CSCM08: Information Security Management Coursework 2: Report

Client-side Web Application security: An analysis of threats, successful attacks, and prevention techniques.

Luke Hengstenberg

878876

Table of Contents

Table of Figures

1: Introduction

Application security covers a broad spectrum of threats, vulnerabilities, and mitigation tools. It can be broken down into topic areas ranging from: platform security – exploring operating systems, software security – investigating application attacks and database security, and web application security – evaluating security in the client-side and server-side. This report presents the focused topic of client-side web application security.

The client-side refers to the activity taking place on the client's computer using "web browsers or other connections to make demands on servers" (Techopedia, 2019). The high-level of interactivity between the client and client-side means organisations hosting web applications are responsible for ensuring usage is risk-free and all parties are protected against malicious activity. Unfortunately, many web applications are rapidly developed and deployed with vulnerabilities that can be exploited by an attacker, harming both the organisation and its customers.

This report references three threats to client-side security: Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF), and Session hijacking attacks *(See Appendix A for definitions)*. The report begins by providing an extensive review of the relevant literature to client-side security and questions the prevalence of these threats in the modern day. Following this it explains the legal, ethical, and professional impact successful attacks have on information security management using real-world cases to illustrate the consequences. Finally, the report concludes with an insight into the preventative techniques and processes available to secure the client-side.

2: Literature review

To understand the prevalence of client-side attacks it is important to review existing literature prior to forming an opinion. Security is constantly changing so comparisons between older and newer studies were necessary to gauge if the three chosen topics (XSS, CSRF, Session Hijacking) are modern day threats. Appropriate literature was selected from books, reports and research papers available on Swansea Universities ifind service, IEEE Xplore, Google scholar, and Cybersecurity organisations.

Oriyano and Shimonski (2012) state that client-side attacks exposing the vulnerability of the end user and their system are one of the biggest threats that users will face today. This is a widely recognised and supported opinion in present day. Data (2018) adds that as well as inherent vulnerabilities, web applications are susceptible to attacks occurring as a result of design, configuration, or implementation flaws. Positive Technologies (2019) reinforce this message in their web application vulnerability statistics report for 2018, putting client-side attacks as the biggest threat to web application security, finding weaknesses in 91% of the 43 tested. Furthermore, when grading the security level 32% were "extremely poor", 23% were "poor", and 14% were "below average". This shows that a significant number of organisations are lacking adequate security measures.

The threat collection network Symantec (2019) reported an increase in web attacks on endpoints by 56% in 2018, blocking more than 1.3 million unique attacks every day, with the most popular attack Formjacking using client-side vulnerabilities like XSS to inject malicious JavaScript code stealing sensitive information from payment forms. These statistics stress the

importance of strong client-side security and show that due to the frequency and diversity of web attacks any present vulnerabilities will eventually be found and exploited.

Stuttard and Pinto (2011) when testing hundreds of web applications between 2007 and 2011 found the most common categories of vulnerability were XSS (94%) and CSRF (92%). Huang, Zhang, Cheng and Shieh (2017) reinforce the notoriety of XSS attacks in modern day by referring to them as being among the most critical security threats to today's websites. Edgescan (2019) concurred that web application security is still the area of most risk from a security breach standpoint in their 2019 vulnerability statistics report and found that XSS (14.69%) was still the most common vulnerability in 2018. Other vulnerabilities that can be exploited by different forms of session hijacking attacks were also common (13.32% collectively), and CSRF was significantly less common (1.75%).

Although less common, Nagpal, Chauhan and Singh (2014) refer to CSRF attacks as the sleeping giant of vulnerabilities due to many sites failing to protect against them because of being largely ignored by the web development community. Shema (2012) describes CSRF attacks as happening behind the scenes without the victim's knowledge, with the targeted application only seeing a valid request from a valid user. This makes it clear that an organisation cannot afford to overlook a CSRF vulnerability as a successful attack is only likely to be discovered after the damage is done. The CSRF protection used must be thoroughly tested to ensure it is performing as expected. Masri and Vlajic (2017) conducted a study of current Google chrome extensions claiming to protect against CSRF and a variation called a Cross-Site Framing Attack (CSFA). When tested against several variants of CSRF and CSFA attacks all five extensions failed to block every variant of the attacks. This research makes it clear that even tools claiming to provide protection cannot be trusted and security will only be achieved after rigorous vulnerability testing has taken place.

3: Professional, Legal, and Ethical Impact

The conducted literature review made it apparent that client-side attacks pose a substantial threat to modern day web application security. XSS, CSRF, and Session hijacking attacks can be utilised in a multitude of different ways to achieve malicious outcomes chosen by the attacker(s). They are not mutually exclusive, with attackers using XSS to achieve Session hijacking, or using an XSS vulnerability to plant code conducting a CSRF attack. This section explores the legal, ethical, and professional impact successful attacks have on information security management, using real-world examples to illustrate the consequences.

The impact of a successful client-side attack can be devasting for both an organisation and its customers. Most organisations use web applications to collect sensitive data ranging from personal information, to financial information, and even health, military and governmental information. Symantec's (2019) Internet Security Threat Report, explored in the literature review, identified intelligence gathering as the motive for 96% of attack groups. Large organisations with many connected computers and visitors are prime targets for methods such as session hijacking as the attacker can blend in with the traffic to stay hidden (Cucu, 2017). This means large organisations handling sensitive data are ideal targets for client-side attacks fashioned to steal this information.

When a successful client-side attack is performed, such as session hijacking or account hijacking with XSS or CSRF, against the users of an organisation (e.g. social networking

sites, financial institutions, medical organisations), the impact can be severe. If a regular users account is compromised the attacker could steal sensitive and identifying information such as an email address, phone number, bank details, private messages, health conditions, and other private details. Stolen information can be used to perform further attacks such as: scamming or forcing the user into making financial transactions, demanding a ransom to return sensitive or incriminating information such as health conditions or private messages, targeted malware downloads, or posing as the user to exploit the trust of the victim's friends.

If an administrators account is compromised the consequences are worse as the attacker could have permissions to gain access to a multitude of customer records, conducting the previously discussed attacks on a larger scale. Through the hijacked administrators account or a vulnerability such as XSS, the attacker can embed malicious code into the web application which when executed may: request further credentials for example passport or national insurance numbers to perform identity theft or download malware that can provide remote access to the user's computer and stored files.

Once the security of an organisation has been compromised the trust that the customer had in it will be diminished. If the customer was badly affected, they are likely to take their custom elsewhere or even take legal action against the organisation. As well as losing customers, breaches in security and leakage of data can result in large fines for organisations that have broken General Data Protection Regulations (GDPR). A recent example is the data breach that affected British Airways in 2018, incurring a hefty £183 million fine (BBC News, 2019).

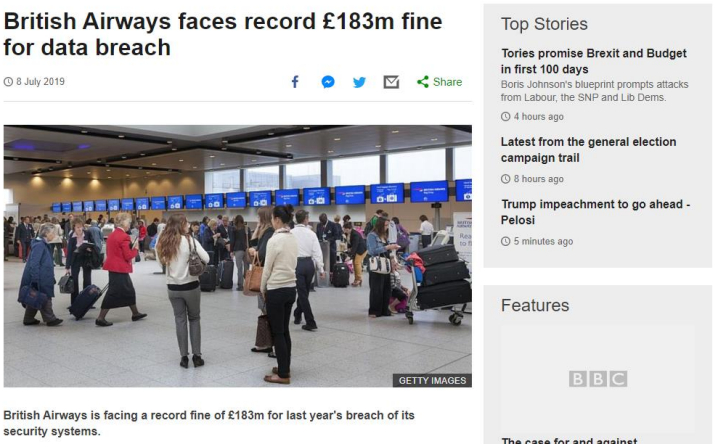

Figure 1: British Airways data breach news article. (BBC News, 2019).

The breach in question was speculated to be possible due to the exploitation of an XSS vulnerability in third-party software used by British Airways. The vulnerability allowed hackers to insert JavaScript redirecting users of British Airways' website to a fraudulent site. Sensitive data of approximately 500,000 customers was harvested including: login details, payment card details, travel bookings and other personal information such as name and address (Muncaster, 2019). The proposed fine of £183m was roughly 367 times as high as the previous record due to the arrival of a new law mirroring GDPR allowing fines of up to 4% of annual turnover (BBC News, 2019).

Variations of the same attack exploiting XSS vulnerabilities in third-party software have been conducted against other large organisations including Ticketmaster, e-commerce store Sweaty Betty, US gun maker Smith & Wesson and US retailer Macy's.

Figure 2: Ticketmaster data breach news article. (TicketingBusinessNews, 2019).

In the case of Ticketmaster, 40,000 customers were believed to have had personal or payment information stolen. Due to a suspected breach of GDPR a £5m lawsuit was launched on behalf of over 650 claimants (TicketingBusinessNews, 2019). Although GDPR regulations do not govern US companies' attacks can still incur massive costs.

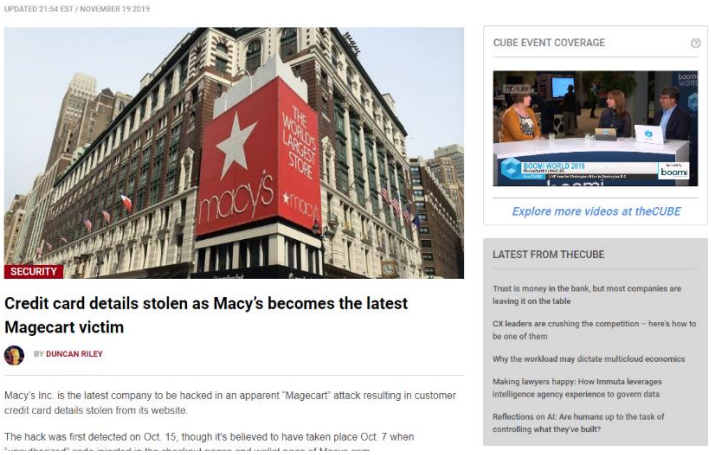

Figure 3: Macy's data breach news article. (Riley, 2019).

In the case of Macy's, the attack took place leading into the holiday shopping season and was not well received by investors leading to stock dropping by 11% (Riley, 2019). This proves that a negative company image can be costly enough for a business regardless of the legal reprimands.

Successful client-side attacks have been shown to not only cause a great deal of harm to the customer but also have devastating professional and legal implications on a business. In these situations, not only has the attacker been unethical but also the organisation that made the decision to cut corners and deploy a product that failed to protect their customers and provide information security. Nine months after the Ticketmaster data breach, HayesConnor Solicitors (2019) discovered that 63% of clients suffered from multiple fraudulent transactions on their payment cards and 31% of clients suffered from distress and/or psychological trauma. This shows how the incident caused by Ticketmaster's carelessness in security was unethical and had a lasting effect.

As it stands this report has illustrated a negative view of an attacker and looked at examples where vulnerabilities have been unethically exploited. To maintain a well-rounded viewpoint, it is also important to consider situations where an "attacker" has acted ethically. Security researcher Kristian Erik Hermansen exposed a vulnerability present in hundreds of banking

web sites provided by a major technology services organisation called Fiserv (KrebsonSecurity, 2018).

Figure 4: Fiserv Flaw exposes customer data news article. (KrebsonSecurity, 2018).

The vulnerability allowed an attacker to modify the sites code in browser to change an event number, which in turn would display the full records of a different customer. If utilised this could have exposed the personal and financial details of countless customers. Hermansen reported the flaw to his local bank. In this situation one would argue that the "attacker" has acted ethically as when given the choice, chose to help the organisation rather than exploit the flaw.

The final real-world example investigated is the attack on the Central Intelligence Agency website by a hacker called "lionaneesh" (Kumar, 2011). As seen in figure 5, an XSS vulnerability has been exploited to inject script inserting text with the hacker's twitter address.

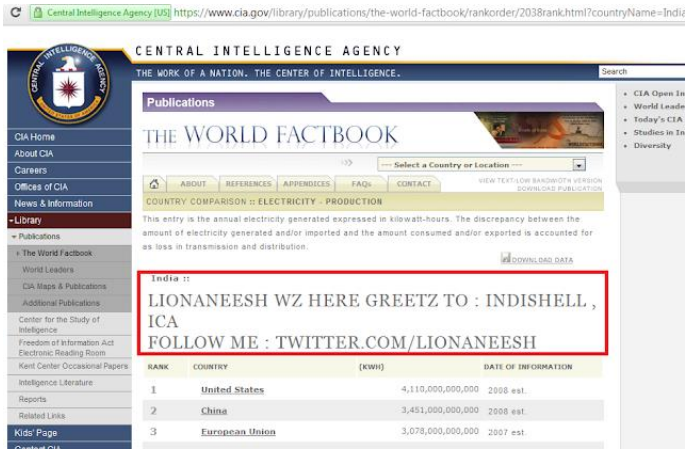

Figure 5:Screenshot of CIA XSS attack. (Kumar, 2011).

Judging whether this attack is unethical is a matter of personal opinion. On one hand, the attacker has exploited a vulnerability to deface a website. On the other hand, the attacker has pointed out an obvious flaw which can now be fixed.

4: Prevention Techniques

In summary, this report has investigated attacks against the client-side with focus on XSS, CSRF and session hijacking. A literature review was conducted to explore the prevalence of client-side attacks in the modern day. The impact of successful attacks was explained, illustrating the professional, legal and ethical consequences with relevant real-world examples. This report concludes by providing suggestions for techniques to secure the client-side.

4.1: Specific Attack Prevention

Content Security Policy (CSP) is an added layer of security which is used as a tool to detect and mitigate XSS and data injection attacks (Nath, 2018). CSP gives server administrators the ability to tell the browser which resources should be considered valid sources of executable scripts. Organisations can use this tool to whitelist approved resources and block unauthorised scripts from executing within their application. CSP's should be carefully designed and requires a high amount of attentiveness to ensure the application is fully covered.

CSRF tokens are unique, hidden, hash values that help prevent CSRF attacks. They are generated by the server-side of the application and transmitted to the client, to be included in the subsequent HTTP request they are making (Portswigger, 2019). The server-side will now validate each request by checking if it contains the expected token, rejecting it if the token is invalid or missing. This tool makes it impossible for an attacker to construct a valid request. For complete CSRF protection other techniques such as Same-Site cookies should be used alongside it. Same-Site cookies are cookies that are only sent if a request is made from its correct origin (Acunetix, 2019).

To help prevent session hijacking, using HTTPS applies SSL/TLS encryption to all session traffic rendering an intercepted session ID as unusable. HTTP Strict Transport Security (HSTS) should be used in conjunction to force connections over HTTPS (Banach, 2019). The developer should also enforce session key regeneration after initial authentication. Modern web frameworks offer sophisticated, secure algorithms for generating and managing the session ID which should be used by the developer.

4.2: Security management processes

After the implementation of specific attack prevention techniques, an organisation should take extra measures to manage security, with special considerations of how customer information is handled. The organisation deploying the application should adopt a security management process such as ITIL, which describes and measures IT security within an organization (Brahmachary, 2018). The organisation should perform objectives including:

- Design of security controls: Design the client-side with the appropriate prevention mechanisms against attacks, and put measures in place to ensure confidentiality, integrity, security and availability of data to the customer.
- Security Testing: Perform rigorous client-side testing using tools that simulate attacks to find vulnerabilities and evaluate effectiveness of security.
- Management of Security Incidents: Implement processes to detect attempted attacks, blocking the source and protocols to mitigate the damage caused by a breach.

- Security Review: Regularly check if security measures and procedures are up to date, and the client-side is operating as intended. Review all previous testing to see if components have been recently tested.

With the appropriate preventative techniques and security management processes in place an organisation should employ specialised employees responsible for managing and maintaining security. Basic security training should also be conducted with regular employees. Once this has been accomplished, the threat posed by client-side attacks should be minimised with both the customer and the business receiving adequate protection.

References

[1] Techopedia. (2019). Client-side. Retrieved from https://www.techopedia.com/definition/439/client-side

[2] OWASP. (2018). Cross-site Scripting (XSS). Retrieved from https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)

[3] Portswigger. (2019). Cross-site request forgery (CSRF). Retrieved from https://portswigger.net/web-security/csrf

[4] Banach, Z. (2019). What Is Session Hijacking: Your Quick Guide to Session Hijacking Attacks. Retrieved from https://www.netsparker.com/blog/web-security/session-hijacking/

[5] Shimonski, R., & Oriyano, S. (2012). Client-side attacks and defense. *Client-Side Attacks Defined* (pp. 1-24). Retrieved from http://www.sciencedirect.com/science/article/pii/B9781597495905000018

[6] Data, B., Bouvin, David, & Walters, Kelley. (2018). Organizational Cultural Barriers on the Implementation of Effective Web Application Security Policy: A Qualitative Case Study. *Web Application and Security* (pp. 27). Retrieved from https://search.proquest.com/docview/2128021899/fulltextPDF/F03CD5CAD2A240E5PQ/

[7] Positive Technologies. (2019). Web application vulnerabilities: statistics for 2018. Retrieved from https://www.ptsecurity.com/ww-en/analytics/web-application-vulnerabilities-statistics-2019/

[8] Symantec (2019). Internet Security Threat Report Volume 24. Retrieved from https://www.symantec.com/content/dam/symantec/docs/reports/istr-24-2019-en.pdf

[9] Stuttard, D., & Pinto, M. (2011). The web application hacker's handbook: Finding and exploiting security flaws. *Web Application (In) security* (pp. 1-15). Retrieved from https://ebookcentral.proquest.com/lib/swansea-ebooks/reader.action?docID=819008

[10] H. Huang, Z. Zhang, H. Cheng and S. W. Shieh. (2017). Web Application Security: Threats, Countermeasures, and Pitfalls. C*omputer*, vol. 50, no. 6, (pp. 81-85). Retrieved from https://ieeexplore.ieee.org/document/7945157

[11] Edgescan (2019). 2019 Vulnerability statistics report. Retrieved from https://www.edgescan.com/wp-content/uploads/2019/02/edgescan-Vulnerability-Stats-Report-2019.pdf

[12] Nagpal, B., Chauhan, N., & Singh, N. (2014). Cross-Site Request Forgery: Vulnerabilities and Defenses. *I-Manager's Journal on Information Technology*, *3*(2), (pp. 13-21). Retrieved from https://search.proquest.com/docview/1553397381/fulltextPDF/38ED0028E02444C6PQ/

[13] Shema, M. (2012). Hacking web apps: Detecting and preventing web application security problems. *Cross-Site Request Forgery (CSRF)* (pp. 79-104). Retrieved from https://ebookcentral.proquest.com/lib/swansea-ebooks/reader.action?docID=1012529

[14] M. E. Masri and N. Vlajic. (2017). Current state of client-side extensions aimed at protecting against CSRF-like attacks. *IEEE Conference on Communications and Network Security (CNS)* (pp. 390-391). Retrieved from https://ieeexplore.ieee.org/document/8228690

[15] Symantec (2019). Internet Security Threat Report Volume 24. Retrieved from https://www.symantec.com/content/dam/symantec/docs/reports/istr-24-2019-en.pdf

[16] Cucu, P. (2017). Session Hijacking Takes Control of Your Accounts. Here's How. Retrieved from https://heimdalsecurity.com/blog/session-hijacking/

[17] BBC News. (2019). British Airways faces record £183m fine for data breach. Retrieved from https://www.bbc.co.uk/news/business-48905907

[18] Muncaster, P. (2019). BA's Magecart Breach Lands it £183m GDPR Fine. Retrieved from https://www.infosecurity-magazine.com/news/bas-magecart-breach-lands-it-183m/

[19] TicketingBusinessNews. (2019). TICKETMASTER SUED AFTER 2018 DATA BREACH. Retrieved from https://www.theticketingbusiness.com/2019/04/08/ticketmaster-sued-2018-data-breach/

[20] Riley, D. (2019). Credit card details stolen as Macy's becomes the latest Magecart victim. Retrieved from https://siliconangle.com/2019/11/19/credit-card-details-stolen-macys-becomes-latest-magecart-victim/

[21] HayesConnor Solicitors. (2019). Updates on the Ticketmaster Data Breach. Retrieved from https://www.hayesconnor.co.uk/ticketmaster-data-breach-updates/

[22] Kumar, M. (2011). XSS attack on CIA (Central Intelligence Agency) Website by lionaneesh. Retrieved from https://thehackernews.com/2011/06/xss-attack-on-cia-central-itelligence.html

[23] Nath, R. (2018). HTTP Content Security Policy (CSP). Retrieved from https://www.rahulpnath.com/blog/http-content-security-policy-csp/

[24] Portswigger. (2019). CSRF tokens. Retrieved from https://portswigger.net/web-security/csrf/tokens

[25] Acunetix. (2019). CSRF Attacks: Anatomy, Prevention, and XSRF Tokens. Retrieved from https://www.acunetix.com/websitesecurity/csrf-attacks/

[26] Brahmachary, A. (2018). ITIL Information Security Management | ITL Foundation | ITSM. Retrieved from https://www.certguidance.com/information-security-management-itil/

Appendix A: Supplementary definitions

Cross-Site Scripting (XSS): A vulnerability in a web application that enables the attacker to inject malicious client-side scripts into web pages viewed by the victim's browser. OWASP (2018) categorises XSS attacks into three categories: Stored/Persistent, Reflected/Non-Persistent and DOM Based.

Stored/Persistent: Attacker forces the server to store malicious code, rendering it on each newly opened page.

Reflected/Non-Persistent: Attacker temporarily inserts code into the current page, usually executes and reflects at the user when the application attempts to render page data.

DOM Based: DOM environment in victims' browser is modified to execute attack payload. Ensures page does not change but contained client-side code executes differently.

Cross-Site Request Forgery (CSRF): A vulnerability that enables the attacker to force the victim's browser to perform actions they did not intend to perform. A successful CSRF attack forces the victim's browser to make a request to carry out an action such as changing their password, which appears real to the server (Portswigger, 2019).

Session Hijacking: An attack that relies on exploiting a vulnerability in the client-side to steal the victim's session ID. The attacker will then use the authenticated session ID for their own browser session, allowing the attacker to impersonate the victim (Banach, 2019). There are many unique approaches to accomplishing this including: Session fixation, Session sniffing, XSS, Man-in-the-middle attacks, and Man-in-the-browser attacks.