to avoid third-party script inclusion by your own developers. Going the other direction, should we create a strict CSP that is a fragment of CSP for specifically fighting XSS? We could have similar fragments for controlling framing.

4.1.1 Capabilities

Today, all JavaScript in an execution context are created equal. The origin of them or whether they are inline or file-based doesn't affect their powers over content, state, and network traffic. Could we restriction JavaScript use of password fields, payment APIs, computational resources, fingerprinting vectors etc to only a trusted subset?

If we have frame separation (cross-origin or not) we could support a CPU policy per frame. We could invent a new restricted script tag, for instance for ad scripts. Responsible (or previously compromised) sites will use this for third parties. These scripts would then be restricted in the ways described above. Or could some JavaScript sandbox be what we want?

A problem here is so called script gadgets which are very prevalent. They allow for ROP-style malicious code injection by inserting specific elements into the DOM that trigger code paths in legitimate libraries/frameworks (with full powers). This can be leveraged to cross the restriction boundary. Iframes and the sandbox directive may be too restrictive today. It's scripting on or off.

With a new script element rather than attribute on current script elements would allow us to get out of the gadget mess since vulnerable libraries/frameworks will not have flaws for the new script element. An alternative would be to ship something like <script capabilities="ad"></script>, wait a year later, then require it or block based on a blacklist of ad tech origins.

4.2 Cookies are Bad (for Authentication)

Cookies are primary targets of security and privacy attacks such as cross-site scripting, rogue scripting, and speculative execution attacks such as Spectre. This breakout session aimed at looking at how cookies are used today and seeing if we can achieve the same functionality with something more secure.

The current state of cookie usage. Recent statistics of cookie usage in Google Chrome:

- HttpOnly cookies $\approx 9\%$
- Secure cookies $\approx 7\%$
- SameSite cookies $\approx 0.03\%$

This shows how low the adoption of security measures are for this important protocol feature. In addition, websites use up to 180 cookies per site and up to 4kb per cookie which hurts network performance significantly.

Cookie purposes today:

- Hold authentication state.
- User recall (know that a series of requests are from the same user agent).
- Ad (re)targeting.
- Ad/click attribution.
- On-device storage.
- User preference (UI choices or other web app settings).

Towards a better authentication mechanism. We would like to deprecate cookies for the purpose of authentication/user identification in browsers, not for HTTP in general. To get there, these two things were mentioned:

1. Drive down the use of plaintext cookies is good.
2. Drive down the JavaScript use of cookies is good.

The rest of the session focused on Mike West's proposal for a different protocol state mechanism: https://mikewest.github.io/http-state-tokens/ which was discussed in depth.

Migration to a new mechanism. If we were to move to such a mechanism, how would be deprecate cookies, at least for authentication purposes in web browsers?

1. Introduce it.
2. Encourage usage.
3. Now we've given developers an alternative and can start removing cookie support.

A final note on migration was that maybe the browser should not send a token on the first page load, but instead have the server to opt in. The browser could announce its support for the token mechanism.

4.3 My Browser Needs a Sanitizer

The session was joined by participants from Google, Microsoft, Mozilla, SAP and Cure53. The goal was to evaluate whether a browser should expose a HTML Sanitizer API or if this should rather be done by external JavaScript libraries.

The participants agreed that indeed the browser should indeed be the one to offer that feature for a variety of reasons. The discussion then focused on challenges and possible limitations and, as a result, the participants agreed on the next steps.

Those steps are as follows:

1. Creation of a proposal for WICG – essentially the authoring of an "explainer doc"
2. The initiation of authoring a specification draft and further discussions.

The "explainer doc" has by now been published, the spec is in preparation and is being authored by Mozilla and Cure53.

4.4 Browser Warning Fatigue

The group first worked on a common understanding of different types of browser "warnings" with different characteristics, e.g. (1) there are those which force you to make a decision (blocking) and those that appear more like a notice (you don't have to make a decision); (2) there are those that appear as icon (e.g., the lock icon; you may get more information when clicking on the icon) and those that contain text (e.g. explaining the situation why this warning now is shown and what the user needs to decide on); (3) there are those provided by the browser and those from the visited webpage; correspondingly also the positioning varies. For the webpage one the question whether tick boxes next to statements that one agrees on (privacy) policies should be considered as 'warning' was discussed.

With respect to the issues with various types of browser warnings, participants discussed the often mentioned habituation issue. The question was whether this issue is a consequence of badly designed warnings appearing too often without any consequence when ignoring them or whether habituation is an issue of any warning and better design will not help. It was agreed that it should be possible to not just decide this one situation but to tell the system that similar situations should be decided automatically the same way without being actively interrupted again in future. It was discussed whether it is possible to predict user's decisions on warning dialogues (in particular in the privacy context) and therefore make the decisions automatically (or at least provide an option for the user that these decisions can be make automatically).

It was furthermore agreed on that warnings in terms of asking the user to decide should only be displayed if users can make an informed decision based on the information provided in the warning.

There was also a discussion on evaluating warnings in particular wrt to whether they cause fatigue. The issue with fatigue is that you may only measure it after people having used a system with the to be evaluated warnings for some time; which means one need to go for a field study but making sure that the underlying system does not introduce any security issues for the participants.

4.5 Browser Extensions

The final break-out session was dedicated to the topic of browser extensions. In this context several orthogonal topics were touched upon.

Permission System. Similar to mobile apps, browsers utilize permission systems to mitigate potential security problems by malicious extensions. Unfortunately, due to the technical intrinsic of the web model, the current permission granularity is insufficient. For instance, in many cases, such as DOM or Network access, the technically available options are to coarse, being essential full or no access. Within the sessions, alternative approaches were discussed, including moving away from tying permissions to technical capabilities and instead moving to activities.

Extension Vetting. A joint cross-vendor approach toward unified vetting of extension was proposed, as – thanks to standards such as the web extension model – an increasing number of extensions are written that simultaneous target multiple browsers.

Protection Users against malicious extensions. Finally, the session addressed methods to support users (and sites) against malicious extensions. In this context, the notion of trust-classes for web sites was brought up. This would allow the disabling of extensions for security sensitive sites, such as banks, will enabling them on sites with lesser security requirements, such as entertainment sites.

4.6 Aftermath

The seminar was perceived as highly inspiring by the participants. In consequence, it had a fertilizing effect on follow-up activities: Besides various informal collaborations that resulted from discussions in Dagstuhl, we would like to single out results which directly can be attributed to the seminar:

- Upcoming paper on hybrid static/dynamic security analysis of web applications
- Various co-supervised students
- Several research visits (e.g., KIT/Abertay University)
- Several ongoing academic-industry collaborations (e.g., SAP/TU Braunschweig)
- Initiation of a cross-browser specification on a web browser-based API for security handling of untrusted data.

Participants

- Frederik Braun
Mozilla – Berlin, DE
- Achim D. Brucker
University of Sheffield, GB
- Stefano Calzavara
University of Venezia, IT
- Luca Compagna
SAP Labs France – Mougins, FR
- Lieven Desmet
KU Leuven, BE
- Steven Englehardt
Mozilla – Mountain View, US
- Thomas Gross
Newcastle University, GB
- Marian Harbach
Audi AG – Ingolstadt, DE
- Daniel Hausknecht
Chalmers University of Technology – Göteborg, SE
- John Hazen
Microsoft Corporation – Redmond, US
- Mario Heiderich
Cure53 – Berlin, DE
- Boris Hemkemeier
Commerzbank AG – Frankfurt, DE
- Martin Johns
TU Braunschweig, DE
- Christoph Kerschbaumer
Mozilla – San Francisco, US
- Pierre Laperdrix
Stony Brook University, US
- Sebastian Lekies
Google Switzerland – Zürich, CH
- Benjamin Livshits
Imperial College London, GB
- Matteo Maffei
TU Wien, AT
- Marius Musch
TU Braunschweig, DE
- Nick Nikiforakis
Stony Brook University, US
- Lukasz Olejnik
Independent researcher, W3C TAG, FR
- Juan David Parra
Universität Passau, DE
- Giancarlo Pellegrino
Stanford University, US
- Karen Renaud
University of Abertay – Dundee, GB
- Tamara Rezk
INRIA Sophia Antipolis, FR
- Konrad Rieck
TU Braunschweig, DE
- Andrei Sabelfeld
Chalmers University of Technology – Göteborg, SE
- Sebastian Schinzel
FH Münster, DE
- Zubair Shafiq
University of Iowa – Iowa City, US
- Lynsay Shepherd
Abertay University – Dundee, GB
- Dolière Francis Somé
INRIA Sophia Antipolis, FR
- Ben Stock
CISPA – Saarbrücken, DE
- Daniel Veditz
Mozilla – Mountain View, US
- Melanie Volkamer
KIT – Karlsruher Institut für Technologie, DE
- Malte Wedel
SAP SE – Walldorf, DE
- Rigo Wenning
W3C / ERCIM, FR
- Mike West
Google – München, DE
- John Wilander
Apple Computer Inc. – Cupertino, US
- Henrik Willert
1&1 Internet SE – Karlsruhe, DE