Dagstuhl Seminar 09141
Web Application Security
29.03. – 03.04.2009
Executive Summary

Dan Boneh[1], Ulfar Erlingsson[2], Martin Johns[3] and Benjamin Livshits[4]

[1] Stanford University, US
dabo@cs.stanford.edu
[2] Reykjavik University, IS
ulfar@ru.is
[3] Universitt Passau, DE
mj@sec.uni-passau.de
[4] Microsoft Research - Redmond, US
livshits@microsoft.com

Abstract. Web applications are ubiquitous nowadays. Consequently, the field of Web application security is of ever rising significance. This Dagstuhl seminar was conducted to assemble researchers active in the domain to gain a first comprehensive overview of this young discipline in security research. From a content perspective, the topic was explored in a great variety of directions, including for instance Web browser-based security measures, language-based techniques, software engineering centric methods, run-time enforcement, static analysis, or formal approaches.

Keywords. Web applications, Security, Ajax, Web 2.0, Analysis for security, Browser design, Distributed applications

1 Introduction

1.1 Motivation

Security of Web applications has become increasingly important over the last decade. This is not at all surprising: Web applications are now ubiquitous, spanning the spheres of e-commerce, healthcare, finance, and numerous other areas. More and more Web-based enterprise applications deal with sensitive financial and medical data, which, if compromised, in addition to downtime can mean millions of dollars in damages. It is crucial to protect these applications from malicious attacks. Yet, to date, a great deal of attention has been given to network-level attacks such as port scanning, even though, about 75% of all attacks against Web servers target Web-based applications, according to recent surveys. Traditional defense strategies such as firewalls do not protect against Web application attacks, as these attacks rely solely on HTTP traffic, which is

Dagstuhl Seminar Proceedings 09141
Web Application Security
http://drops.dagstuhl.de/opus/volltexte/2010/2725

usually allowed to pass through firewalls unhindered. Thus, attackers typically have a direct line to Web applications. Furthermore, traditional vulnerabilities such as buffer overruns, pervasive in applications written in C and C++, that have been the subject of intense for over a decade are now largely superseded by Web applications vulnerabilities such as cross-site scripting, SQL injection, and cross-site request forgery attacks.

Web applications have progressed a great deal in the last decade since their humble beginnings as CGI scripts. Todays Web applications are sophisticated multi-tier systems that are built on top of complex software stacks. Web applications are also distributed: a Web application typically includes both a server-side component running on top of an application server such as JBoss, as well as a client-side component that usually consists of HTML and JavaScript. Consequently, Web application security touches upon many aspects of systems research. The topic of Web application security has attracted researchers from diverse backgrounds in recent years. In addition to core security experts, this includes specialists in programming languages, operating systems, and hardware. Similarly, the research directions proposed so far range from improving security through Web browser changes to low-level hardware-level support and in-depth analysis of server code. Last but not least, much work remains to be done in social engineering for security as applied to Web applications.

The last several years have seen dramatic changes in Web application development. We are now in the middle of the Web 2.0 revolution, triggered by demand for better, more interactive user experience and enabled by Ajax (asynchronous JavaScript and XML). However, extra functionality of rich-client applications is generating new security concerns. A good example of that is JavaScript worms, which first emerged in 2005 and have grown increasingly popular in the last year or so. JavaScript worms take advantage of the ability of the Web client to programmatically issue server requests through Ajax to propagate malicious payload.

1.2 Seminar objectives

A sea-change is taking place in how Web applications deliver their functionality, which is starting to give end users greater access to richer, more interactive Web services. Underlying this change are three developments: a new generation of richer Web content, such as interactive video, the aggregation of Web functionality from many services, and, finally, the migration of Web application functionality to the client Web browsers, in the form of scripts and other executable content. However, as often happens, these changes are being driven by functionality, with consideration of security included mostly as an afterthought. The resulting objectives of this seminar were two-fold.

Objective 1: We wanted to discuss new ideas for making the Web a safer place, where end users can be given guarantees about security, integrity, and availability, as well as about their privacy. We were particularly focused on techniques for the construction of such robust and secure Web applications. Recently, there

have been numerous industry conferences and other venues for the discussion of Web application security issues between industry practitioners. The discussion of these issues from a more fundamental, Our intent with this seminar was to enable participants from academia and industry to discuss these Web application security issues from a principled, more formal perspective with longer-term goals in mind.

Objective 2: We wanted to foster a productive discussion on what the future holds in terms of Web application security, given current Internet trends. In particular, in addition to the migration towards substantial client-side script execution, there is a strong shift towards video and other rich content on the Web. What can be done today to ensure that Web applications developed to serve future content do not repeat the mistakes of the current Web? What are the important lessons we should apply from the first 10 years of Web application (in)security? It seems clear that more and more sophisticated applications will be deployed over the Web, as exemplified the Web-base Google office suite. Through this seminar, we hope that design proposals may be developed whose implementation now can lead to the security-by-design of Web applications in the coming decades.

2 Overview

2.1 Participants

The seminar was well attended with 38 participants. A good balance of European and American researchers was present. Furthermore, the group represented a nice mix of participants of academia and industry (including members of companies such as Mozilla, Microsoft, SAP, and Google).

2.2 Structure

This was the first Dagstuhl seminar on Web application security. In addition, academic research on this topic is a rather young discipline. For this reason, the seminar's organisation favored presentations over open workgroups or plenum style discussions. This way, a good, comprehensive view on current activities and open problems in the realm of Web application security could be achieved.

2.3 Documentation of content

Since the seminar took place, the underlying research of most talks has been presented at conferences and the corresponding papers have been published in the associated proceedings. Hence, we list a comprehensive list of publications that are directly associated with the seminar's content in the bibliography of this document.

3 Summary

Each day of the seminar was structured to explore one or more dedicated topics within the field of Web application security. The this section follows the structure of the seminar.

3.1 Monday: Attacks and defenses

The first day of the seminar was devoted to setting the mood for the rest of the seminar by reviewing various attacks and weaknesses of the current generations of web applications and browsers:

- Dan Boneh and Martin Johns [1] each gave overview talks discussing various attacks such as Cross-site Request Forgery, timing attacks, DNS rebinding, or browser-based intranet exploration.
- Engin Kirda presented his work on automated identiy theft in social networking site, such as Facebook [2].
- Giovanni Vigna gave an insightful and entertaining talk discussing lessons that he learned during his work with his Web security start-up company.
- Thorsten Holz, a member of the Honeynet project, gave an overview of Web-based malware that he encountered through his work.
- John Wilander presented the results of an informal survey on security awareness among developers that he conducted.
- Jochen Haller ended the day with an account of experiences that can be made while trying to create a secure, non-trivial web application in the ERP world.

3.2 Tuesday: Browsers, Blue Sky and Crypto

The first half of the second day was devoted to a central component in the Web paradigm – the Web browser. Various topics were discussed, including future directions in browser scripting (Eich), overcoming unpredictable rendering of user-provided content (Venkatakrishnan), or protecting the user against low-level vulnerabilities in the Web browser (Holz).

- The inventor of JavaScript Brendan Eich presented his view on the language along with his current approach towards future information flow tracking within JavaScript.
- V.N. Venkatakrishnan presented *Blueprint* [3] an XSS defense strategy designed to minimize trust placed on browsers for interpreting untrusted content. The approach's main method is based on a combination of server-side preprocessing of user-provided content and a client-side counterpart that reliably renders this content in a secure fashion, thus, effectively preventing attacks that are based on non-standard behaviour of certain browser versions.

- The path towards better isolation between the individual principals in Web browser was the topic of Charlie Reis' presentation. The focus of the presentation was on identifying challenges, posing questions, and outlining potential solutions (which in parts build the basis of [4]).
- Using current threats, such as Phishing or UI redressing (which is also known under the term *ClickJacking*) as a motivation, David Evans introduced ideas to systematically incorporate user intentions in security policies [5].
- Thorsten Holz's talk was devoted to the topic of protecting Web browsers against drive-by download attacks [6]. The core of the presented approach centers around a local analysis step which extracts and analysis all JavaScript upon receiving an HTTP response. Only webpages that pass this test and exclusively contain JavaScript that is apparently non-malicious are subsequently rendered by the browser.
- The session was concluded by Dieter Gollmann, who directed the attention to the important problem of handling identity and authenticity in a browser-driven application model. Based on the insight that the Dolev-Yao model does not apply to Web applications, he reasoned that the current practice of deriving fundamental security properties from sub-application-layer level data, such as DNS-names, is flawed. Instead, he argued, the Web needs fine-grained authentication of requests, responses, tags, and fields based on name space for principals with minimal reliance on third parties.

Later on, in a session titled *Blue Sky: What is Web application security? What should it be?* an open-ended discussion between all seminar participants on the definition and state of the field took place. The intent of the discussion was twofold: First, to see if any ideas could be surfaced if participants were encouraged to think from first principles, and explicitly asked not to consider legacy constraints. Second, to see if the participants could find consensus on one or more aspects of the overall problem space – or, equally interestingly, if no consensus could be reached on anything. Would it be possible to define the concept of a Web application, and would it perhaps even be possible to say with some certainty what a secure Web application might look like.

The net result of this session was that little consensus could be reached on anything, and none of the participants felt that they could clearly define what a Web application was, or how one might proceed to secure that amorphous abstraction. Some recurring points were raised however. In particular, people-centric aspects ranging from privacy to such topics as politics and porn, were a common set of traits that was seen as rather unique to Web applications – as opposed to previous software. Also related to the human end users, the need for better authentication, better identity management, and better user interfaces and trusted paths was seen as key. On a more technical note, the dynamic nature of the Web, and of Web software (being provisioned on use, and potentially specialized for each use) was seen as unique, and offering opportunities. Other than this, the participants expressed strong desires for pervasive use of more traditional software isolation and access control mechanisms, including program partitioning ala microkernel operating systems.

The day ended with a session on cryptography and related formal topics.

- Dan Boneh presented symmetric cryptography implemented in pure JavaScript for client-side execution within the Web browser [7].
- Following an analysis of generic weaknesses of today's single sign-on protocols (SAML, MS Cardsapce, Liberty Alliance etc.), Joerg Schwenk proposed three different methods for a better integration of TLS into these protocols [8,9].
- Staying in the realm of browser-based protocols, Sebastian Gajek spoke about utilizing the universally composable framework for the analysis of such security protocols [10].

3.3 Wednesday: Analysis

The focal topic of the Wednesday session were analysis-centric approaches. Within this session, the individual targets of said analysis varied widely from plain HTTP traffic (Robertson), over isolated application data payloads (Borders), to observed user interaction with the application (Krishnamurthi), thus, covering many facets of the heterogeneous Web world.

- Will Robertson presented research on intrusion detection for web applications based on machine learning methods [11]. A central concern in the proposed technique is addressing *concept drift* in Web applications, i.e., the handling of changes in the applications behaviour resulting from updated code. This topic is of high relevance in the field of Web application, as inovation and update circles are significantly higher for Web applications compared to the classic application paradigm.
- Based on his work in respect to uncovering information leaks in Web traffic [12], Kevin Borders presented a work in progress on confidentiality enforcement. His hypothesis is that through isolating application-data from protocol overhead, better control in respect to outgoing data can be guaranteed.
- Shriram Krishnamurthi presented work on control flow enforcement in Web applications [13]. The presented technique is built around the correlation between client-side events and resulting HTTP requests. Through an analysis of the behaviour of the application this correlation can be established and a control-flow graph (CFG) of the application can be built. Then this CFG is used to extract a model of expected client behavior as seen from the server and create an intrusion-prevention proxy.
- The session was closed by an inspiring and thought provoking talk by Trevor Jim titled "2020 Foresight: Web application security in 11 years". Trevor utilized observations made in other fields in respect to short lived "hot topics" in research and long lasting success stories. Based on his observations, he concluded that the next important advances in web application security has already been in the pipeline for several decades.

The afternoon was reserved for the traditional hike.

3.4 Thursday: Information flow / restricting JavaScript

Insecure information flow in Web applications is the cause for the overwhelming majority of security problems in Web applications. For this reason, the first half of the day was reserved for approaches that deal with this topic. The presented works take effect on the server-side of web application, either on run-time (Sabelfeld), on compile-time (Pistoia), or in a hybrid fashion (Swamy).

- The day started with a keynote by Andrei Sabelfeld who gave a overview on the foundations of information flow tracking and its application in the Web world.
- He was followed by Nikhil Swamy who presented his approach towards cross-tier label-based security enforcement [14].
- William Halfond showed how dynamic information flow tracking can be used to mark the path of trusted values through the application (*positive tainting*), opposed to the wide-spread technique to track untrusted data. Subsequently, he showed how this technique can be used to protect against injection attacks [15] and to improve testing of Web applications [16].
- Then, Cedric Fournet gave a talk on secure compilation of programs with crypto primitives in order to preserves information-flow properties [17].
- Finally, Marco Pistoia presented work on static analysis of Web application source code to detect insecure information flows which in turn could lead to code injection vulnerabilities [18].

The second half of the day shifted the focus towards the client-side and examined techniques to restrict execution of untrusted JavaScript:

- Úlfar Erlingsson opened the topic area by giving an overview of technologies used for safe extensions on the Web. Taking a historic point of view, the talk recalled the enthusiasm Java as the first widely adopted, practical technology for safely providing active content – not just static HTML – on web pages. The talk further outlined how, from Java to HTML5, the need for client safety (i.e., of making active content equally safe as static HTML) has been a key driver of Web technologies; in that context, the talk considered technologies such as OS sandboxing (from mid-90s Janus to modern Chrome) and language-based techniques (especially the 90s SFI to todays Native Client).
- Salvatore Guarnieripresented *Gatekeeper* [19], an approach to analyse and instrument third party JavaScript widgets. This way the widget's code can be matched against predefined security policies to detect potential policy violations which might result in security issues.
- Sergio Maffeis spoke about measures towards a more formal foundation in our understanding of JavaScript [20]. Subsequently, he showed how this fundamental understanding of JavaScript's semantics could be used to identify security problems in the JavaScript subset that was employed by Facebook for third party content.

- Phu Phung proposed a lightweight approach to deprive untrusted JavaScript of potential harmful abilities [21]. The approach is based on the highly dynamic nature of the JavaScript run-time and object model which allows the redefinition of central components and interfaces before these are exposed to potentially malicious code.
- Jasvir Nagra presented *Google Caja* [22], a program rewriting framework that translates untrusted JavaScript into a JavaScript variant which supports capabilities. The run-time actions of resulting scripts can be effectively governed by applying security policies to the introduced capabilities, thus, transferring the untrusted code into a trustworthy variant which can be safely included into a Web page.

3.5 Friday: Security by construction

The last day was devoted to solutions that aim to integrate security concerns into the development process or the operation of Web applications.

- Frank Piessens discussed ideas how to extend the concept of *security by contract (SxC)* towards web applications. SxC is technique which was originally conceived in the context of mobile phones. The core of the method relies on formally defined security policies to restrict the actions of applications running on the enduser's device. These policies are compiled into the application's code and, thus, tightly interwoven with the application's logic. In his talk Frank outlines the upcoming challenges that might occur when this concept is moved to the Web.
- He was followed by Ben Livshits, who presented *RIPLEY* [23], a system that uses replicated execution to automatically preserve the integrity of a distributed computation. RIPLEY replicates a copy of the client-side computation on the trusted server tier. Every client-side event is transferred to the replica of the client for execution. RIPLEY observes results of the computation, both as computed on the client-side and on the server side using the replica of the client-side code. Any discrepancy is flagged as a potential violation of computational integrity.
- Christoph Kern, a member of Google's security group, gave an account of a currently at Google utilized practical approach to reduce the danger of creating XSS conditions in the software engineering process. The presented method builds upon the *Google Web Toolkit (GWT)*, an open-source development framework for AJAX web applications. Using a (from a developer's point of view reasonable and acceptable) coding discipline in respect to the usage of GWT's primitives with light-weight static checking, the engineers are able to develop and maintain GWT applications that are free of XSS with a high degree of confidence.
- Martin Johns [24] presented an approach to fundamentally prevent string-based code injection vulnerabilities, such as XSS or SQL injection, through strictly separating data and code. For this purpose a new, dedicated datatype is introduced that is exclusively utilized to assemble embedded computer

syntax, such as HTML or SQL. The datatype is filled using a language preprocessor that allows the developer to include the embedded syntax in an almost unmodified fashion.
– Finally, Florian Kerschbaum [25] showed a lightweight method to operate Web applications in a fashion that aims to combat the majority of all cross-site attacks. In addition, he showed how the utilized Alloy-based model checking to validate his approach.

The seminar ended with a short wrap-up by the organizers and a final discussion.

4 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 two results which directly can be attributed to the seminar:

For one, during the seminar the observation was made, that Europe at that point in time did not offer a compelling venue for academic Web application research. For this reason, a set of present participants decided to pursue this issue. The result of this effort was the *OWASP AppSec Research* conference, which had its first iteration in June 2010 in Stockholm.

Furthermore, based on initial discussions during the seminar, a consortium formed for further collaboration in a larger research project. This resulted in a successful proposal for a EU FP7 project. Out of the five primary drivers of the proposal, four (in the form of the seminar participants from SAP, Chalmers, KU Leuven, and Uni Passau) had met at the seminar. The project is called *WebSand* and will start in October 2010 its three year run. It will target research questions in the field of Web application security in multi-party scenarios.

5 Conclusions

The dominant result of the seminar was that *the* field of Web application security research simply does not exist. Instead, the topic is approached from a highly heterogeneous set of directions, ranging from low-level vulnerability countermeasures, through ad-hoc run-time enforcement mechanisms, over security protocol analysis, to fully formalized typing approaches. Research in this field has to be agile and versatile as even the most fundamental building blocks of the young application paradigm are still evolving and constantly changing – sometimes for the better, sometimes for the worse from a security point of view. The fight for secure Web applications is still an uphill battle. We live in interesting times.

References

1. Johns, M.: On JavaScript Malware and Related Threats - Web Page Based Attacks Revisited. Journal in Computer Virology, Springer Paris **4** (2008) 161–178

2. Bilge, L., Strufe, T., Balzarotti, D., Kirda, E.: All your contacts are belong to us: Automated identity theft attacks on social networks. In: 18th International World Wide Web Conference (WWW 2009). (2009)
3. Louw, M.T., Venkatakrishnan, V.: BluePrint: Robust Prevention of Cross-site Scripting Attacks for Existing Browsers. In: IEEE Symposium on Security and Privacy (Oakland'09). (2009)
4. Reis, C., Gribble, S.D.: Isolating web programs in modern browser architectures. In: Proceedings of the 4th ACM European conference on Computer systems (EuroSys '09), New York, NY, USA, ACM (2009) 219–232
5. Shirley, J., Evans, D.: The user is not the enemy: Fighting malware by tracking user intentions. In: New Security Paradigms Workshop (NSPW 2008). (2008)
6. Dewald, A., Holz, T., Freiling, F.C.: ADSandbox: Sandboxing JavaScript to fight Malicious Websites. In: Proceeding of the 25th Symposium On Applied Computing (SAC), Track on Information Security Research and Applications, ACM (2010)
7. Stark, E., Hamburg, M., Boneh, D.: Symmetric cryptography in javascript. In: ACSAC '09: Proceedings of the 2009 Annual Computer Security Applications Conference, Washington, DC, USA, IEEE Computer Society (2009) 373–381
8. Gajek, S., Manulis, M., Schwenk, J.: Enforcing user-aware browser-based mutual authentication with strong locked same origin policy. In: ACISP'08. (2008)
9. Gajek, S., Manulis, M., Sadeghi, A.R., Schwenk, J.: Provably Secure Browser-Based User-Aware Mutual Authentication over TLS. In: ASIACCS'08. (2008)
10. Gajek, S.: A universally composable framework for the analysis of browser-based protocols. In: Proceedings of ProvSec'08. Volume 5324 of LNCS. (2008) 313–328
11. Maggi, F., Robertson, W., Kruegel, C., Vigna., G.: Protecting a Moving Target: Addressing Web Application Concept Drift. In: Proceedings of the International Symposium on Recent Advances in Intrusion Detection (RAID'09). (2009)
12. Borders, K., Prakash, A.: Quantifying information leaks in outbound web traffic. In: Proceedings of the IEEE Symposium on Security and Privacy. (2009)
13. Guha, A., Krishnamurthi, S., Jim, T.: Using static analysis for Ajax intrusion detection. In: Proceedings of the 18th international conference on World wide web (WWW'09), New York, NY, USA, ACM (2009) 561–570
14. Swamy, N., Corcoran, B., Hicks, M.: Fable: A Language for Enforcing User-defined Security Policies. In: Proceedings of the IEEE Symposium on Security and Privacy (Oakland). (2008) 369–383
15. Halfond, W.G., Orso, A., Manolios, P.: Using Positive Tainting and Syntax-Aware Evaluation to Counter SQL Injection Attacks. In: 14th ACM Symposium on the Foundations of Software Engineering (FSE). (2006)
16. Halfond, W., Choudhary, S.R., Orso, A.: Penetration Testing with Improved Input Vector Identification. In: Proceedings of the IEEE International Conference on Software Testing (ICST 2009). (2009)
17. Fournet, C., Le Guernic, G., Rezk, T.: A Security-Preserving Compiler for Distributed Programs: From Information-Flow Policies to Cryptographic Mechanisms. In: ACM conference on Computer and Communications Security, New York, NY, USA, ACM (2009) 432–441
18. Geay, E., Pistoia, M., Tateishi, T., Ryder, B., Dolby, J.: Modular String-Sensitive Permission Analysis with Demand-Driven Precision. In: Proceedings of the 31st International Conference on Software Engineering (ICSE 2009). (2009)
19. Guarnieri, S., Livshits, B.: Gatekeeper: Mostly Static Enforcement of Security and Reliability Policies for JavaScript Code. In: in Proceedings of the Usenix Security Symposium. (2009)

20. Maffeis, S., Mitchell, J., Taly, A.: Run-time enforcement of secure javascript subsets. In: Proc of W2SP'09, IEEE (2009)
21. Phung, P.H., Sands, D., Chudnov, A.: Lightweight self-protecting javascript. In: Proceedings of ACM Symposium on Information, Computer and Communications Security (ASIACCS 2009). (2009)
22. Miller, M.S., Samuel, M., Laurie, B., Awad, I., Stay, M.: Caja - Safe active content in sanitized JavaScript. Whitepaper, http://google-caja.googlecode.com/files/caja-spec-2008-01-15.pdf (2008)
23. Vikram, K., Prateek, A., Livshits, B.: Ripley: Automatically securing distributed Web applications through replicated execution. In: Conference on Computer and Communications Security. (2009)
24. Johns, M., Beyerlein, C., Giesecke, R., Posegga, J.: Secure code generation for web applications. In: 2nd International Symposium on Engineering Secure Software and Systems (ESSoS '10). Volume 5965 of LNCS., Springer (2010) 96 – 113
25. Kerschbaum, F.: Simple Cross-Site Attack Prevention. In: Proceedings of the 3rd International Conference on Security and Privacy in Communication Networks (SecureComm'07). (2007)